Real-Time Analog Global Optimization with Constraints: Application to the Direction of Arrival Estimation Problem

Thomas M. Jelonek James P. Reilly*
Qiang Wu
Communications Research Laboratory
McMaster University
Hamilton, Ontario
Canada
L8S 4K1

October 26, 1994

Abstract

An analog technique for real-time, multi-variate, global optimization with constraints is presented. The basic structure is a simple gradient descent loop, where the gradients are computed using an analog neural network. Constraints are implemented using a variation of an idea due to [17], where neural networks are also used to implement the required constraint functions. It is shown that the system converges to a stable equilibrium point, which satisfies the Kuhn-Tucker conditions for a constrained minimum. *Global* optimization is achieved by introducing a diffusion process into the governing differential equation. This procedure is a continuous-time analog of the *simulated annealing algorithm.*

Even though the proposed method is applicable to a wide range of engineering problems, the real-time, global and other capabilities of this method are demonstrated specifically with an optimization problem from array signal processing- the maximum likelihood direction of arrival estimator. The satisfactory performance of all aspects of this proposed optimization technique is demonstrated by simulations.

*Person to whom correspondence should be addressed

1 Introduction

Real-time nonlinear optimization is a generic problem often encountered in several engineering disciplines. Despite the enormous advances made in the area of optimization in modern times, there has been relatively little attention given to the *real-time* aspect of this problem. Yet, many engineering problems would be greatly alleviated if an effective real-time process for nonlinear optimization were available.

Digital computing methods have not proven completely adequate for this problem. In all but the simplest cases, digital optimization methods are too slow for real-time use, especially when global optimizations are required. This is in spite of the recent simulated annealing methods [19] which have emerged.

Recently, much work has been done in proposing analog circuit techniques for nonlinear programming or optimization with the idea of real-time processing in mind [13] [1] [2] [17] [36] [24] [21]. Hopfield [13] proposed an analog circuit which seeks a minimum point of a quadratic energy function. Kennedy and Chua [17] extended the work of [1] to realize nonlinear programming circuits with constraints. Zhang and Constantinides generalized the constraint aspect of [17] and proposed a set of differential equations which could be realized by neural network circuits to solve the nonlinear programming problem. However, none of these previous approaches have considered the *global* optimization aspect of the problem.

In this paper, we further generalize on the use of analog neural networks for real-time optimization with constraints, and also, we propose a technique for attaining the global minimum of the desired function [16]. This paper is an extension and combination of other disparate contributions in this field; one is from [17] [1] [2] [36] which propose non-global optimizations for constrained non-linear optimization; another is the work of [3] [8] and others which lead to a simple elegent means of implementing global minimization.

The proposed optimization technique is applied to the direction of arrival (DOA) problem in array signal processing as an example. In this problem, it is desired to estimate the directions of incident plane waves onto arrays of sensors. There are many well-known high-resolution algorithms for performing this estimation process, yet, due to the computational complexities involved, none are appropriate for true real-time application. One well-known DOA technique, which is the method considered here, is the *maximum likelihood estimator* (MLE) e.g., [22] [37]. This technique involves the maximization of a computationally intensive likelihood function to determine the desired directions of arrival. Even though the proposed optimization method is applicable to a wide range of engineering problems, the specific DOA example is intended to be a useful contribution in its own right. This is because the real-time implementation of modern DOA estimation algorithms continues to be a difficult problem. An alternative approach which in essence performs a discrete exhaustive search over the range of possible DOA's using a Hopfield network, is presented in [12].

The method of [17] is limited in practice (but not in concept) to solving linear or quadratic

programming problems with constraints, instead of arbitrary nonlinear problems. This is due to the difficulty in implementing the derivative functions required by the method. Here, we propose to extend this work to the case of arbitrary functions by using *analog neural networks* to implement the required derivatives and constraint functions. Further, by implementing a practical version of a method proposed by [8], a technique for real- time global optimization of arbitary functions with constraints is achieved.

The global convergence aspect of our proposed system is realized by a continuous-time analog of the simulated annealing algorithm [19]. A white noise component is added to the gradient of the Lagrangian of the function to be minimized. As is shown, the variance of this noise must decay according to a specified annealing schedule for the system to converge to a global optimum. However, this proposed annealing schedule is too slow for real-time use. We therefore propose to accelerate this annealing schedule, with the result that global convergence can no longer be guaranteed, but only assured with high probability. For our specific DOA example, we propose a simple statistical criterion to test whether an equilibrium point achieved by the system does indeed correspond to the desired global optimum.

We first present the analysis relating to the proposed optimization system. A brief summary of the mathematics of maximum likelihood DOA estimation is then presented. After this, several details concerning the realization of the DOA example using this proposed optimization technique are given. The performance of the proposed optimization method is demonstrated by a variety of computer simulations carried out on the DOA example. Global constrained convergence of the system is successfully demonstrated for a range of conditions. The real- time capability of the proposed system is also demonstrated.

2 Development of Circuit for Optimization

In this section, we first discuss a simple circuit for local optimization without constraints, using a steepest descent procedure. The method is then expanded to incorporate constraints, and then a simple extension is incorporated so that the resulting circuit performs *global* optimization.

In this paper we consider the minimization of a scaler objective function $L(\boldsymbol{\phi})$, where $\boldsymbol{\phi}$ is a $K \times 1$ vector of variables. We make the following mild asumptions on $L(\boldsymbol{\phi})$:

1. $L(\boldsymbol{\phi})$ is assumed twice continuously differentiable.
2. at least one minimum exists, and all minima of the objective function are unique.

It is widely known that minimization by steepest descent may be implemented by a system

whose dynamics are described by

$$C_k \frac{d\phi_k}{dt} = -\frac{\partial L(\boldsymbol{\phi})}{\partial \phi_k} \qquad k = 1 \ldots K, \tag{1}$$

where C_k is a capacitance. In the analog system we propose, $\boldsymbol{\phi}$ is interpreted as a voltage, and L as a potential or energy function. Note that the complete system consists of K such loops, with each gradient term being a function of all K variables. Eq. (1) implies that the rate of change of all the variables is proportional to the gradient of the objective function. Thus, the system moves in the direction of steepest descent, towards a local minimum.

The difficulty until recently with such systems, for all but the simplest functional forms, has been in implementing the gradient terms. In the general case, the gradients can be arbitrarily complex functions of $\boldsymbol{\phi}$, and thus can be very difficult to implement using conventional circuit elements. In this paper, we propose to use an analog neural network for the evaluation of the gradients. It is shown in [20] that a neural network with a sufficiently large number of hidden nodes can approximate an arbitrary function to any degree of accuracy. This property is the basis for training a neural network to approximate the required gradient of the desired objective function. The use of neural networks for this purpose is discussed in more detail in Sect. 4. The neural network output is interpreted as a current source.

The introduction of constraints on this optimization problem offers several advantages. First, as is discussed later, constraining the feasible region of the parameter values makes the training of the neural networks much easier. Also, the use of constraints is valuable in its own right, since in some cases it may be known *a priori* that some of the parameters may be confined to certain regions of the parameter space.

