An Analog Scheme for Fixed-Point Computation–Part II: Applications

K. Soumyanath, *Member, IEEE,* and Vivek S. Borkar

***Abstract*—In a companion paper [6] we presented theoretical analysis of an analog network for fixed-point computation. This paper applies these results to several applications from numerical analysis and combinatorial optimization, in particular: 1) solving systems of linear equations; 2) nonlinear programming; 3) dynamic programing; and 4) network flow computations. Schematic circuits are proposed for representative cases and implementation issues are discussed. Exponential convergence is established for a fixed-point computation that determines the stationary probability vector for a Markov chain. A fixed-point formulation of the single source shortest path problem (SPP) that will always converge to the exact shortest path is described. A proposed implementation, on a 2-μ complementary metal–oxide–semiconductor (CMOS) process, for a fully connected eight-node network is described in detail. The accuracy and settling time issues associated with the proposed design approach are presented.**

I. Introduction

TWO important problems in applied mathematics are computations of maxima/minima and fixed points. Recently, there has been tremendous activity in applying analog computational methods for the former problem. These methods are based primarily on energy minimization techniques and have been extensively studied [17]. The object of this paper is to provide a complementary paradigm for fixed-point computations.

Fixed-point computations occur in a wide variety of applications, most notably in nonlinear and dynamic programming. Classical approaches to fixed-point computations are based on the following discrete time iteration:

$$x_{n+1} = F(x_n), \qquad n \geq 0. \tag{1}$$

Parallel implementations of these iterations have been extensively studied for contractions and a limited class of nonexpansive mappings [3]. The above reference also discusses synchronous and, under certain conditions, asynchronous parallel implementations of (1) for several nonexpansive mappings that satisfy the assumptions in [3].

Manuscript received November 19, 1997; revised March 31, 1998. This work was supported in part by the U.S. Army Research Office under Grant ARO DAAL 03-92-G-0115. Preliminary versions of this paper were presented at the ICCD 1990 and the IEEE ASIC Conference 1992. This paper was recommended by Associate Editor A. Csurgay.

K. Soumyanath is with the Circuits Research Laboratory, Intel Corporation, EY2-07, Hillsboro, OR 97124-6497 USA.

V. S. Borkar is with the Department of Computer Science and Automation, Indian Institute Of Science, Bangalore 560012, India.

Publisher Item Identifier S 1057-7122(99)00844-2.

In a companion paper [6] we introduced a continuous time version of the above iteration given as

$$\dot{x}(t) = F(x(t)) - x(t), \qquad t \geq 0. \tag{2}$$

The mapping F associated with (2) requires no assumptions other than nonexpansiveness as a condition for its convergence and is therefore applicable to a wide variety of useful mappings. The system can be viewed as the limiting case of the classical overrelaxed version of (1) as the relaxation parameter goes to zero [6]. The notion of synchronism does not feature in the continuous-time version unless we reformulate the above differential equation to include delays. We do not consider delays in the present paper.

Analog parallel computation schemes have been proposed for several problems in early vision, image processing, and optimization. [33]. The work by Sudarshanan and Sunderasan [30] and its improvements [7] describe an analog parallel technique for quadratic optimization. An analog network for solving systems of linear equations is also discussed in [17]. Analog techniques for nonlinear programming are presented in [22] and their realizations, using conventional as well as switched capacitor integrators, are described in [2] and [26]. A dynamic programming network operating in the analog domain is also reported in [15]. Analog parallel networks for the shortest path problem (SPP), in particular, based on neural networks have also been reported in [1] and [34]. Our system is in the spirit of the former, but markedly is different in one important aspect; it is not based exclusively on the energy-minimization paradigm (although there is an area of overlap in gradient-descent techniques as described in Section III). Instead, the system studied in this paper executes what may be considered a continuous-time version of the over relaxed fixed-point iteration. Two recent monographs [24], [21] on this theme give a comprehensve survey and bibliography. It should be emphasized that by energy minimization methods we mean gradient flows driven by the gradient of an appropriate energy function. Any ordinary differential equation (o.d.e.) with a globally asymptotically stable attractor will have a Liapunov function which decreases along the nonconstant trajectories thereof by virtue of the converse Liapunov theorems [32]. This does not imply, however, that the driving vector field is a gradient of the same.

