Progress In Electromagnetics Research, PIER 25, 53–76, 2000

EFFICIENT FAULTY ELEMENT DIAGNOSTICS OF LARGE ANTENNA ARRAYS BY DISCRETE MEAN FIELD NEURAL NETS

G. Castaldi, V. Pierro, and I. M. Pinto

The Waves Group
D.I.I.I.E., University of Salerno
via Ponte don melillo
I-84084 Fisciano (SA), Italy

1. INTRODUCTION

The identification of faulty elements in *large* (hundreds to thousands of elements) antenna arrays, e.g., radiotelescopes, from (complex, amplitude and phase) field or (real) intensity measurements is a problem of considerable practical and theoretical relevance [1, 2].

In the simplest case, where only two states are allowed (element *on* or *off*), the state of a faulty array can be represented by a point in the lattice $\{0,1\}^N$, where N is the number of radiators, and the state "0" and "1" denote a faulty and a working antenna, respectively.

Identifying the (unknown) state of the array amounts to solving a (factorially complex, and non convex in general) minimization problem

over the set of all 2^N possible array states, where the functional to be minimized measures the distance between two M-dimensional vectors, representing the measured data, and the fields (intensities) produced by the (unknown) state of the array, respectively, at M given measurement points. Depending on whether intensity (easier to perform) or field measurements are available, the related minimization problem will be linear and nonlinear, respectively. For the linear (field measurements) case, std. (e.g., Gauss-Seidel [3]) methods would be plagued by ill-conditioning especially as the distance between the source and the measurement plane is increased [1]. For the nonlinear (intensity measurements) case, naive conjugate-gradient (henceforth CG) minimization algorithms would be prone to trapping by *local* minima, in view of the non-convex character of the problem [2]. On the other hand, global no-derivative minimum-seeking algorithms (e.g., Rosenbröck [4]) might be quite slow.

Due to these inherent limitations, during the last few years, attention has been focused on a number of hopefully more effective tools for (global) minimization problems where many local minima do exist, including both stochastic (genetic algorithms [5], simulated annealing [6], Boltzmann machines [6]) and deterministic (mean field neural networks [7]) methods.

In this paper we adopt, for the first time, to the best of our knowledge, a discrete (asynchronous) version of Vidyasagar [8] (originally continuous) mean field neural network, to attack the (binary) array faulty element identification problem. Vidyasagar net can be regarded as a mean-field version of Boltzmann machine. We show that the net (as well as its discrete version used here) is totally stable, evolving to the stationary points of a (weak) Lyapounov function, but for a set of initial conditions of measure zero. We further prove that under suitable (sufficient) assumptions for the involved threshold function, the above Lyapounov function has a unique minimum, which coincides with the sought global minimum of the functional to be minimized on the lattice $\{0,1\}^N$. A number of critical technical issues (computation of the multilinear energy polynomial [9], optimum choice of the weights, and of the updating schedule of the threshold parameter [10]) are discussed.

The performance of the algorithm is investigated for various values of the ratio d/D, d being the distance between the source and the measurement plane, and D the array and measurement domain diam-

eter. It is shown that a judicious choice of the weights leads in a few steps to the correct solution with probability close to one as $d/D \to 0$. The algorithm is shown to be at least as robust as the smartest local-trap-avoiding CG algorithms [11] w.r.t both additive (gaussian) noise in the measurements, and random errors in the assumed radiators' positions, while being overall considerably faster and more reliable.

The remaining of the paper is organized as follows. In Section 2 the minimization problem is set up. In Section 3 the (continuous and discrete asynchronous) Vidyasagar network is introduced, and its convergency properties are discussed. In Section 4 a discrete asynchronous implementation of the proposed net is proposed. In Section 5 a number of technical implementation issues are highlighted. Numerical experiments are presented in Section 6. Conclusions follow under Section 7. In Appendix A and B optimum updating schedules for the neural threshold parameter are discussed in detail.

2. ANTENNA ARRAY DIAGNOSTICS FROM LINEAR AND NON-LINEAR MEASUREMENTS

The following functionals are to be minimized according to whether field or intensity measurements are available:

$$f(\vec{s}) = \sum_{r=1}^{M} \alpha_r \left| \left[\sum_{i=1}^{N} \vec{E}_i(\vec{x}_r) s_i \right] - \vec{\mathcal{E}}(\vec{x}_r) \right|^2 , \tag{1}$$

$$f(\vec{s}) = \sum_{r=1}^{M} \alpha_r \left\{ \sum_{i=1,j=1}^{N,N} \vec{E}_i(\vec{x}_r) \cdot \vec{E}_j^*(\vec{x}_r) s_i s_j - \mathcal{I}(\vec{x}_r) \right\}^2 , \tag{2}$$

where $\vec{s} = \{s_1, \ldots, s_N\} \in \{0,1\}^N$ represents the (unknown) state of the array; $\{\vec{x}_r\}_{r=1}^M$ is the set of measurement positions, $\vec{\mathcal{E}}(\vec{x}_r)$ are the (complex, frequency domain) measured field values, $\mathcal{I}(\vec{x}_r)$ are the (real) measured field-intensity values, and $\vec{E}_i(\vec{x}_r)$ are the (complex, frequency domain) fields produced by the *i-th* radiator. The α_r's denote the relative weights of the *r-th* measurement, and $*$ denotes complex conjugation.

By noting that if $\vec{s} \in \{0,1\}^N$ then $s_i = s_i^q$, $\forall q \in \aleph$, the functions $f(\vec{s})$ in (1), (2) can be rewritten as multilinear ($2nd$ and $4th$ order, respectively) polynomials in $s_1, \ldots, s_N$ taking on the same values as