We now wish to generalize the minimization of the objective function $L(\boldsymbol{\phi})$ by introducing J inequality constraints. Our problem of interest thus becomes

$$\min_{\boldsymbol{\phi}} L(\boldsymbol{\phi}) \tag{2}$$

subject to

$$\psi_j(\boldsymbol{\phi}) \geq 0, \qquad j = 1 \ldots J. \tag{3}$$

We assume the following mild conditions hold on the constraint functions $\psi_j(\boldsymbol{\phi})$:

1. the constraints satisfy certain constraint qualifications [9]
2. Each ψ_j is twice continuously differentiable
3. at least one local minimum of the constrained problem exists, and all constrained local minima are unique

In this treatment we distinguish between active constraints and inactive constraints. A constraint $\psi_j(\boldsymbol{\phi})$ is *active* if $\psi_j(\boldsymbol{\phi}^*) = 0$, where $\boldsymbol{\phi}^*$ is the solution to (2) and (3). Otherwise

the constraint is *inactive* and $\psi_j(\boldsymbol{\phi}^*) > 0$. The *feasible region* is the parameter subspace in which all the constraint equations (3) are satisfied.

A standard means of solving (2) and (3) is to consider the Lagrangian $H(\boldsymbol{\phi}, \boldsymbol{\lambda})$ given by

$$H(\boldsymbol{\phi}, \boldsymbol{\lambda}) = L(\boldsymbol{\phi}) - \sum_{j=1}^{J} \lambda_j \psi_j(\boldsymbol{\phi}) \tag{4}$$

where λ_j are Lagrange multipliers.

The set of Kuhn-Tucker sufficient conditions for any local minimum satisfying (2) and (3) are based on the above Lagrangian, and are given as [9]

1. $$\frac{\partial L}{\partial \phi_k}(\boldsymbol{\phi}^*) - \sum_{j=1}^{J} \lambda_j^* \frac{\partial \psi_j}{\partial \phi_k}(\boldsymbol{\phi}^*) = 0 \qquad k = 1 \ldots K. \tag{5}$$

   This equation imposes the condition that the Lagrangian $H(\boldsymbol{\phi}, \boldsymbol{\lambda})$ is stationary at $\boldsymbol{\phi} = \boldsymbol{\phi}^*$. The Lagrange multipliers corresponding to inactive constraints are implicitly zero.

2. $$\psi_j(\boldsymbol{\phi}^*) \geq 0, \qquad j = 1 \ldots J, \tag{6}$$

   This condition enforces satisfaction of the constraints at the solution.

3. $$\lambda_j^* \geq 0, \qquad j = 1 \ldots J, \tag{7}$$

   where the λ^*'s are the Lagrange multipliers corresponding to $\boldsymbol{\phi}^*$. If a constraint is active, the inequality in (7) holds; for inactive constraints, the equality is satisfied. If condition (7) is not satisfied, then there may exist feasible directions for $\boldsymbol{\phi}$ about $\boldsymbol{\phi}^*$ such that $L(\boldsymbol{\phi}^*)$ decreases in value. This contradicts the notion of a local minimum.

4. $$\left[\frac{d^2}{d\theta^2} L(\alpha(\theta))\right]_{\theta=0} > 0 \tag{8}$$

   where $\alpha(\theta)$ is an arc satisfying the equality of any of the active constraints (3), parameterized by θ, where $\theta = 0$ at $\boldsymbol{\phi} = \boldsymbol{\phi}^*$ [9]. This statement implies that the objective function $L(\boldsymbol{\phi})$, for $\boldsymbol{\phi}$ satisfying any active constraint $\psi_j(\boldsymbol{\phi}) = 0$, has positive curvature at $\boldsymbol{\phi} = \boldsymbol{\phi}^*$.

Thus, any point $\boldsymbol{\phi}^*$ satisfying the above conditions is a local minimum of the problem defined by (2) and (3).

Consider the circuit whose governing differential equation is an extension of (1) and is defined by

$$C_k \frac{d\phi_k}{dt} = -\frac{\partial L}{\partial \phi_k} + \sum_{j=1}^{J} \lambda_j \frac{\partial \psi_j(\boldsymbol{\phi})}{\partial \phi_k}, \qquad k = 1 \ldots K. \tag{9}$$

Also consider the network function $\mathcal{N}(\cdot)$ [17] defined by :

$$\mathcal{N}(x) = \begin{cases} 0 & \text{if } x > 0 \\ -\frac{1}{R}x & \text{otherwise} \end{cases} \tag{10}$$

where R is a resistance. Let us now substitute the Lagrange multipliers λ_j in (9) by the network function $\mathcal{N}(\psi_j(\boldsymbol{\phi}))$. The governing differential equation of this circuit is described by (11) and a physical embodiment is shown in Fig. 1a. This circuit is the same as the canonical circuit of Kennedy *et al* for nonlinear programming [17], and is discussed in more detail in the sequel.

$$C_k \frac{d\phi_k}{dt} = -\frac{\partial L}{\partial \phi_k} + \sum_{j=1}^{J} \mathcal{N}_j(\psi_j(\boldsymbol{\phi})) \frac{\partial \psi_j(\boldsymbol{\phi})}{\partial \phi_k}, \qquad k = 1 \ldots K. \tag{11}$$

We now discuss the stability analysis of this system. Consider the energy function corresponding to (11) that is obtained by integrating the negative of (11) with respect to $\boldsymbol{\phi}$

$$E(\boldsymbol{\phi}) = L(\boldsymbol{\phi}) - \sum_{j=1}^{J} \int_0^{\psi_j} \mathcal{N}_j(x) dx. \tag{12}$$

Differentiating $E(\boldsymbol{\phi})$ with respect to time, we get

$$\frac{dE(\boldsymbol{\phi})}{dt} = \sum_{k=1}^{K} \frac{\partial L}{\partial \phi_k} \frac{d\phi_k}{dt} - \sum_{k=1}^{K} \sum_{j=1}^{J} \mathcal{N}_j(\psi_j(\boldsymbol{\phi})) \frac{\partial \psi_j}{\partial \phi_k} \frac{d\phi_k}{dt} \tag{13}$$

However, factoring out $d\phi_k/dt$ and substituting (11) into (13) we get

$$\frac{dE}{dt} = -\sum_{k=1}^{K} C_k \left(\frac{d\phi_k}{dt} \right)^2. \tag{14}$$

We note that because (10) is continuous for finite R, and due to our assumptions on the function $L(\boldsymbol{\phi})$, the function $E(\boldsymbol{\phi})$ has the following properties:

1. $E(\boldsymbol{\phi})$ is continuously differentiable.
2. $E(\boldsymbol{\phi}^*) = 0$. [1]
3. $E(\boldsymbol{\phi}) > 0$ everywhere except at $\boldsymbol{\phi}^*$.

[1] Without loss of generality, a suitable constant may be added to $E(\boldsymbol{\phi})$ so that this condition is satisfied.

4. Furthermore, from (14), note that C_k is strictly positive and $(d\phi_k/dt)^2 \geq 0$ for all ϕ_k, for $k = 1 \ldots K$. Hence for the function $E(\boldsymbol{\phi})$ defined in (12) we have $dE/dt \leq 0$. Since $(d\phi_k/dt)^2, k = 1 \ldots K$ are all zero only at the equilibrium points, dE/dt is *strictly* less than zero everywhere else.