This approach provides a simple and natural analog parallel approach to several fixed-point computations, most notably, the dynamic programming formulations of the SPP, which is fundamental to network computations. The SPP has a wide variety of applications and analog approaches are potentially

useful when the edge costs are dynamic, e.g., in data networks [3], [4]. If the the edge costs vary at a rate that is slow compared to the settling time of the processor, the proposed approach enables the real-time computation of shortest paths without recourse to high-speed area-intensive A/D conversions. Furthermore, the approach to dynamic programming based on our system has the desirable attribute of *always* leading to the global optimum. This is in contrast to energy minimization formulations of SPP's, [1], [15], [34], which, in addition to being complex, suffer from local minima traps (see the concluding paragraph of Section III-B for an elaboration on this point). In addition, the fixed-point computational view of nonlinear optimization permits us to make interesting observations about the behavior of gradient-descent schemes near a local minimum.

The rest of the paper consists of four sections. In Section II we introduce the notation, describe some preliminaries and restate the main convergence results of [6] without proof. Section III describes several applications of fixed-point computations under nonexpansive mappings and includes the SPP which requires convergence under the ∞ norm. In Section IV we describe a design study for a possible 2-μ complementary metal–oxide–semiconductor (CMOS) very large scale integration (VLSI) implementation of the paradigm for solving the single source SPP on an eight-node fully connected network. Section V concludes the paper with some relevant remarks.

II. Notation and Preliminaries

Let $\mathcal{R}^d$ denote the d-dimensional Euclidean space and $x = [x_1 \cdots x_d]^T$ its typical element. We shall consider the following family of norms on $\mathcal{R}^d$:

$$\|x\|_p = \left| \frac{1}{d} \sum_{i=1}^{d} |x_i|^p \right|^{1/p}, \qquad 1 \le p < \infty$$
$$\|x\|_\infty = \max_i |x_i|. \tag{3}$$

The following lemma collects some well-known and easily established properties of these norms.

Theorem 2.1:

1) For $q > p$ there exists a constant $c = c(p,q) > 0$ such that
$$c(p,q)\|x\|_q \le \|x\|_p \le \|x\|_q.$$
2) The map $p \in [1,\infty] \to \|x\|_p$, $x \in \mathcal{R}^d$ is continuous.

In addition, the following weighted maximum norms can be defined on $\mathcal{R}^d$:

$$\|x\|_{p,w} = \left(\frac{1}{d} \sum_{i=1}^{d} w_i^p |x_i|^p \right)^{1/p}, \qquad p \in (1,\infty)$$
$$\|x\|_{\infty,w} = \max_i w_i |x_i| \tag{4}$$

where $w = [w_1, \cdots, w_d]$, $w_i > 0\ \forall i$ is a vector of weights.

Definition 2.1: For $p \in [1,\infty]$ a map F: $\mathcal{R}^d \to \mathcal{R}^d$ is said to be p nonexpansive if $\|F(x) - F(y)\|_p \le \|x - y\|_p$ $\forall x, y \in \mathcal{R}^d$. It is said to be a p contraction if, in addition, there exists an $\alpha \in (0,1)$ such that $\|F(x) - F(y)\|_p \le \alpha \|x - y\|_p$ $\forall x, y \in \mathcal{R}^d$.

Definition 2.2: $x^* \in \mathcal{R}^d$ is said to be a fixed point of F $\mathcal{R}^d \to \mathcal{R}^d$ if $F(x^*) = x^*$.

If F is a p contraction, the contraction mapping theorem [27] ensures that it has a unique fixed point x^* and the iteration $x_{n+1} = F(x_n)$, $n \ge 0$ with arbitrary initial condition $x_0 \in \mathcal{R}^d$ satisfies

$$\|x_{n+1} - x^*\|_p \le \alpha^n \|x_1 - x^*\|_p \overset{n\uparrow\infty}{\longrightarrow} 0.$$

If F is only nonexpansive, however, it may have no fixed point (for example, F: $\mathcal{R} \to \mathcal{R}$ given by $F(x) = x + 1$) or, uncountably, many fixed points (e.g, $F(x) = x$). The set of fixed points, denoted by S henceforth, is, however, always closed.

Let F: $\mathcal{R}^d \to \mathcal{R}^d$ be a p nonexpansive map for $p \in (1,\infty]$ with a nonempty set of fixed points S.

Theorem 2.2: [6] The set S is connected and the coupled dynamical system of (2) converges to a single point in S which may depend on the initial condition. Furthermore, the convergence is at an exponential rate if F is a p-contraction.

Remark: The result and the proof detailed in [6] extend in a straightforward manner to the family of weighted norms defined in (4) above.

III. Analog Fixed-Point Computations for Nonexpansive Mappings

The dynamical system of (2) can be applied to several problems that admit nonexpansive fixed-point formulations.