$f(\vec{s})$ on the points of the lattice $\{0,1\}^N$, viz.:

$$H(\vec{s}) = \sum_{p=1}^{Q} \sum_{i_{1,\ldots,1_p}}^{1,\ldots,N} W^{(p)}_{i_{1,\ldots,i_p}} s_{i_1} s_{i_2} \cdot \ldots \cdot s_{i_p}, \tag{3}$$

where $Q = 2$ or 4 according to whether one refers to (1) or (2). These polynomials will be henceforth denoted as $H(\vec{s})$[1]. As seen from (1), (2), the polynomial $H(\vec{s})$, can be written as follows:

$$H(\vec{s}) = H_0(\vec{s}) + H_m(\vec{s}) \tag{4}$$

where only $H_m(\cdot)$ depends on the measurements, and is a (homogeneous) function of degree one or two, in $\vec{s}$, according to whether one refers to (1) or (2). A similar splitting obviously applies to the $W^{(n)}$ matrices defined in (3) as well, viz.:

$$W^{(n)} = W_0^{(n)} + W_m^{(n)}. \tag{5}$$

In order to find the minimum of $f(s)$ in $\{0,1\}^N$, one can use the simulated annealing technique [6], whereby to each transition between nodes of $\{0,1\}^N$ one associates a (Boltzmann) probability, depending on the associated variation of H, and a control parameter β^{-1} (the so called *temperature*) which is decreased during the process according to a suitable schedule. An alternative implementation of simulated annealing is the stochastic Boltzmann-machine neural-network [7]. Simulated annealing, however, can be pretty slow, the associated computational burden being exponential with the problem size. In order to speed-up the computation, it has been suggested to evolve the statistical average of the state, instead of the state itself, computed Boltzmann or some other likely suitable distribution law. The resulting mean-field neural-network can be quite faster. Loosely speaking, the above averaging procedure wipes out the many local (irrelevant) minima, leaving *only one* fixed point, which hopefully corresponds to the sought *global* minimum of $f(s)$ on the state lattice[2].

[1] A constructive procedure for computing this latter is given in [9].

[2] It should be noted that while the annealing procedure and its sto-chastic Boltzmann-machine implementation can be proven to converge with probability one to the global minimum in the limit of an infinite number of transitions, provided the temperature is decreased no faster than logarithmically [6], a similar convergency theorem for the mean-field neural network does not exist, to the best of our knowledge. Convergency (in probability) is however observed in a wide variety of applications [8], including the examples discussed below

3. NEURAL NETWORK PROPERTIES

The non-linear mean field neural network, introduced by M. Vidyasagar [8] and used in this paper is described by:

$$\begin{cases} \dfrac{du_i}{dt} + u_i = -\dfrac{\partial H}{\partial s_i}, \\ s_i = g(\beta u_i), \quad i = 1, \ldots, N, \end{cases} \tag{6}$$

where $H(\cdot)$ is the multilinear polynomial defined above, $\vec{u}$ and $\vec{s}$ denote the potential and the state of the net, respectively[3], $g(\cdot)$ is a strictly increasing C^∞ threshold function such that:

$$\lim_{x \to -\infty} g(x) = 0, \quad \lim_{x \to +\infty} g(x) = 1, \quad g(0) = g'(0) = .5, \quad g'(x) > 0, \tag{7}$$

and β a (real, non-negative) parameter. In view of (7), one has in general $\vec{s} \in [0,1]^N$, whereas for any *finite* $\beta, \vec{s} \in [0,1]^N$.

The main abstract properties of (6) can be established as follows.

Lemma I - All solutions of (6) are contained in a sphere of $\mathcal{R}^N$ with radius $R < \infty$.

Proof - This follows from a theorem of Cohen and Grossberg [12], whose assumptions are verified *ad abundantiam* in view of the properties of $H(\cdot)$ and $g(\cdot)$. •

Lemma II - The function

$$\Lambda(\vec{s}) = H(\vec{s}) + \sum_{i=1}^{N} \beta^{-1} \int_0^{s_i} g^{-1}(v) dv, \quad \vec{s} \in U_N = [0,1]^N \tag{8}$$

is a (weak) Lyapounov function.

Proof - The function $\Lambda(\vec{s})$ as defined by (8) is continuous and defined in a bounded set, in view of Lemma I. Accordingly, its level-sets

$$\{\vec{s} \in U_N | \Lambda(\vec{s}) \leq c\}, \tag{9}$$

[3] Note that the s_i in (6) can take *any* value in $[0,1]$.

are compact $\forall c \in \mathcal{R}$. Further,

$$\frac{d\Lambda(\vec{s})}{dt} = -\beta^{-1} \sum_{i=1}^{N} \frac{d}{ds_i}[g^{-1}(s_i)] \left(\frac{ds_i}{dt}\right)^2 \leq 0, \tag{10}$$

as seen from (6), since $g(\cdot)$ is an increasing function. •

Lemma III - The minima of $\Lambda(\vec{s})$ in U_N coincide with those of $H(\vec{s})$ in the limit $\beta \to \infty$.

Proof - This follows immediately from (8). •

Lemma IV - The minima of $H(\vec{s})$ in U_N are located on $\{0,1\}^N$, where, by construction $H(\vec{s}) = f(\vec{s})$.

Proof - This follows trivially from Weierstrass theorem, since the function: $H(\cdot)$ is linear in s_i $\forall i$ and hence its second derivative w.r.t s_i vanishes identically. •

Lemma V - The stationary points of Λ where $\nabla\Lambda = 0$ form an invariant set Z, i.e., every solution of (6) with initial value $\vec{u}_0 \in Z$ remains in Z, $\forall t$.

Proof - This follows from the fact that the stationary points of Λ are the (only) fixed points of (6), where:

$$s_i - g\left(-\beta \frac{\partial H}{\partial s_i}\right) = 0, \qquad i = 1, 2, \ldots, N. \bullet \tag{11}$$

Lemma VI - The Ω-limit set[4] of (6) is a subset of the invariant set Z referred in Lemma V.

Proof - This follows from LaSalle theorem [13] in view of the (already verified) assumptions that *i)* all solutions of (6) remain bounded

[4] We recall that [14] the Ω-limit set is the collection of ω-limit points of (6). These latter are those points $\vec{u}_\omega \in \mathcal{R}^N$ such that $\exists$ a solution $\vec{u}(t)$ of (6) and a sequence $\{t_k\}_{k=1}^{\infty}$ such that

$$\lim_{k\to\infty} t_k = +\infty, \qquad \lim_{k\to\infty} \vec{u}(t_k) = \vec{u}_\omega.$$

(Lemma I), and *ii)* a (weak) Lyapounov function exists (Lemma II) for (6). •

As a consequence of the above lemmas, we can conclude that:

Theorem I - The solutions of (6) converge in the limit as $t \to \infty$ to the minima of Λ in U_N, but for a set of initial conditions of measure zero. The minima of Λ in U_N in the limit as $\beta \to \infty$, are located on $\{0,1\}^N$, (and are thus isolated) and coincide with the minima[5] of $H(\vec{s})$ and $f(\vec{s})$.

Proof - The first statement follows immediately from Lemma VI. The second one is a direct consequence of Lemmas III and IV. •

We are now in a position to show that, under suitable assumptions for $g(\cdot)$, $\Lambda(\cdot)$ has a *unique* stationary point. To this end we need a number of preliminary results.