This implies that $E(\boldsymbol{\phi})$ is a *Lyapunov function* [18] of the system. Thus, the system is *asymptotically stable*. Consequently, any trajectory of $\boldsymbol{\phi}(t)$ eventually converges to some stable equilibrium point $\boldsymbol{\phi}^*$ in $\Re^K$, which depends on the initial state.

Further, for sufficiently small R, it is easily shown, e.g., [17], that this equilibrium point satisfies the Kuhn-Tucker sufficient conditions (5) - (8).

These conditions are satisfied under the interpretation that the capacitor voltages of Fig. 1a are taken as the variables ϕ_i, and the constraint currents $i_j = \mathcal{N}_j(\psi_j(\boldsymbol{\phi}))$ are taken as the Lagrange multipliers λ_j.

Thus, we have the important result that our proposed system indeed converges to a stable equilibrium point, and furthermore because this equilibrium point satisfies the Kuhn-Tucker sufficient conditions, it corresponds to a constrained local minimum of our objective function.

We now discuss details of the system described by (11). The variables $\boldsymbol{\phi}$ in Fig. 1a appear as voltages across the capacitances. The right-most column of sources are voltage-controlled current sources, representing the first (gradient) term of (11), and are generated by the analog neural network. The center column of sources represent the second (constraint) term of (11), and are current controlled current sources. The left-most column shows voltage controlled voltage sources which generate the constraint functions $\psi_j(\boldsymbol{\phi})$.

If a constraint $\psi_j(\boldsymbol{\phi}^*)$ is inactive, then the generated voltage is positive, resulting in $i_j = 0$ due to the reversed diode. Then, this constraint does not contribute to the center-column source current. As the resistance R in $\mathcal{N}(\cdot)$ approaches zero, the current i_j is nonzero when the corresponding constraint is active, and the voltage value $\psi_j(\boldsymbol{\phi}) \rightarrow 0$ as required to satisfy the constraint. In this case, the constraint does contribute to the current generated in the center-column current sources.

If the system described by (11) is not in equilibrium, the trajectory of $\boldsymbol{\phi}$ moves in the downhill direction, closer to a point where where the Lagrangian is stationary. Since $\mathcal{N}$ only passes current if the constraints are not satisfied, the second term of (11) enforces the constraints by playing the role of a penalty function, contributing in the opposite direction of the gradient if the constraints are not satisfied.

We now consider the implementation of the constraint term in (11). Since each second term of (11) is a function only of $\boldsymbol{\phi}$, we propose that the constraint functions also be implemented using an analog neural network. The constraint sub-circuits on the left of Fig. 1a may thus be compacted into the constraint neural network shown in Fig. 1b. The advantage of

using neural networks to implement the constraint sub-circuit is that the complexity of the constraint functions ψ_j's which can be realized by this circuit is limited only by the ability of the neural network to approximate the desired function. Previous approaches are restricted to very simple functional forms of the constraints because they use only conventional circuit elements to realize the constraint sub- circuit.

A "constraint" neural net must exist for each variable loop. The K variables $\boldsymbol{\phi}$ are input to each net, which outputs the appropriate scaler constraint term corresponding to the second term of (11). In the event the ψ_j functions are simple, e.g., simple upper or lower bounds on the variables, then the constraint sub-circuit may be implemented in a much simpler way.

2.1 Global Optimization

We now consider how the system described so far can be adapted for global optimization. As previously seen, it is guaranteed that the equilibrium of the proposed circuit is within the Kuhn-Tucker set. However, it is not guaranteed that the whole Kuhn-Tucker set, and thus the global optimum, can be reached simply by following trajectories of the circuit (i.e., the trajectories can fall into local minima). However, we now show that random fluctuations introduced into the path of $\boldsymbol{\phi}$ greatly improve the ability of the circuit to eventually converge to the global optimum. We propose a continuous-time analog of the *simulated annealing algorithm* to accomplish this desired characteristic.

Simulated annealing (annealed diffusion process, etc.) refers to a class of numerical algorithms that imitate the slow cooling (annealing) of a substance to obtain a crystalline state. The cooling stage of an annealing procedure is usually done very slowly so that the number of defects in the crystal structure is reduced, and the potential energy stored in the molecular structure minimized.

Simulated annealing has enjoyed much success in many diverse optimization algorithms. Kirkpatrick, Gellat, and Vecchi [19] found simulated annealing to be a useful algorithm for optimizing VLSI layout and rooting problems arising in the design of computer systems. Geman and Geman [7] used an annealing algorithm to carry out image restoration of noisy blurred images. Simulated annealing was also used by El Gamal [5] to design good communication codes. More recently, Sharman [29] used simulated annealing to achieve global optimization of the maximum likelihood objective likelihood function for array signal processing.

While these techniques have focused on discrete simulated annealing, we are investigating simulated annealing in a *continuous form.* A continuous path, $\boldsymbol{\phi}(t)$ seeking a *global* minimum will in general be forced to "climb hills" as well as follow down-hill gradients. One way of introducing hill- climbing, while preserving the tendency to descend along gradients, is to introduce random fluctuations into the path of $\boldsymbol{\phi}$.

The overall global-seeking characteristic is accomplished in a rather simple way by using the very elegant result of [8]. The method is introduced by means of the following theorem:

Theorem 1 *Consider the diffusion process defined by*

$$d\boldsymbol{\phi}(t) = -\mathbf{Y} \odot [\nabla E(\boldsymbol{\phi}(t))]\, dt + \sigma(t) dW(t) \tag{15}$$

where $\mathbf{Y} = (1/C_1, 1/C_2, \ldots, 1/C_K)^T$, *the symbol* $\odot$ *means Schur product (element-by-element multiplication),* $\nabla(\cdot)$ *denotes the* $K \times 1$ *gradient operator,* $E(\boldsymbol{\phi}(t))$ *is the potential function defined by (12),* $W(t)$ *is a standard Wiener Process in* $\Re^K$, *and* $\sigma(t)$ *is a noise "temperature" defined by*

$$\sigma^2(t) = \frac{c}{\log(1+t)} \tag{16}$$

for large t *and with* $c > c_*$, *where* c_* *is a constant which may be determined from* $L(\boldsymbol{\phi})$ *and is defined later.*
Under suitable mild conditions [3] [8] on $L(\boldsymbol{\phi})$, *the trajectory* $\boldsymbol{\phi}(t)$ *approaches (weakly), an equilibrium distribution which is a Gibbs distribution* $\pi_\sigma(\boldsymbol{\phi})$, *with density*

$$\pi_\sigma(\boldsymbol{\phi}) = \frac{1}{Z_\sigma} \exp[\frac{-2L(\boldsymbol{\phi})}{\sigma^2}], \quad \textit{where} \quad Z_\sigma = \int_{R^K} \exp[\frac{-2L(\boldsymbol{\phi})}{\sigma^2}] d\boldsymbol{\phi}. \tag{17}$$

Then, as $t \to \infty$, π_σ *concentrates on the global minimum of* $L(\boldsymbol{\phi})$.

Proof: see [8].

Hence, as $t \to \infty$, $\sigma(t) \to 0$, and we can expect to find $\boldsymbol{\phi}$ at the global optimum.