A. Solution of Equations

If f: $\mathcal{R}^d \to \mathcal{R}^d$ is such that the map F: $x \to x + f(x)$ is p nonexpansive for some $p \in (1,\infty]$ then a solution of $f(x) = 0$, which is a fixed point of f, can be found by using the system in (2). In particular, the linear system $\boldsymbol{A}x = b$ where $\boldsymbol{A} = [[a_{ij}]]$ is a $d \times d$ matrix can be solved using (2) if $F(x)$ is defined as follows [3]:

$$F(x) = -D^{-1}Bx + D^{-1}b \tag{5}$$

where D is a diagonal matrix whose diagonal entries are equal to the corresponding entries in A and $B = \boldsymbol{A} - D$. The condition for nonexpansiveness of $F(x)$, as defined in (5), is that all the eigenvalues of $M = D^{-1}B$ be less than or equal to unity or equivalently, the spectral radius of $M, \rho(M) \le 1$. Since M, as defined above, is such that $|m_{ij}| = |a_{ij}|/|a_{ii}|$ for $i \ne j$ and $|m_{ii}| = 0$, then $\rho(M) \le 1$ if and only if $\Sigma_{i \ne j} |a_{ij}| \le |a_{ii}|\ \forall i$. This is the weak diagonal-dominance condition on $\boldsymbol{A}$. If the above inequality is strict, then A is diagonally dominant and the solution to (3) converges to its unique fixed point at an exponential rate as the associated F becomes a contraction [3].

Fig. 1 shows the schematic for a computational module evaluating a component x_i, using the above realization for F_i. The summers, multiplier,s and integrators can be implemented using conventional analog circuitry. The settling time of the circuit will be dominated by the time constant of the integrator which, when implemented in CMOS with an operational transconductance $(= g_m)$ amplifier (OTA) and an integrating capacitor C, is given by g_m/C. The accuracy of the computations is limited by the matching accuracy of the current mirrors

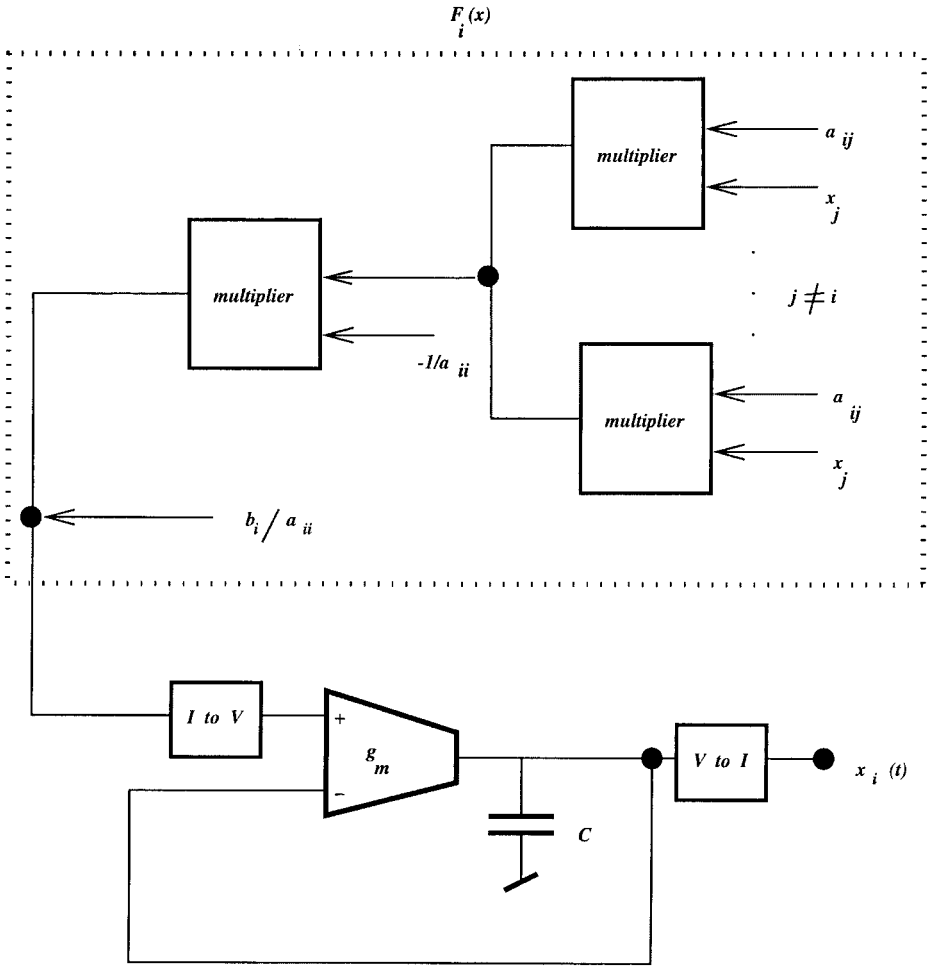


Fig. 1. Computational element x_i for linear system.