Definition I - We define the Leray-Schauder index of a function $F \in C^1 : \overline{A} \subset \mathcal{C}^N \to \mathcal{C}^N$ as follows:

$$i_{LS}(F, A) := \sum_{k=1}^{M} \mathrm{sgn}\{\det\left[\mathcal{I} - \nabla F\right]_{\vec{s}=\vec{s}_k}\}, \tag{12}$$

where $\{\vec{s}_k\}_{k=1}^{M}$ are the fixed points of F in A, i.e., the (regular[6]) solutions of:

$$\vec{s}_k - F(\vec{s}_k) = 0, \qquad k = 1, 2, \ldots, M. \bullet \tag{13}$$

Lemma VII (Leray-Schauder global continuity principle [15]) - Let $G : [\beta_1, \beta_2] \times \vec{A} \subset \mathcal{R} \times \mathcal{C}^N \to X \subset \mathcal{C}^N$ a compact operator, and A a bounded open set in a complex Banach space. Then, if *i)* $\vec{s} - G(\beta, \vec{s}) = 0$ has *no* solution in $[\beta_1, \beta_2] \times \partial A$, and *ii)* $i_{LS}[G(\beta_1, \cdot), A] \neq 0$, then the Leray-Schauder index $i_{LS}[G(\beta, \cdot), A]$ is constant in $[\beta_1, \beta_2]$.

---

5 Following [8] a point $\vec{s}_0 \in \{0,1\}^N$ is called a *local* minimum of $f(\vec{s})$ iff

$$f(\vec{s}_0) \leq f(\vec{s}), \quad \forall \vec{s} \in \mathcal{I}(\vec{s}_0)$$

where $\mathcal{I}(\vec{s}_0) = \{\vec{s} \mid D_H(\vec{s}, \vec{s}_0) = 1\}$, D_H being the Hamming distance [8].

6 These are the solutions of (13) where $\det[\mathcal{I} - \nabla F] \neq 0$.

Proof - The proof of this lemma can be found, e.g., in [15]. •

Lemma VIII - The Leray-Schauder index of the operator:

$$\vec{F}(\beta, \vec{s}) = \left\{ g\left(-\beta \frac{\partial H}{\partial s_i} \right) \right\}_{i=1}^{N}, \tag{14}$$

in $A : [0,1]^N \subseteq A \subset \mathcal{C}^N$ is equal to one.

Proof - By Definition I, one has:

$$i_{LS}[\vec{F}(0, \vec{s}), A] = \sum_{i=1}^{M} \operatorname{sgn} \left\{ \det \left[\mathcal{I} - \nabla_{\vec{s}} \vec{F}(0, \vec{s}) \right]_{\vec{s}=\vec{\sigma}_i} \right\} \tag{15}$$

where $\{\vec{\sigma}_i\}_{i=1}^{M}$ are the fixed points of F in A. However, for $\beta = 0$ there is only one fixed point of F, given by $\vec{\sigma} = 0.5\hat{u}$, where $\hat{u}$ is the N-dimensional vector with unitary components. Hence:

$$i_{LS} \left[\vec{F}(0, \vec{s}), A \right] = \operatorname{sgn} \left\{ \det \left[\mathcal{I} - \nabla_{\vec{s}} \vec{F}(0, \vec{s}) \right]_{\vec{s}=0.5\hat{u}} \right\} \tag{16}$$

On the other hand, $\nabla_{\vec{s}} \vec{F}(0, \vec{s})$ vanishes for $\beta = 0$. Hence:

$$i_{LS}[\vec{F}(0, \vec{s}), A] = \operatorname{sgn} \{\det[\mathcal{I}]\} = 1. \bullet \tag{17}$$

Lemma IX - If for a suitable choice of $g(\cdot)$, $\exists$ an open bounded set $A : [0,1]^N \subseteq A \subset \mathcal{C}^N$ such that $\forall \vec{s} \in \partial A$, $\forall \beta \in \mathcal{R}^+$, $\vec{F}(\beta, \vec{s}) \neq \vec{s}$, then $i_{LS}[\vec{F}(\beta, \vec{s}), A] = i_{LS}[\vec{F}(0, \vec{s}), A] = 1$.

Proof - This follows immediately from Lemma VII. •

Under the same assumptions of Lemma IX we can prove the following two Lemmas:

Lemma X - The number of fixed points of $\vec{F}$ in an open bounded set $A : [0,1]^N \subseteq A \subset \mathcal{C}^N$ is equal to one.

Proof - In view of the properties of $g(\cdot)$ and $H(\cdot)$, the operator $\vec{F}$ is analytic and compact in $A : [0,1]^N \subseteq A \subset \mathcal{C}^N$. Furthermore, by the hypotheses of Lemma IX, it has no fixed points in $\partial A, \forall \beta \in \mathcal{R}^+$.

Then, By Cronin-Schwartz theorem [15] the number of fixed points of $\vec{F}$ in A cannot exceed the Leray-Schauder index $i_{LS}[\vec{F}(\beta, \vec{s}), A] = 1$. On the other hand, $\vec{F}$ has at least one fixed point in A, and hence it has exactly one fixed point. •

Lemma XI - The Lyapounov function $\Lambda(\cdot)$ has only one stationary point.

Proof - This follows immediately from Lemma V. •

The above Lemmas, in turn, can be used to prove that

Theorem II - In the limit as $\beta \to \infty$ the above unique minimum of $\Lambda(\cdot)$ coincides with the global minimum of $f(\cdot)$ in $\{0,1\}^N$.

Proof - Let $\vec{s}_0$ the unique minimum of Λ, which, in view of Lemma V will be located on $\{0,1\}^N$, as $\beta \to \infty$. In view of (8) one has:

$$\lim_{\beta \to \infty} \Lambda(\vec{s}_0) = f(\vec{s}_0). \tag{18}$$

Now we proceed *per absurdum*, and assume that $\exists\, \vec{s}\,' \neq \vec{s}_0$ in $\{0,1\}^N$, and is the global minimum of $f(\cdot)$, so that:

$$f(\vec{s}_0) > f(\vec{s}'). \tag{19}$$

However, in view of (8), one has also

$$\lim_{\beta \to \infty} \Lambda(\vec{s}\,') = f(\vec{s}\,') \Longrightarrow \lim_{\beta \to \infty} \Lambda(\vec{s}_0) > \lim_{\beta \to \infty} \Lambda(\vec{s}\,'), \tag{20}$$

which contradicts the hypotheses. •

Note that the conditions on $g(\cdot)$ in Lemma IX are only *sufficient*, and might be by far too restrictive. The corresponding *necessary* conditions are not known. As a matter of fact, *no* constructive procedure is available to find out/construct such a function. On the other hand, all smooth step-like functions verifying (7) are natural candidates[7]. In practice, for any possible choice of $g(\cdot)$ a critical value β^* is observed such that for $\beta > \beta^*$ convergency of the algorithm described in Section

[7] These later include, e.g., $g(v) = 0.5[1 + \mathrm{erf}(v)]$ and $g(v) = .5[1 + \tanh(v)]$, as well as many cumulative distribution functions.