The differential equation governing the behaviour of the resulting system including provision for global convergence, is given by substituting the gradient of (12) (which is given by (11)) into (15) as:

$$C_k \frac{d\phi_k}{dt} = -\frac{\partial L}{\partial \phi_k} + \sum_{j=1}^{J} \mathcal{N}_j(\psi_j(\boldsymbol{\phi})) \frac{\partial \psi_j(\boldsymbol{\phi})}{\partial \phi_k} + C_k \sigma(t) \frac{dW(t)}{dt}, \qquad k = 1 \ldots K. \tag{18}$$

The term $C_k \sigma(t) dW(t)/dt$, on the one hand helps the algorithm to "escape" from local stationary points and proceed to find the global optimum, and on the other hand it is a random process and has to attenuate fast enough in order for the algorithm to converge.

The following three conditions on $\sigma(t)$ are sufficient [8] for convergence of $\boldsymbol{\phi}$ to the global optimum:

1. When $t \to \infty$, $\sigma(t) \to 0$

2. $\sigma(t)$ is continuously differentiable with respect to t.

3. Define the variable $T(t) = \sigma^2(t)/2$. When $t \to \infty$,

$$\frac{dT(t)/dt}{T^3(t)} e^{2\Delta_L/T(t)} \to 0 \tag{19}$$

where

$$\Delta_L = \sup_{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2 \in \mathcal{R}^K} |L(\boldsymbol{\phi}_1) - L(\boldsymbol{\phi}_2)|$$

denotes the largest possible difference between any two values of $L(\boldsymbol{\phi})$ in the K-dimensional space.

The quantity c_* associated with (16) represents the initial noise variance which must be chosen carefully. If the initial noise variance is set too small, the state space trajectories may not be able to easily escape local minima in the objective function. Likewise, if c_* is initiated too large, the time evolution to a global minimum will be prohibitively long. The following theorem provides a proper determination of the quantity c_*.

Theorem 2 *Eq. (19) is satisfied if*

$$c_* > 2\Delta_L. \tag{20}$$

Proof: [35].

Thus, convergence is assured if c_* is chosen according to (20). Note that the annealing schedule is determined solely by the characteristics of the objective function $L(\boldsymbol{\phi})$.

With the advent of commercially available analog neural networks (e.g., the Intel *electronicaly trainable analog neural network* (ETANN) chip, [14]), all aspects of this proposed optimization process are readily implemented in analog form.

3 Maximum Likelihood Estimation of Directions of Arrival

In this section, we briefly discuss the problem of direction of arrival (DOA) estimation onto arrays of sensors, so that our proposed optimization technique may be applied to this problem as an example. The details of the application are discussed in Sect. 4. DOA estimation is a common problem experienced in the field of signal processing. There are many algorithms which have been proposed for solving this problem [28] [25] [22] [30] [31] etc., yet all have difficulty with the aspect of *real-time* implementation. The performance

of the maximum likelihood estimator (MLE) method e.g. [22], for DOA estimation which we propose is not the best of all available algorithms; e.g., [30] performs better; however, the MLE objective function is simple in structure and hence is well-suited as an example of this proposed optimization technique, its performance is close to optimum, and it does perform significantly better than several other commonly-used high-resolution techniques. The MLE method is discussed elsewhere; however, for the sake of completeness of this paper it is briefly developed here.

Consider a uniformly spaced, linear array of sensors consisting of M elements with K plane waves incident onto the array from distinct directions. (We consider the uniform linear array scenario for ease of discussion; the treatment may easily be extended to non-uniform sensor spacings and two-dimensional arrays.) We assume that these plane waves are narrowband, with the same carrier frequency, f_c, and that the number K is known, or can be estimated by a suitable detection method, e.g., [32]. The problem is to estimate the DOA's $\boldsymbol{\phi}$ of the incident plane waves by observing the array outputs $\mathbf{x}_n$.

The M-dimensional complex received signal vector, $\mathbf{x}_n \in \mathcal{C}^M$, represents the data received from a linear array of M sensors at the n^{th} snapshot. The complex received signal vector, for $n = 1, \ldots, N$, is given by

$$\mathbf{x}_n = \mathbf{S}\mathbf{a}_n + \mathbf{v}_n, \tag{21}$$

where $\mathbf{v}_n \in \mathcal{C}^M$ is the noise signal, $\mathbf{a}_n \in \mathcal{C}^K$ describes the instantaneous complex amplitudes and N is the number of snapshots observed. The k^{th} column $\mathbf{s}_k$ (the steering vector) of $\mathbf{S}$, is given as

$$\mathbf{s}_k = \left(1, e^{j\phi_k}, e^{j2\phi_k}, \ldots, e^{j(M-1)\phi_k}\right)^T, \quad k = 1 \ldots K, \tag{22}$$

where superscript T denotes transpose. The variable ϕ_k is the electrical phase shift between the signals received from adjacent array elements, due to the kth incident wave.

Assuming that the noise, $\mathbf{v_n}$, is Gaussian distributed with zero mean and covariance matrix $\gamma^2\mathbf{I}$, the maximum likelihood estimate of $\boldsymbol{\phi}$ given the received data is determined as the solution to

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} L(\boldsymbol{\phi}), \tag{23}$$

where

$$L(\boldsymbol{\phi}) = \mathrm{tr}\left[\mathbf{P}^{\perp}\hat{\mathbf{R}}_N\right], \tag{24}$$

and

$$\mathbf{P}^{\perp} = \mathbf{P}^{\perp}(\boldsymbol{\phi}) = \mathbf{I} - \mathbf{S}(\boldsymbol{\phi})(\mathbf{S}^{\dagger}(\boldsymbol{\phi})\mathbf{S}(\boldsymbol{\phi}))^{-1}\mathbf{S}^{\dagger}(\boldsymbol{\phi}) \tag{25}$$

is the projection matrix onto the orthogonal complement subspace spanned by $\mathbf{S}$, and
$\hat{\mathbf{R}}_N$ is the estimate of the covariance matrix of $\mathbf{x}_n$ over N snapshots:

$$\hat{\mathbf{R}}_N = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^{\dagger}. \tag{26}$$

It is shown in [23] that a sufficient value for the quantity ΔL is given by the expression

$$\Delta L < \mathrm{tr}\left(\hat{\mathbf{R}}_N\right). \tag{27}$$

The quantity $\mathrm{tr}(\hat{\mathbf{R}}_N)$ is a very simple value to calculate, and this form is used later for determination of the constant c_*.

Due to the symmetry of the state space, we propose incorporating the following constraint into the DOA network:

$$\psi_k(\boldsymbol{\phi}) = \phi_k - \phi_{k+1} > 0, \qquad k = 1 \ldots K-1. \tag{28}$$

These constraints reduce the searching field of the algorithm and significantly simplify the training of the neural networks. This consideration is discussed in more detail in Sect. 6. Since the ordering of the DOA's is arbitrary, these constraints do not alter the mathematical behaviour of the network.