and the fidelity of the integrator and multiplier components. The current mirror accuracy can be improved using cascoded current mirrors (for matching across a wide dynamic range) and Gilbert multipliers with bipolar components whereby an accuracy of 0.01% over three orders of magnitude can be achieved [33]. The use of current-mode computations is clearly desirable given the simplicity of current-mode summing and copying operations. The above implementation uses an operational transconductance amplifier (OTA)-C integrator; thus the dynamic range of $F(x) - x$ is limited to the linear range of the OTA, which is typically a few tens of millivolts in CMOS implementations. The dynamic range (equivalently the word length) of the computational element can be improved by using current-mode integrators, which remove the need for wide-range linear I/V convertors [25].

Another related problem is to solve the system of equations $P^T x = x$ where $P \in \mathcal{R}^{d\times d}$ is a stochastic matrix. Then $F(x) = P^T x$ is 1-nonexpansive. Though we have not considered the case $p = 1$ in our analysis, this special instance of it can be handled quite easily. Assuming P is irreducible, the fixed points of F are scalar multiples of the unique stationary probability vector x^* for the Markov chain with transition matrix P. Equation (2) then becomes

$$\dot{x}(t) = (P^T - I)x(t) \tag{6}$$

where I is the identity matrix. Since $x \to P^T x$ is as shown in Appendix A, 1-nonexpansive $x(t) \to \{\alpha x^* | \alpha \in \mathcal{R}\}$. Left multiplying (6) by the row vector $e = [1, 1 \cdots, 1]$, we get

$$\frac{d}{dt}\left(\sum_{i=1}^{d}(x_i(t))\right) = 0.$$

Thus, $\Sigma_{i=1}^{d}(x_i(t))$ remains constant with t, implying that

$$x(t) \to \sum_{i=1}^{d}(x_i(0))x^* \triangleq \overline{x}.$$

The vector x^* can be retrieved from this as long as $\Sigma_{i=1}^{d}(x_i(0)) \neq 0$; a restriction that needs to be imposed on the initial condition. Though the exponential convergence condition of Theorem 2.1 is not directly applicable, we can prove exponential convergence by using linear systems theory.

Lemma 3.1: $x(t) \to \overline{x}$ exponentially

Proof: See Appendix A.

B. Nonlinear Programming

Many recursive schemes for solving nonlinear programming problems involve iterations of the type $x_{n+1} = F(x_n), n \geq 0$ where F is either an ∞ contraction or a contraction with respect to the weighted norm defined in (4), [3]. Our analysis

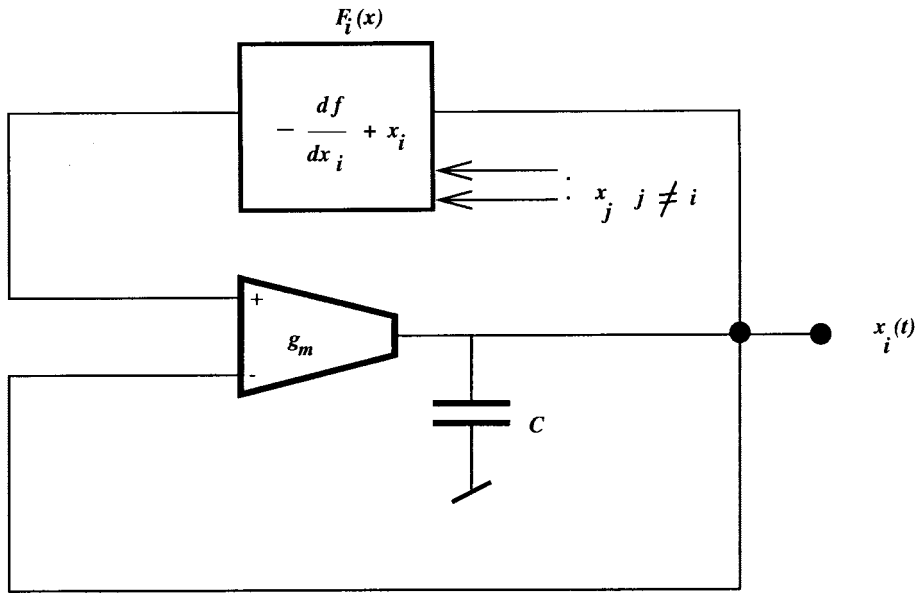


Fig. 2. Computational element x_i for quadratic programming.