3 breaks down. Better choices of $g(\cdot)$ correspond to larger values of β^*. Numerical experiments collected in Section 6 were obtained using $g(v) = .5 \cdot [1 + \tanh(v)]$. We are currently investigating alternative, hopefully better, choices.

4. DISCRETE IMPLEMENTATION

The possibly simplest discrete numerical implementation of (6) is:

$$\begin{cases} u_i^{(n+1)} = (1-\Delta)u_i^{(n)} - \Delta \left(\dfrac{\partial H}{\partial s_i}\right)_{\vec{s}=\vec{s}^{(n)}}, \\ s_i^{(n)} = g[\beta u_i^{(n)}], \quad i = 1, \ldots, N; \quad n = 1, 2, \ldots \end{cases} \tag{21}$$

The above discretization is *consistent* in the sense that one can immediately prove that *any* stationary point of (21) is also a fixed point of (6) and vice-versa. We shall now sketch the proof of the discrete versions of Lemmas I and II.

Lemma XII - The trajectories of the discrete neural network (21) are bounded.

Proof - Let $r^{(n)} = |u^{(n)}|^2$. Then, the gradient of H being bounded by M,

$$r^{(n+1)} - r^{(n)} \leq \sum_{i=1}^{N} \Delta(2-\Delta)[u_i^{(n)}]^2 + 2M\Delta|1-\Delta||u_i^{(n)}| + M^2\Delta^2. \tag{22}$$

If $0 < \Delta < 2$, all solutions are contained in a ball of radius $2\Delta(2-\Delta)^{-1}M$, the l.h.s. of (22) being negative outside. •

We are now in a position to prove that:

Lemma XIII - The asynchronous implementation of (21) allows to defined a (weak) Lyapounov function.

Proof - We define a weak Lyapunov's function Λ by analogy with the continuous case. In order to prove that the variation of Λ after updating the *i-th* state only is negative, we note that:

$$\Lambda(\vec{s}^{(n+1)}) - \Lambda(\vec{s}^{(n)}) = \left(\frac{\partial H}{\partial s_i}\right)_{\vec{s}=\vec{s}^{(n)}} \left(s_i^{(n+1)} - s_i^{(n)}\right) + \frac{1}{\beta}\int_{s_i^{(n)}}^{s_i^{(n+1)}} g^{-1}(s)ds. \tag{23}$$

From Bolzano theorem[8], $\exists t \in [0,1]$:

$$\int_{s_i^{(n)}}^{s_i^{(n+1)}} g^{-1}(s)ds = (1-t)\left(s_i^{(n+1)} - s_i^{(n)}\right) g^{-1}(s_i^{(n)}) + t\left(s_i^{(n+1)} - s_i^{(n)}\right) g^{-1}(s_i^{(n+1)}), \tag{24}$$

whence:

$$\begin{aligned} &\Lambda(\vec{s}^{(n+1)}) - \Lambda(\vec{s}^{(n)}) \\ &\quad = \left(t - \Delta^{-1}\right)\beta^{-1} \cdot \left\{g^{-1}(s_i^{(n+1)}) - g^{-1}(s_i^{(n)})\right\}\left(s_i^{(n+1)} - s_i^{(n)}\right). \end{aligned} \tag{25}$$

In view of the monotonic nature of $g(\cdot)$, the Lyapunov function decreases iff $t - \Delta^{-1} \leq 0$, i.e., iff $\Delta \in (0,1)$. Furthermore, (25) implies:

$$\begin{aligned} &-\Delta^{-1}\beta^{-1}\left\{g^{-1}(s_i^{(n+1)}) - g^{-1}(s_i^{(n)})\right\}\left(s_i^{(n+1)} - s_i^{(n)}\right) \\ &\quad \leq \Lambda(\vec{s}^{(n+1)}) - \Lambda(\vec{s}^{(n)}) \\ &\quad \leq \frac{\Delta - 1}{\Delta}\beta^{-1}\left\{g^{-1}(s_i^{(n+1)}) - g^{-1}(s_i^{(n)})\right\}\left(s_i^{(n+1)} - s_i^{(n)}\right). \bullet \end{aligned} \tag{26}$$

Lemmas III to VI are obviously still valid, and thus a discrete version of Theorem I follows. On the other hand, Lemmas VII to XI are still valid, and Theorem II follows.

5. IMPLEMENTATION HINTS

In the following we touch briefly the problem of the choice of the parameters β and $\{\alpha_r\}_{r=1}^{M}$. The general problem of determining the transition range $(\beta_{min}, \beta_{max})$, which corresponds to the onset and completion of the migration of the sought fixed point of (6) from $s_i \approx 0.5$ to

[8] We use also the more or less obvious result that for any $\phi : \mathcal{R} \to \mathcal{R}$ continuous and non-decreasing one has:

$$\phi(s_i)(s_{i+1} - s_i) \leq \int_{s_i}^{s_{i+1}} \phi(s)ds \leq \phi(s_{i+1})(s_{i+1} - s_i),$$

which follows trivially from the hypothesis using the integral mean value theorem.

$\{0,1\}^N$ has been discussed in [10]. For the linear (field measurement) case, we let (see Appendix A for details):

$$\beta_{min} = 2\left\{\max_i\left(\sum_{j=1}^{N}\left|\frac{\partial^2 H}{\partial s_i \partial s_j}\right|\right)\right\}^{-1}, \tag{27}$$

$$\beta_{med} = 2\left\{\max_{i\neq j}\left(\left|\frac{\partial^2 H}{\partial s_i \partial s_j}\right|\right)\right\}^{-1}, \tag{28}$$

$$\beta_{max} = 2\left\{\min_{i\neq j}\left(\left|\frac{\partial^2 H}{\partial s_i \partial s_j}\right|\right)\right\}^{-1}. \tag{29}$$

The computationally cheapest strategy was found to be that of evolving the system according to the rule:

$$\beta_i = \beta_{min} + (i-1)(\beta_{med} - \beta_{min}), \tag{30}$$

starting from $s_k = .5$ at $i = 1$, and iterating the procedure until all states are saturated. Note that this usually occurs far before β_{max} is reached.