It is apparent in the next section that an explicit expression for the likelihood gradient, $\partial L/\partial \boldsymbol{\phi}$, is required for our implementation. Thus the gradient of the objective function (23) with respect to $\boldsymbol{\phi}_k$ is given by [15]:

$$\frac{\partial L}{\partial \phi_k} = -2\,\Re e\left\{\mathrm{tr}\left[\mathbf{P}^{\perp}(\boldsymbol{\phi})\frac{\partial \mathbf{S}(\boldsymbol{\phi})}{\partial \phi_k}(\mathbf{S}^{\dagger}(\boldsymbol{\phi})\mathbf{S}(\boldsymbol{\phi}))^{-1}\mathbf{S}^{\dagger}(\boldsymbol{\phi})\hat{\mathbf{R}}_N\right]\right\}, \qquad k = 1, \ldots, K. \tag{29}$$

This is the equation that is mapped onto the neural network for evaluation of the gradient.

4 Mapping the Objective Function onto the Circuit

In this section, we present the direction of arrival estimation problem as an example for our proposed optimization technique. We show details of the circuit for the specific DOA implementation, and discuss several practical issues relating to the realization of the method.

A feedforward analog neural network (also referred to as a *connectionist* system) is used to approximate the gradient of the objective function to be minimized. The overall block diagram of the system for solving the maximum likelihood DOA estimation problem is shown in Fig. 2. Note from the gradient expression (29) that the argument of the trace operator consists of two terms; one is $\mathbf{P}^{\perp}\partial\mathbf{S}/\partial\phi_k(\mathbf{S}^{\dagger}\mathbf{S})^{-1}\mathbf{S}^{\dagger}$ which is a function of $\boldsymbol{\phi}$ only. The other term is $\hat{\mathbf{R}}_N$. The fact that the gradient is separable in this way greatly aids the evaluation of the gradient, since the neural network need only provide the matrix function which is dependent on $\boldsymbol{\phi}$, and does not need to be trained to accomodate the wide variation of possible values of the matrix $\hat{\mathbf{R}}_N$. Thus, the input to the network is $\boldsymbol{\phi}$ and the output is $\mathbf{P}^{\perp}\partial\mathbf{S}/\partial\phi_k(\mathbf{S}^{\dagger}\mathbf{S})^{-1}\mathbf{S}^{\dagger}$. The network output is then multiplied in analog fashion by the covariance matrix, $\hat{\mathbf{R}}_N$. The trace and real operations are then performed to yield the

desired gradient expression. There are K networks; one for each of the partial derivatives that compose the gradient vector. Note that the input and output values of the feedforward network are continuously varying functions of $\boldsymbol{\phi}$.

The derivative of a Wiener process is white noise [6]. Hence, the last term of (18) is represented by the white noise source shown in Fig. 2. The variance of the noise is controlled by eq. (16). Also shown in the figure is the simple network for implementing the constraints (28). In this example, the constraints are so simple that the constraint sub-circuit may be implemented using only resistors, diodes and op-amps. The quantity leaving the summing junction is a current representing the quantity $C_k d\phi_k/dt$ of (18). This current may be integrated by the capacitor to yield a voltage value representing $\phi_k(t)$. These values are then fed back into the tops of each of the K neural network sub-systems to complete the loop.

The capacitance C_k adjusts the time scale of the system. To see this, the variable t in (11) may be transformed into $\tau = t/C_k$; hence, the time axis is scaled inversely proportionally to C_k. The entire theoretical treatment of Sect. 2.1 may be undertaken with t replaced by τ. The result is that the governing differential equation (18), in the new time scale, becomes

$$\frac{d\phi_k}{d\tau} = -\frac{\partial L}{\partial \phi_k} + \sum_{j=1}^{J} \mathcal{N}_j(\psi_j(\boldsymbol{\phi})) \frac{\partial \psi_j(\boldsymbol{\phi})}{\phi_k} + \sigma(\tau)\frac{dW(\tau)}{d\tau}, \qquad k = 1 \ldots K. \tag{30}$$

The overall time delay of the loop of Fig. 2 is dominated by the delay through the analog neural network, and by the multiplications for calculation of the gradient. Using Intel's ETANN chip, for a 4-layer network, implies a processing delay of $6\mu s$ for the neural network to operate. Including the multiplication by the covariance matrix and some overhead, the processing time delay τ_D of the system may be assumed to be $10\mu s$. It is therefore reasonable to set the capacitance values on the order of $10^{-5}F$ for operation under these circumstances.

Note that a practical implementation of this system could be significantly simplified from that shown in Fig. 2, by exploiting the fact that only one column of the term $\partial \mathbf{S}(\boldsymbol{\phi})/\partial \phi_k$ is non-zero. Also, since the matrix expressions are preceded by the $\mathrm{tr}(\cdot)$ and $\Re e(\cdot)$ operators, only a small portion of the total number of computations for evaluating the full matrix product of $\mathbf{P}^{\perp}\partial \mathbf{S}/\partial \phi_k (\mathbf{S}^{\dagger}\mathbf{S})^{-1}\mathbf{S}^{\dagger}$ and $\hat{\mathbf{R}}_N$ need be evaluated.

A significant difficulty with this proposed global optimization strategy for any application is the very slow convergence rate of the noise. The $\frac{1}{log(t)}$ decay rate (annealing schedule) of the noise process is slow to the point where convergence requires an unacceptably long period of time. From (16) we can see, assuming $c_* = 1$, even when $t = 10^4$, $\sigma(t)$ is approximately equal to 0.3, which is too large for the convergence of $\boldsymbol{\phi}$. Also, if we want $\sigma(t)$ to be 0.01 (assuming this is small enough for convergence), t must reach the value e^{10^4}, which is intolerable in real-time systems.

To ensure the practicality of simulated annealing, the annealing schedule must be accelerated. We propose the noise process $W(t)$ be turned off after a certain pre-determined time t_o. It is expected that within this period, the parameter values will have settled into the global

valley. After the disappearance of the noise, the process converges quickly to the nearest local minimum. The time t_o at which the noise is turned off is determined empirically. However, since $\boldsymbol{\phi}$ only converges globally as $t \to \infty$, there is now no guarantee the process will have converged to the *global* optimum, since the annealing schedule is now accelerated.

To overcome this difficulty, we present a statistical criterion which can be used for the DOA example, to test if global convergence has in fact occured. The strategy is to allow the circuit to follow the annealing schedule of (16) until time t_o, when the noise is turned off. The proposed criterion can then be applied. If the test succeeds, the converged result is accepted. If the test fails, the optimization process is re- initiated and the strategy repeated until the test succeeds.

The criterion to test for global convergence is presented with the aid of the following theorem. The proof is easy but is not shown:

Theorem 3 *For white noise, the value of the quantity $NL(\boldsymbol{\phi}_o)/\gamma^2$, where $\boldsymbol{\phi}_o$ is the true value of $\boldsymbol{\phi}$, is asymptotically distributed as a χ^2 random variable with $N(M-K)$ degrees of freedom.*

In practice, the global optimum $\boldsymbol{\phi}^* \neq \boldsymbol{\phi}_o$. However, under reasonable conditions, we may substitute $\boldsymbol{\phi}^*$ for $\boldsymbol{\phi}_o$, and the result used as the basis of a hypothesis test to check if $\boldsymbol{\phi}^*$ has globally converged. If

$$\frac{NL(\boldsymbol{\phi}^*)}{\gamma^2} < T(\alpha) \tag{31}$$

then the test succeeds. Otherwise, it has failed and the optimization process must be re-initiated. The quantity $T(\alpha)$ is the value above which a χ^2 random variable with $N(M-K)$ degrees of freedom is expected to fall 100α % of the time.

An alternate criterion which does not require knowledge of the noise variance γ^2 but is somewhat more computationally complex, is presented in [35]. For other applications of this proposed optimization process, similar criteria for testing global convergence could be conceived.

In this treatment, we have assumed the value of K is known. However, in practice, it is unknown and variable and must be estimated from the data. Previous methods [32] [34] exist for this purpose, but are computationally intensive and hence are not appropriate for our purposes. A method of reasonable computational complexity is now described. The proposal is to build a fixed system for the maximum number $\tilde{K} < M$ of signals which the system is likely to encounter. The system will then provide an estimate $\tilde{\boldsymbol{\phi}}$ consisting of $\tilde{K}$ values of $\boldsymbol{\phi}$. It is shown in [34] that in this situation, the MLE will provide K values of $\boldsymbol{\phi}$ which correspond to real incident signals, and $\tilde{K}-K$ values which are extraneous. The issue is thus to separate the desired values of $\boldsymbol{\phi}$ from the extraneous. The problem may be solved in the following way. An estimate $\hat{\tilde{\mathbf{R}}}_s \in \mathcal{C}^{\tilde{K}\times\tilde{K}}$ of the source covariance matrix $\tilde{\mathbf{R}}_s = E(\tilde{\mathbf{a}}_n\tilde{\mathbf{a}}_n)$

is given as

$$\tilde{\hat{\mathbf{R}}}_s = \frac{1}{N-K}\tilde{\mathbf{S}}^+(\tilde{\boldsymbol{\phi}})\hat{\mathbf{R}}_N\tilde{\mathbf{S}}^+(\tilde{\boldsymbol{\phi}})^\dagger \tag{32}$$

where $\tilde{\mathbf{S}}^+ = \left(\tilde{\mathbf{S}}^\dagger\tilde{\mathbf{S}}\right)^{-1}\tilde{\mathbf{S}}^\dagger$ is the pseudo-inverse of $\tilde{\mathbf{S}} \in \mathcal{C}^{M\times\tilde{K}} = [\mathbf{s}(\phi_1),\ldots,\mathbf{s}(\phi_{\tilde{K}})]$. The kth diagonal entry $\hat{\tilde{r}}_k$ of $\tilde{\hat{\mathbf{R}}}_s$ is the estimated power of the kth incident source. The kth diagonal entry e_k of the matrix $\mathbf{E}$ given by

$$\mathbf{E} = \left(\tilde{\mathbf{S}}^\dagger\tilde{\mathbf{S}}\right)^{-1}\gamma^2 \tag{33}$$

is the corresponding error or noise power component of this quantity e.g. [4]. The noise power γ^2 is assumed known with no uncertainty.

Now let $\mathcal{S}$ be the set containing the signal indeces which correspond to the extraneous signals. Then, the ratios

$$\frac{\hat{\tilde{r}}_k}{e_k}, \qquad k \in \mathcal{S} \tag{34}$$

are F–distributed with $(N-K,\infty)$ degrees of freedom. Therefore, the above ratios may be evaluated for $k = 1,\ldots,\tilde{K}$. The indeces which significantly exceed the corresponding F–statistic at an appropriate level of confidence correspond to true signals.

5 Demonstrations of Performance

In this section, we show some simulation results of the performance of the proposed optimization system obtained from the DOA example. We also present some brief discussion regarding various aspects of the results.

The DOA estimation environment which was simulated to test the proposed method is now described. A specific incident signal model and background noise field were constructed to simulate a realistic array signal processing situation. Our proposed circuit was then tested using this simulated data.

For all simulations presented the following parameters of the problem are fixed:

- the number of incident signals $K = 2$,
- the number of snapshots $N = 20$,
- the number of sensors $M = 6$ on a linear array geometry with uniform spacing,
- the components of the complex signal amplitude vector $\mathbf{a}_n$, in (21) are uncorrelated random variables with equal power,

- The true angles of arrival $\boldsymbol{\phi}_o = (0.25, -0.25)$ standard beamwidths, where 1 standard beamwidth equals $2\pi/M$ radians.

Note that standard beamwidths are used as a convenient measure of angle separation as they normalize simulation results over varying array size.

The background noise field (i.e., the term $\mathbf{v}_n$ in (21)) was constructed as spatially coloured, rather than white noise as indicated by the treatment of Sect. 3. The maximum likelihood DOA algorithm is somewhat sub-optimal in unknown coloured noise, but so are the majority of other such algorithms. The reason for using coloured noise instead of white is because the noise can be used to control various local minima in the objective function. This in turn gives us a better feel for the behaviour of the global optimization process.

If the background noise is spatially stationary but coloured, then the noise covariance matrix $\boldsymbol{\Gamma}$ is in general a Toeplitz structure instead of the form $\gamma^2\mathbf{I}$ as it is for white noise. The ij^{th} element γ_{ij} of $\boldsymbol{\Gamma}$ used in our simulations is that corresponding to a first-order complex autoregressive process and is given by

$$\gamma_{ij} = \mu^{|i-j|} \exp\left(j\nu^{|i-j|}\right). \tag{35}$$

The angular spectrum corresponding to (35) consists of a single peak positioned at ν radians, of height $\frac{1}{1-\mu}, \mu < 1$. Essentially, the greater the value μ less than one, the more concentrated is the noise energy around ν. For our simulations the value used for the noise decay, μ, is 0.99. This value of μ, although very high, was used to accentuate the noise as a local minima, and thus had value in illustrating the algorithm's properties of global convergence.

The value of the constant c_* used for these simulations was determined using (20), with $\Delta L = \mathrm{tr}\hat{\mathbf{R}}_N$ as suggested by (27).

Training the Neural Networks: In this example, the neural network was used only to implement the gradient function, and not the constraints. The feedforward model used is similiar to its usual manifestations. The final network layout had 3 input nodes, 2 hidden layers, each with 36 units, and an output layer of 18 units. In our realization no effort was made to reduce the complexity of the feedforward network by reducing the number of weights or processing nodes. The network inputs are real valued and the output is a complex vector. Therefore, the network is designed to incorporate complex weights.

The back propagation algorithm [27] was used to train the network. The learning rate, η, was initialised to 0.5 and decreased over time (by hand tuning) – this was found to result in reasonable convergence. Also, epoch learning was used: the error gradient is averaged over a number of patterns, here 30 patterns, before modifying the weights, to obtain a better estimate of the error gradient. In presenting the training data to the network we employed a "multiscale" approach. This implies that different sized training sets are used – each with an increasingly finer resolution of $\boldsymbol{\phi}$. The first training set had a resolution of 0.30 for each $\boldsymbol{\phi}_k$, then next 0.1 and finally 0.05. This is done to reduce convergence time while ensuring

reasonable accuracy. It was observed that convergence is excruciatingly slow if too large a dataset is presented initially. Heuristically, this approach first provides a general description of the desired input-output mapping. Once that is achieved, the role of the subsequent finer grained training data is to allow the network to converge to solutions with increasingly higher precision. Ultimately, the network was trained so that the output was accurate to two decimal digits of accuracy.