We adopted a similar schedule also for the nonlinear case, where the Hessian components depend on $\vec{s}$, by letting (see Appendix B):

$$\beta_{min} = 2\left\{\max_i\left(\sum_{j=1}^{N}\left|\frac{\partial^2 H}{\partial s_i \partial s_j}\right|\right)\right\}^{-1}_{\vec{s}=0.5\hat{u}} \tag{31}$$

$$\beta_{med} = 2\left\{c\,\max_{i\neq j}\left(\left|\frac{\partial^2 H}{\partial s_i \partial s_j}\right|\right)\right\}^{-1}_{\vec{s}=\vec{0}}. \tag{32}$$

$$\beta_{max} = 2\left\{c\,\min_{i\neq j}\left(\left|\frac{\partial^2 H}{\partial s_i \partial s_j}\right|\right)\right\}^{-1}_{\vec{s}=\vec{0}}, \tag{33}$$

where:

$$c = \left\{\frac{D[g^{-1}(s^3)]}{D[g^{-1}(s)]}\right\}_{s_q=1/2}, \tag{34}$$

$D[\cdot]$ being the derivative operator.

A further degree of freedom in the algorithm is related to the choice of the weight factors $\{\alpha_r\}_{r=1}^{M}$ in (1), (2). Two main alternative philosophies can be envisaged, namely measurement normalization, viz.:

$$\alpha_r = |\vec{\mathcal{E}}(\vec{x}_r)|^{-2}, \qquad r = 1,2,\ldots,M, \tag{35}$$

$$\alpha_r = \mathcal{I}(\vec{x}_r)^{-2}, \qquad r = 1,2,\ldots,M, \tag{36}$$

(to be used in (1) and (2) respectively) and operator normalization, viz.:

$$\alpha_r = \left[\sum_{i=1}^{N} |\vec{E}_i(\vec{x}_r)|^2 \right]^{-1}, \qquad r = 1, 2, \ldots, M, \tag{37}$$

$$\alpha_r = \left[\sum_{i=1,j=1}^{N} |\vec{E}_i(\vec{x}_r) \cdot \vec{E}_j^*(\vec{x}_r)|^2 \right]^{-1}, \qquad r = 1, 2, \ldots, M, \tag{38}$$

(to be used in (1) and (2) respectively). Numerical experiments show that measurement normalization performs rather poorly. All results presented in the next section refer accordingly to operator normalization.

6. NUMERICAL RESULTS

In this section we shall focus on the special (numerically harder) case of intensity measurements. Results pertinent to the field-measurements case have been published elsewhere [16]. The proposed algorithm has been probed under several operating conditions, and compared to a benchmark smart (trap-avoiding) CG code [11]. The performance of the algorithm can be gauged from its success rate (fraction of correct faulty-element(s) identifications, over 10^4 trial array states randomly generated from a prescribed value of the single-radiator fault probability P_F). Our (and any) diagnostic algorithm will fail if the number of radiators N exceeds the number N_c of degrees of freedom of the measured field, viz. [17]:

$$N \geq N_c = \frac{2}{\pi}(k_0 R_a)\frac{R_m}{d}, \tag{39}$$

where R_a and R_m are the (minimum) radii of the antenna and measurement domains, respectively, d is the (minimum) distance between the two, and $k_0 = 2\pi/\lambda_0$ the free-space wavenumber. In all referred numerical experiments we took $R_a = R_m = D/2$, and $\lambda_0/2$-equispaced radiators, so that $D = N\lambda_0/2$. Hence:

$$\frac{N}{N_c} = \frac{2d}{D} \leq 1. \tag{40}$$

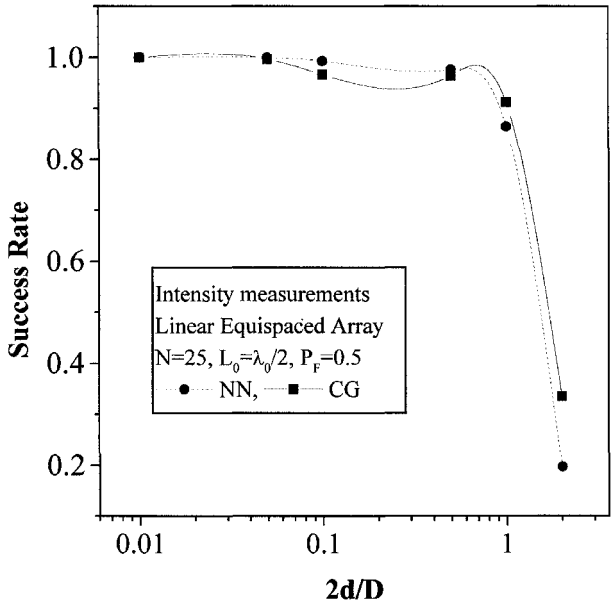

Figure 1. Success rate vs. $2d/D$ (ratio between the number of unknowns and the number of field degrees of freedom).

The typical dependence of the success rate from $N/N_c = 2d/D$ is displayed in Fig. 1. As expected, the success-rate drops down steeply to zero as soon as $2d/D \approx 1$, i.e., as $N \approx N_c$, while keeping essentially equal to one for $2d/D < 1$, i.e., $N < N_c$. The success rate as a function of the radiators fault probability P_F at different values of $2d/D$ is shown in Fig. 2. It is deduced that the proposed algorithm is more reliable than CG. The robustness of the proposed algorithm is illustrated in Figs. 3 and 4. In Fig. 3 the measured data are corrupted by additive (uncorrelated) gaussian noise and the success rate is displayed vs. the noise r.m.s value. In Fig. 4, radiators' positions are affected by random errors uniformly distributed in $(-\Delta, \Delta)$, and the success rate is displayed vs. Δ/λ_0. It is seen that the proposed algorithm is at least as robust as CG.

We shall now discuss the memory and CPU budgets of the proposed method.

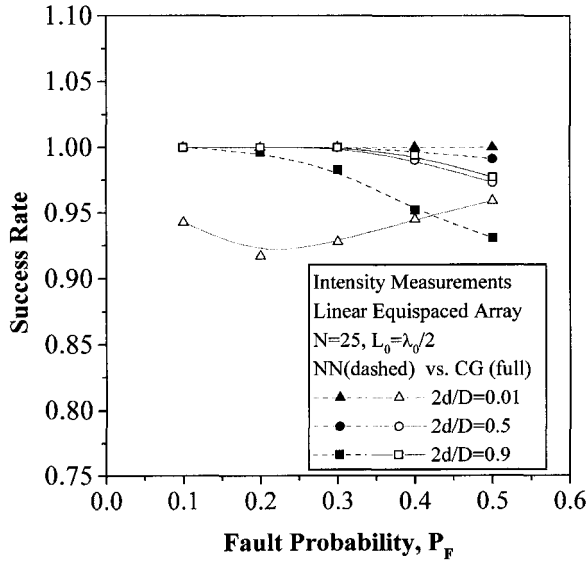

Figure 2. Success rate vs. $2d/D$ (ratio between the number of unknowns and the number of field degrees of freedom).