Note that the constraints of (28) are employed in creating the training set. This effectively reduces the size in the state space by a factor of $2^{-(K-1)}$, which in this case is one half. By eliminating redundancy and complexity in the range of input values the network need be presented with only a subset of input combinations. Minimizing the size of required training data decreases the computational effort required to achieve the desired input-output relation.

Simulation Results: The results presented in this section relating to the trajectories of $\boldsymbol{\phi}$ are obtained by solving (30) using a Runge-Kutta numerical technique. The intention of these simulations is to create an objective function that is computationally difficult from an optimization perspective, in terms of the number and value of local minima. To this end, we simulated the noise term $\mathbf{v}_n$ in (21) as highly correlated Gaussian noise corresponding to (35), with a value of $\nu = \frac{3}{\pi}$, and $\mu = 0.99$. The $K = 2$ true DOA angles were held fixed at $\boldsymbol{\phi} = [0.25, -0.25]$ standard beamwidths. We show results for an elemental SNR of 8 dB in Fig. 3 and 10 dB in Fig. 4. By "elemental SNR", we mean the ratio of the total signal power to the noise power at a single element of the array. For each SNR value, we show:

1. The surface plot of $-\log_{10}[L(\boldsymbol{\phi})]$. This surface plot is provided to give the reader a visualization of the objective function. Note that we have converted minima into maxima through the negative sign shown, in order to present a clearer pictorial image.

2. State space trajectory, starting from -2.0 and -2.5 standard beamwidths, over the corresponding contour plot of the objective function.

3. State space trajectory, starting from 2.5 and -2.5 standard beamwidths, over the corresponding contour plot of the objective function.

4. Time history equivalent of case 3, to show the progression of the state variables over time.

Figs. 3a and 3a show surface plots of $-\log_{10}[L(\boldsymbol{\phi})]$ which clearly reveal that multiple minima are present. Note the occurance of high "plateaus" that have multiple albeit poor local minima. As well, there is the presence of a series of "valleys", which include multiple minima close to and including the global minimum. We can observe the desired signal peak at $(\phi_1, \phi_2) = (.25, -.25)$ and significant undesired peaks due to the noise at $(\phi_1, \phi_2) = (0, 3/\pi)$ and $(\phi_1, \phi_2) = (3/\pi, 0)$, respectively. It is evident from the shape of these objective function plots that the maximum likelihood DOA algorithm is indeed an excellent example to illustrate the capabilities of this proposed optimization procedure.

Figs. 3b,c and 4b,c (cases 2 and 3 from above, respectively), show trajectory plots of the angle estimates from two different, very poor starting points. We note from these experiments the annealed diffusion mechanism consistently found the global optimum, regardless of the initial value of the process.

Examining the time trajectories of Fig. 3d and 4d, (case 4 above) we see that convergence to a stable set of estimates of $\boldsymbol{\phi}$ is achieved within a relatively few "time constants". We define time constant is this context as a unit of the variable $\tau = t/C_k$ in (30) (we assume all capacitance values are equal).

A series of experiments were conducted to empirically determine the time t_o at which the noise is to be turned off. Using the signal and noise model described, for an SNR of 8 and 10 dB, and for a wide range of initial values, the length of the run-time was increased until convergence into the global bowl was achieved in all cases. This resulted in a run-time of 150 time constants.

Recall that, for the ETANN chip, one time constant is about $10\mu s$. An additional 15 time constants for convergence to the global optimum, after the noise is removed, is necessary. Thus, a solution is attainable in about $1.65ms$. This performance is quite within the realms of real-time operation. Significantly improved performance would be achieved if a faster analog neural network were available.

In order to evaluate how the proposed method generalizes to larger values of M, we trained a network for the values $M = 20$ and $K = 3$, for which the system operated successfully. The same number of network layers were used, with each layer consisting of a larger number of units. The network eventually trained to provide the same order of error as in the previous case. Since this network involves the same number of layers, the time for a single loop iteration for the larger problem remains roughly unchanged. However, the number of iterations required for convergence can increase with problem size, so some increase in computational time is likely as the problem size grows.

Even though the trajectories may have visited outside the feasible region, it was noted that in all cases the final estimate obeyed the constraints of (28). Thus, we observe a verification of the effectiveness of the proposed constraint implementation procedure.

The performance of the proposed analog implementation of the MLE was compared, by simulations, to a conventional digital numerical implementation, programmed using the MATLAB Nelder-Meade simplex optimization algorithm. The criterion of performance used is the mean-squared error of the DOA estimates. At high SNR, the proposed analog implemetation did slightly worse (1-2 dB) than the digital, due to errors in the neural network approximation of the gradient. However, at lower values of SNR near the threshold, the analog version actually did better. This behaviour is attributed to the fact that the digital optimization routine occasionally falls into a local minimum point relatively far from the global minimum, resulting in high error. However, the analog version, by virtue of its global seeking capability, finds the global optimum with much higher probability.

6 Conclusions

We have presented a simple and effective scheme for multivariate real-time global optimization with constraints. The basic structure is a realization of the dynamics of optimization by *steepest descent*. The requisite computations for the objective function gradient and for the constraint function implementation are provided by an analog neural network, which is trained to approximate the desired expressions for any relevant input value. The constraints are implemented in a simple way by penalizing the gradient if the trajectories wander out of the feasible region.

Convergence to the global optimum is achieved by injecting a stochastic process into the differential equation governing the behaviour of steepest descent with constraints. The resulting noise source is controlled by an annealing schedule. Initially, the noise power is set sufficiently high to facilitate escape from any local minimum. The decay rate of the annealing schedule must be slow enough to allow the process to settle within, yet fast enough to prevent escape from, the global bowl. It is shown that the schedule given by (16) is sufficient to allow convergence to the global optimum in the weak sense, as $t \to \infty$.

Practically, this result in its original form is of little value. This is because the thrust of this paper is towards *real- time* systems, yet the proposed annealing schedule requires infinite time. A practical approach is to turn off the noise after an empirically-determined length of time, which is sufficient to yield global convergence with high probability. However, since the process may now fail to converge globally, a criterion which is dependent on the application must be implemented which can test for global convergence. If the test fails, the diffusion process must be re-initiated and the test re-applied.

The performance of the proposed optimization process is demonstrated by the direction of arrival (DOA) estimation problem of plane waves incident onto arrays of sensors. The specific algorithm used is the maximum likelihood (ML) algorithm for this purpose. A statistical criterion for testing global convergence was proposed. Demonstrations showing the global convergence characteristic of the method were provided for this specific example. It was also noted that the proposed technique for incorporating constraints operated effectively. The practical embodiment of this proposed optimization technique is greatly aided by the existence of commercial analog chips, e.g., the ETANN circuit [14]. It was shown that real-time performance is indeed attainable with the maximum likelihood DOA problem, using the ETANN chip.

It is expected this proposed optimization structure would be effective in a large number of control and signal procesing problems. Specifically however, the array signal processing community has long been plagued with the difficulties of real- time implementation of the more modern high-resolution DOA estimation algorithms. It is therefore expected this proposed technique should at least have significant impact in this specific area.

7 Acknowledgements

The authors wish to thank Mr. Jian Kan Yang for useful input. Also, the financial assistance of the following agencies are gratefully acknowledged: the Telecommunications Research Institute of Ontario (TRIO), the Natural Sciences and Engineering Research Council of Canada (NSERC), and COM DEV Ltd., Cambridge, Ont.

References

[1] L.O. Chua and G.-N. Lin, "Nonlinear programming without computation," *IEEE Trans. Circuits Syst.*, Vol. 31, pp. 182-188, Feb. 1984.

[2] –, "Errata to 'Nonlinear programming without computation'," *IEEE Trans. Circuits Syst.*, Vol. 32, pp. 736, July 1985.

[3] T.S. Chiang, C.R. Hwang and S.J. Sheu, "Diffusions for global optimisation in $\mathcal{R}^n$," *SIAM J. Control and Optimization*, Vol. 25, pp. 737-753, May 1987.

[4] N. Draper, H. Smith "Applied Regression Analysis", 2nd. Ed., J. Wiley and Sons, Inc.

[5] El Gamal, "Using simulated annealing to design good codes", *IEEE Trans. Info Theory*, Vol. IT-33, pp. 116- 123, Jan. 1987.

[6] T. C. Gard, "Introduction to stochastic differential equations", Marcel Dekker, Inc., 1988.

[7] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images", *IEEE Trans. Pattern analysis and machine intelligence* Vol. PAMI-6, pp. 721-741, Nov. 1984.

[8] S. Geman and C.R. Hwang, "Diffusions for global optimisation ," *SIAM J. Control and Optimization*, Vol. 24, pp. 1031-1043, Sept. 1986.

[9] P.E. Gill, W. Murray and M.H. Wright, "Practical Optimization", Academic Press, 1981

[10] G.H. Golub and V. Pereyra, "The differentiation of pseudoinverses and nonlinear least squares problems whose variables separate", *SIAM J. Numer. Anal.*, Vol. 10, No. 2, pp. 413-432, Apr. 1973.

[11] G.H. Golub and C.F. Van Loan, *Matrix Computations*, The John Hopkins University Press, 1983.

[12] D. Goryn and M. Kaveh, "Direction finding networks based on the approximate maximum likelihood and covariance fit formulations", *Proc. ICASSP91*, May 14-17, 1991, Toronto, Canada, pp. 3045-3048.

[13] J.J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons", *Proc. Natl. Acad Sci.*, Vol. 81, pp. 3088-3092, 1984.

[14] Intel Report *Electrically trainable analog neural network*, 1990.

[15] T.M Jelonek "Maximum likelihood signal estimation using continuous-time connectionist networks", M.Eng Thesis, Dept. of Elec. and Comp. Eng., McMaster University, Hamilton, Ont. Canada, Aug., 1990.

[16] T.M. Jelonek and J.P. Reilly, "Maximum likelihood estimation for direction of arrival using a nonlinear optimising neural network," *International Joint Conference on Neural Networks*, San Diego, Vol. I, pp. 253-259, June 1990.

[17] M.P. Kennedy and L.O. Chua, "Neural networks for nonlinear programming ," *IEEE Trans. Circuits Syst.*, Vol. 35, pp. 554-562, May 1988.

[18] Hassan K. Khalil, "Nonlinear Systems", Macmillan Publishing Company, New York. 1992.

[19] S. Kirkpatrick, C.D. Gellat, Jr. and M.P. Vecchi,"Optimization by simulated annealing,"*Science*, Vol. 220, pp. 621-680, 1983.

[20] R. Hecht-Nielson, "Theory on the back-propagation neural network", *Proc. IEEE Int'l Joint Conf. on Neural Networks*, Washington, D.C., pp. 593-606, June, 1989.

[21] C.R.K. Marrian and M.C. Peckerar, "Electronic neural net algorithm for maximum entropy solutions of ill-posed problems", *IEEE Trans. Circuits and Systems*, Vol. 36, No. 2, pp. 288-294, Feb. 1989.

[22] J.P. Reilly and S. Haykin, "Maximum likelihood receiver for low-angle tracking radar," Part 2, *Proc. IEE*, Vol. 129, Pt. F, pp. 331-340, 1982.

[23] J.P. Reilly, Q. Wu, T. Jelonek and J.K. Yang, "Real-time analog implementation of the maximum-likelihood direction-of-arrival estimation algorithm", submitted to Proc. IEE May, 1994.

[24] A. Rodriguez-Vazquez, R. Dominguez-Castro, *et al*, "Nonlinear Switched-Capacitor Neural Networks for Optimization Problems", *IEEE Trans. Circuits and Systems*, Vol. 37, No. 3, pp. 384-398, March, 1990.

[25] R.Roy and T. Kailath, "ESPRIT - Estimation of signal parameters via rotational invariance techniques", *IEEE Trans ASSP*, Vol. ASSP-37, No. 7, pp. 984-995, July 1989.

[26] G. Royer "A remark on simulated annealing of diffusions processes," *SIAM J. Control and Optimization.*, Vol. 27, pp. 1403-1408, Sept. 1989.

[27] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, D.E. Rumelhart and J.L. McClelland, eds., Vol. 1, pp 318-362, MIT Press, Cambridge, 1986.

[28] R.O. Schmidt, "Multiplie emmitter location and parameter estimation", *IEEE Trans. Antennas Propag.*, vol. AP-34, No.3, pp. 276-280, Mar. 1986.

[29] K.C. Sharman, "Maximum likelihood parameter estimation by simulated annealing", *Proc. IEEE ICASSP'92*, San Francisco, March, 1992.

[30] M. Viberg and B. Ottersten, "Sensor array processing based on subspace fitting" *IEEE Trans. Signal Processing*, Vol. 39, pp. 1110-1121, May, 1991.

[31] M. Viberg, B. Ottersten, and T. Kailath, "Detection and estimation in sensor arrays using weighted subspace fitting" *IEEE Trans. Signal Processing*, Vol. 39, pp. 2436-2449, Nov. 1991.

[32] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria", *IEEE Trans. ASSP*, Vol. 33, pp. 387-392, Apr. 1985.

[33] K.M. Wong, J.P. Reilly, Q. Wu, and S. Qiao, "Estimation of the direction of signals in unknown correlated noise, Part I: The MAP approach and its implementation *IEEE Trans. SP*, Vol. 40, No. 8, pp. 2007-2017, Aug. 1992.

[34] Q. Wu and D.R. Fuhrmann, "A Parametric Method for Determining the Number of Signals in Narrow-Band Direction Finding", *IEEE Trans Signal Processing* Vol. 39, No. 8, pp. 1848-1857, Aug. 1991.

[35] J.K. Yang, " Estimation of Calibration Errors in array signal processing" M.Eng. thesis, Dept. of Elec. and Comp. Eng., McMaster University, Hamilton, Ont., Canada, Aug., 1992.

[36] S. Zhang and A.G. Constantinides, "Lagrange Programming Neural Networks", *IEEE Trans. Circuits and Systems-II: Analog and Digital Processing*, Vol. 39, No. 7, pp. 441-452, July, 1992.

[37] I. Ziskind and M. Wax, "Maximum likelihood localization by multiple sources by alternating projection," *IEEE Trans. ASSP*, Vol. 36, no. 10, pp. 1553-1560, Oct., 1988.