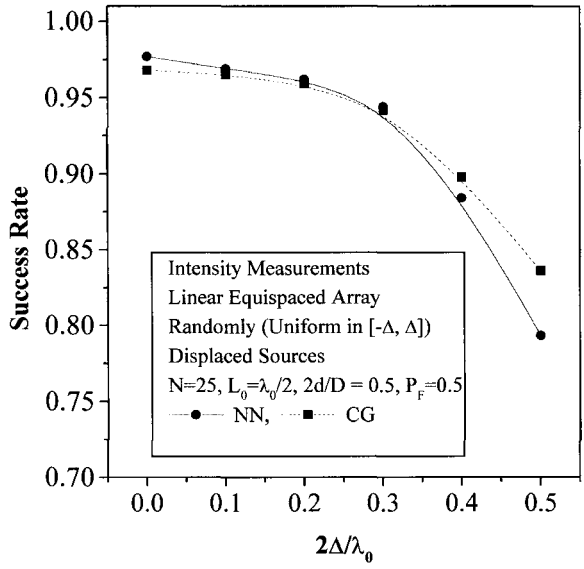

Figure 3. Noisy measurements. Success rate vs. (scaled) r.m.s. noise.

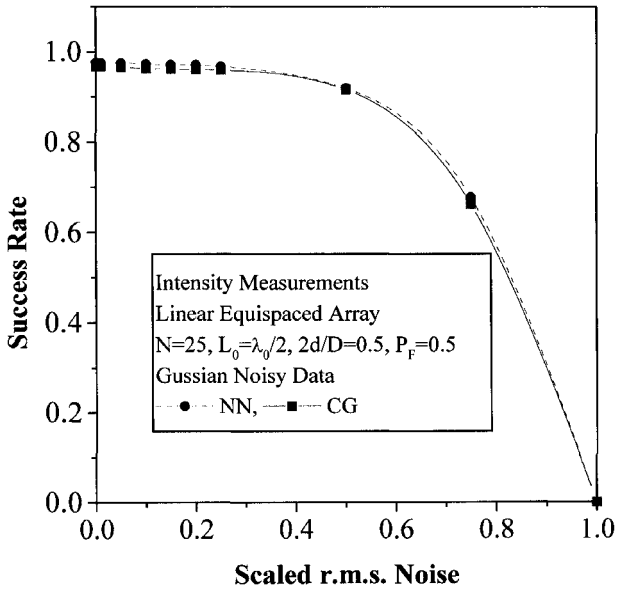

Figure 4. Random (uniform) displaced sources. Success rate vs. scaled maximum displacement.

Given the array geometry and the (nominal) feeding currents, the measurement-independent multilinear polynomial $H_0(\vec{s})$ i.e., the completely symmetric matrices $W_0^{(n)}$, $n = 1, 2, 3, 4$ in (5) can be computed and stored once and for all (step-0). The related CPU and memory budgets scale-up as $M \cdot N^4$ and N^4, respectively. Whenever a set of measured intensities is available for diagnostics, the measurement-dependent polynomial $H_m(\cdot)$, i.e., the completely symmetric matrices $W_m^{(n)}$, $n = 1, 2$ can be computed first (step-1). The related CPU and memory budgets scale-up, respectively, as $M \cdot N^2$ and N^2. Then, the neural net can be evolved until a fixed point is reached, using a suitable threshold parameter updating schedule (step-2). Each iteration of (18) requires $\mathcal{O}(N^3)$ floating-point operations. The typical number of iterations required does not exceed 10^2 for $N \leq 10^3$.

It is understood that the maximum tractable array size turns out to be essentially limited by the amount of available memory in step-0. For a top-end desktop workstation $N \sim 10^3$. Note that this limitation applies, e.g., to CG techniques as well. On the other hand, the overall

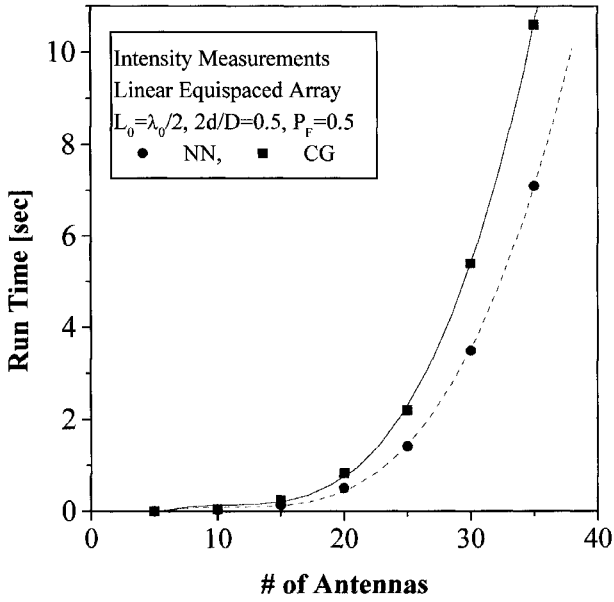

Figure 5. Neural network run-time vs. number of antennas on a P200pro desktop workstation.

diagnostic algorithm burden can be identified with that required to evolve the discrete neural-net (21).

Typical running times are exemplified in Fig. 5, which refers to a P200Pro desktop-WS under WIN95. Relative computing times (scaled to their values at $P_F = 0.50$ as functions of the radiators fault probability P_F at different values of $2d/D$ are shown in Fig. 6. It is seen that the proposed algorithm compares favorably with std (CG) techniques not only in terms of robustness and reliability, but also in terms of speed.

7. CONCLUSIONS - HINTS FOR FUTURE RESEARCH

We presented an efficient framework for faulty elements identification in large antenna arrays, based on discrete asynchronous mean field neural network, and compared its performance to std. (CG) techniques. We are presently investigating the issue of parallelization of (21) through a study of the properties of the corresponding pipelined

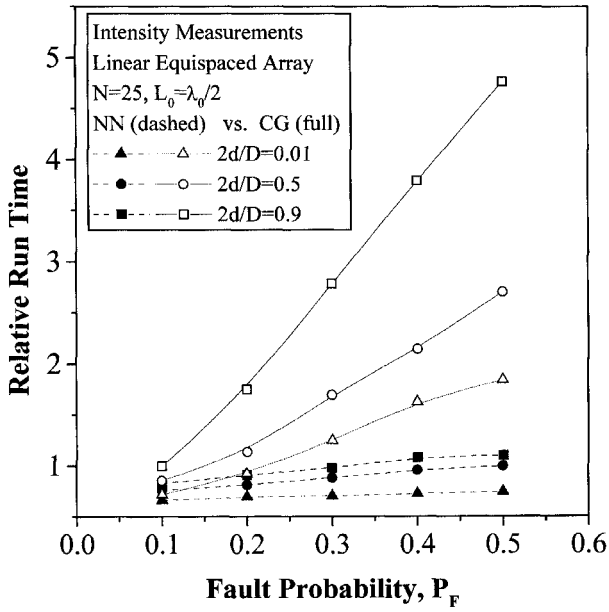

Figure 6. Relative run-time vs. radiator fault probability, at various values of $2d/D$.

and fully-synchronous implementations. A less trivial extension is required to cover the case where each radiator can be in *several* different states (e.g., steerable arrays using stepped phase shifters or attenuators).

APPENDIX A - RELEVANT TO EQS. (25), (27)

In this Appendix we derive eqs. (27)–(29), for the special case of field measurements.

For linear measurements, one can capitalize on Laplace transform. This can be done in two paradigm cases. The first case is: *i)* when all states are unsaturated. In this case, by Gerschgorin theorem [18], the poles of the associated Laplace transform lie in circles with centers $(-1, 0)$ and radii $r_i = .5\beta \sum_j |W_{ij}|$. Hence, if

$$\begin{array}{c} .5\beta \sum_j |W_{ij}| < 1, \\ i = 1, 2, \cdots, N \end{array} \quad \Longrightarrow \quad \begin{array}{c} \beta < 2 \frac{1}{\sum_j |W_{ij}|}, \\ i = 1, 2, \cdots, N, \end{array}$$

no pole belongs to the right complex half-plane. This condition is certainly verified if:

$$\beta < 2 \ \min_i \left[\frac{1}{\sum_j |W_{ij}|} \right] = \frac{2}{\max_i \sum_j |W_{ij}|}; \tag{A1}$$

We denote the last value as β_{min}, viz.:

$$\beta_{min} = \frac{2}{\max_i \sum_j |W_{ij}|}, \tag{A2}$$

such that for $\beta < \beta_{min}$ all poles lie in the left half-plane.

On the other hand, consider case *ii)* where *all* states but two (p and q) are already saturated[9]. In this case, the poles of the relevant Laplace transform are:

$$\lambda_{p,q} = -1 \pm .5\beta |W_{pq}|. \tag{A3}$$

One of the above poles is always in the left half-plane. THe other one belongs to the right half-plane if:

$$.5\beta |W_{pq}| > 1, \quad \Longrightarrow \quad \beta > \frac{2}{|W_{pq}|}. \tag{A4}$$

This condition is certainly verified if:

$$\beta > \max_{p,q} \frac{2}{|W_{pq}|} = \frac{2}{\min_{p,q} |W_{pq}|}, \quad p \neq q. \tag{A5}$$

We denote the last value as β_{max}, viz.:

$$\beta_{max} = \frac{2}{\min_{p,q} |W_{pq}|}, \tag{A6}$$

such that for $\beta > \beta_{max}$ all poles but one lie in the right half-plane.

---

9 Note that if only one state is unsaturated, the corresponding dynamical evolution is trivially $s = s_0 e^{-t}$.

In practice, it is more convenient to introduce an intermediate value β_{med},

$$\beta_{med} = \frac{2}{\max_{p,q}|W_{pq}|}, \tag{A7}$$

such that, for $\beta < \beta_{med}$ it is *impossible* that given $n-2$ saturated states, one of the unsaturated two might saturate.

The following simple strategy can be thus envisaged: evolving the system according to the rule:

$$\beta_i = \beta_{min} + (i-1)(\beta_{med} - \beta_{min}), \tag{A8}$$

starting from $s_k = .5$ at $i = 1$, and iterating the procedure until all states are saturated. This usually occurs far before β_{max} is reached.

APPENDIX B - RELEVANT TO EQS. (29), (31)

In this Appendix we deduce eqs. (31)–(33), for the special case of intensity measurements, where $H(\vec{s})$ is a multinomial operator of fourth degree, and eq. (6) reads explicitly[10]:

$$\begin{cases} \dfrac{du_q}{dt} + u_q = -4\displaystyle\sum_{i,j,k}^{1\ldots N} W^{(4)}_{ijkq} s_i s_j s_k - 3\sum_{i,j}^{1\ldots N} W^{(3)}_{ijq} s_i s_j \\ \qquad\qquad - 2\displaystyle\sum_{i}^{1\ldots N} W^{(2)}_{iq} s_i - W^{(1)}_q, \\ u_q = \beta^{-1} g^{-1}(s_q), \quad q = 1, 2, \ldots, N. \end{cases} \tag{B1}$$

Assuming all states to be saturated, with the exception of states p, q, the r.h.s. of (B1) can be expanded to first order[11] in a neighborhood of $s_r = .5$. In doing so for the unsaturated states, we use the first order binomial expansion:

$$s_r^n = 2^{-n} + 2^{-n+1} n (s_r - 2^{-1}), \tag{B2}$$

while for the saturated states, we use the obvious identities

$$s_k^p = s_k, \qquad \forall p \in \aleph. \tag{B3}$$

[10] It is always possible to assume the $W^{(n)}$ matrices, $n = 1, 2, 3, 4$, to be completely symmetrical w.r.t permutations among the indexes.

[11] Note that, $H(\vec{s})$ being multilinear, $\partial H / \partial s_q$ does not depend on s_q at all.

Hence:

$$\begin{cases} \dfrac{du_q}{dt} + u_q = -4 \displaystyle\sum_{i,j,k}^{1\ldots N} (W^{(4)}_{ijkq}\delta_{ir} + W^{(4)}_{ijkq}\delta_{jr} + W^{(4)}_{ijkq}\delta_{kr}) s_i s_j s_k \\ \qquad\qquad - 3 \displaystyle\sum_{i,j,k}^{1\ldots N} (W^{(3)}_{ikq}\delta_{ir}\delta_{kj} + W^{(3)}_{ikq}\delta_{jr}\delta_{ik}) s_k s_i s_j \\ \qquad\qquad - \dfrac{8}{3} \displaystyle\sum_{i,j,k}^{1\ldots N} W^{(2)}_{rq}\delta_{ir}\delta_{jr}\delta_{kr} s_i s_j s_k - \dfrac{1}{3} W^{(1)}_q, \\ u_q = \beta^{-1} g^{-1}(s_q), \quad q = 1, 2, \ldots, N, \end{cases} \tag{B4}$$

where δ_{rs} is the Kronecker symbol. Next we define:

$$U_q = \beta^{-1} g^{-1}(s_q^3), \tag{B5}$$

where s_q is an unsaturated state. Expansion of the second eq. in (B4) and of eq. (B5) in a neighborhood of $s_q = 0.5$ gives:

$$u_q - \beta^{-1} g^{-1}(.5) = \beta^{-1} D\left[g^{-1}(s_q)\right]_{s_q=0.5} (s_q - .5), \tag{B6}$$

and

$$U_q - \beta^{-1} g^{-1}(.125) = \frac{3}{4} \beta^{-1} D\left[g^{-1}(s_q^3)\right]_{s_q=0.5} (s_q - .5), \tag{B7}$$

whence:

$$\frac{u_q - \beta^{-1} g^{-1}(1/2)}{U_q - \beta^{-1} g^{-1}(1/8)} = \frac{4}{3} \left\{ \frac{D[g^{-1}(s_q)]}{D[g^{-1}(s_q^3)]} \right\}_{s_q=1/2}, \tag{B8}$$

where $D[\cdot]$ is the derivative operator. Differentiating in time eqs. (B6) and (B7), and comparing the resulting identities, we further get:

$$\frac{\dfrac{du_q}{dt}}{\dfrac{dU_q}{dt}} = \frac{4}{3} \left\{ \frac{D[g^{-1}(s_q)]}{D[g^{-1}(s_q^3)]} \right\}_{s_q=1/2}. \tag{B9}$$

Using eqs. (B9) and (B8) in (B4) gives:

$$\begin{cases} \dfrac{dU_q}{dt} + U_q = T_q^{ijk} s_i s_j s_k + h_q, \\ U_q = \beta^{-1} g^{-1}(s_q^3), \end{cases} \tag{B10}$$

where:

$$T_q^{ijk} = -3c(W_{i,jkq}^{(4)}\delta_{ir} + W_{ijkq}^{(4)}\delta_{jr} + W_{ijkq}^{(4)}\delta_{kr}) - (9c/4)(W_{ikq}^{(3)}\delta_{ir}\delta_{kj} + W_{ikq}^{(3)}\delta_{jr}\delta_{ik}) - 2cW_{rq}^{(2)}\delta_{ir}\delta_{jr}\delta_{kr}, \quad \text{(B11)}$$

and:

$$c = \left\{ \frac{D[g^{-1}(s_q^3)]}{D[g^{-1}(s_q)]} \right\}_{s_q=1/2}, \quad \text{(B12)}$$

h_q being an irrelevant constant. Equation (B10) can be regarded as a *linear* neural net in the *compound state* $s_{ijk} = s_i s_j s_k$. Little thought reveals that the net (B10) and (6) converge (or not) together. Thus, we can compute β_{med} and β_{max} for the linear network (B10), using eqs. (28) and (29). Hence:

$$\beta_{max} = \frac{2}{\min_{i,j,k\neq q}|T_q^{ijk}|}, \quad \text{(B13)}$$

and:

$$\beta_{med} = \frac{2}{\max_{i,j,k\neq q}|T_q^{ijk}|}. \quad \text{(B14)}$$

Restricting to the *diagonal* states $i = j = k = h$, one finally gets, in view of (B11),

$$\beta_{max} = \frac{1}{c\ \min_{h\neq q}|W_{h,q}^{(2)}|}, \quad \text{(B15)}$$

and:

$$\beta_{med} = \frac{1}{c\ \max_{h\neq q}|W_{h,q}^{(2)}|}. \quad \text{(B16)}$$

ACKNOWLEDGMENT

The authors thank V. Galdi for a number of constructive discussions. This work has been done in partial fulfillment of G. Castaldi's Ph.D. programme, and has been sponsored in part by MURST.

REFERENCES

1. Gattoufi, L. et al., "Matrix method for near-field diagnostic technique of phased array," *Proc. IEEE Intl. Symp. on Phased Array Systems and Technology,* 52–57, Boston, 1996.
2. Yaccarino, R. G., and Y. Rahmat-Samii, "Microwave antenna imaging and phaseless reconstruction," *Int. Journal of Imaging Systems and Technol.,* Vol. 199, 396–406, 1997.
3. Burlisch, A., and B. Stohr, *Numerical Analysis,* Springer Verlag, Berlin, 1979.
4. Beveridge, G. S., and R. S. Schechter, *Optimization Theory and Practice,* McGraw-Hill, New York, 1970.
5. Fogel, D. B., "An introduction to simulated evolutionary optimization," *IEEE Trans. Neural Networks,* Vol. 5, 3–14, 1994.
6. Aarts, E. H. L., and J. Korst, *Simulated Annealing and Boltzmann Machines,* Wiley, New York, 1990.
7. Haykin, S., *Neural Networks,* MacMillan, New York, 1994.
8. Vidyasagar, M., "Minimum-seeking property of analog neural network with multilinear objective functions," *IEEE Trans. Automatic Control,* Vol. 40, 1359–1375, 1995.
9. Vidyasagar, M., "Convergence of higher-order neural networks with modified updating," *Proc. IEEE Int. Conf. on Neural Networks,* 1379–1384, 1993.
10. Lee, B. W., and B. J. Sheu, "Paralleled hardware annealing for optimal solutions on electronic neural network," *IEEE Trans. Neural Networks,* Vol. 4, 588–598, 1993.
11. Press, W. H. et al., *Numerical Recipes in C,* Cambridge Univ. Press, Cambridge, 1995.
12. Cohen, M. A., and S. Grossberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks," *IEEE Trans. Systems Man Cybern.,* Vol. 13, 815–825, 1983.
13. La Salle, J. P., and S. Lefschetz, *Stability by Lyapounov's Direct Method,* Academic Press, New York, 1961.
14. Hirsch, M. W., and S. Smale, *Differential Equations, Dynamical Systems and Linear Algebra,* Academic Press, New York, 1974.
15. Zeidler, E., *Nonlinear Functional Analysis and its Applications,* Vol. I, Springer Verlag, Berlin, 1986.
16. Castaldi, G. et al., "Mean-field neural networks for antenna array diagnostics," in *Neural Nets,* M. Marinaro and R. Tagliaferri Eds., 185–193, Springer Verlag, Berlin, 1998.
17. Bucci, O. M., and G. Franceschetti, "On the degrees of freedom of scattered fields," *IEEE Trans. Antennas Propagat.,* Vol. 37, 918–926, 1989.

18. Smith, G. D., *Numerical Solutions of PDE by Finite Difference Methods,* Oxford Univ. Press, Oxford, 1985.