extends easily to this case. An important instance here is the gradient-descent algorithm

$$\dot{x}(t) = -\gamma \bigtriangledown f(x(t)) \tag{7}$$

which reduces to (3) for

$$F(x) = x - \gamma \bigtriangledown f(x(t)). \tag{8}$$

For quadratic f, say, $f(x) = x^T Q x + Cx$ with $Q \geq 0$, F is a two-contraction if $\gamma < 2/\lambda_{\max}(Q)$. Note that we do not change the gradient-descent algorithm; only the view point. Hence, the system is equivalent to those reported in [2], [7], [14], and [30] for quadratic programming. Fig. 2 is a schematic of a computational element for performing general nonlinear optimization. This network is distinguished from previous work in that no nonlinear resistors or saturating elements are required for unconstrained optimization. In the case of quadratic f computation of (8) reduces to a linear combination of summation and multiplication. It can therefore be accomplished using current summers and wide-range Gilbert multiplier components in a manner similar to the linear systems case. The use of high-precision circuit techniques, such as switched capacitor filters, can offset the problems associated with noise and parameter drift. The fixed-point view of gradient-descent schemes represented by (8), however, enables us to analyze the behavior of gradient-descent schemes in the vicinity of local minima for a general class of nonlinear programming problems. Rewrite (7) as

$$\dot{x}(t) = -a \bigtriangledown f(x(t)) \tag{9}$$

where a is a scaling parameter and f: $\mathcal{R}^d \rightarrow \mathcal{R}^d$ is twice continuously differentiable with

$$\lim_{\|x\| \rightarrow \infty} |f(x)| = \infty.$$

(For example, f could be the energy function in Hopfield-type neural networks.) It is easy to see that $t \rightarrow f(x(t))$ is strictly decreasing away from the set $\{x | \bigtriangledown f(x) = 0\}$ and is thus converging to a connected set of local minima, say M. Suppose M has an open neighborhood on which the Hessian matrix

$$H(x) = \left[\left[\frac{\partial^2 f}{\partial_i \partial_j}(x)\right]\right]$$

is nonnegative definite. Again viewing (9) as a fixed-point computation with $F(x)$ as in (8), the Jacobian of $F(x)$ is given by $I - aH(x)$, I being the identity matrix with eigenvalues $1 - a\lambda(x), \lambda(x)$ being an eigenvalue of $H(x)$. It follows that for sufficiently small $a > 0$ ($a \leq 2/\lambda_{\max}(x)$, to be precise) F is two-nonexpansive and thus the arguments leading to Theorem 3.1 can be employed to show that $x(\cdot)$ converges to a single point of M. This point, however, can vary from trajectory to trajectory of the o.d.e. In fact, the above can be strengthened for general $a > 0$ by the following time-scaling argument. If $x(\cdot)$ is as above and $y(t) = x(ct)$ for some $c > 0$, then $dy(t)/dt = -ac \bigtriangledown f(y(t))$ and ac can be made small by making c small. Then $y(\cdot)$ and therefore $x(\cdot)$ must converge to a single local minimum of $f(\cdot)$.

The gain here is nontrivial on at least three counts. Convergence of $x(\cdot)$ to a single point as opposed to its wandering around a connected set of local minima could be a desirable property in itself, e.g., when the $x(\cdot)$ triggers a control device and its stability (i.e., convergence) is preferred to jitter that could be caused by its wandering. Also, when one is reading components of $x(\cdot)$ after a sufficiently long time in order to evaluate approximately the local minimum, the following problem can arise.

Suppose in a two-dimensional (2-D) situation, $[x_1, x_2]$, $[y_1, y_2]$ are distinct local minima in a connected set of local minima that has attracted $x(\cdot)$ and due to recording device delays or otherwise one reads $[x_1, y_2]$. Then $[x_1, y_2]$ need not be a local minimum at all. This is just a simple illustration of the complications that can arise if $x(\cdot)$ wanders. More complicated examples can be constructed. Finally, it is easier to evolve good stopping criteria for the convergent case.

So far we have considered converting a gradient algorithm for minimization into a fixed-point computation, wherein we did not change the algorithm, only the viewpoint. The reverse situation is difficult; i.e., given a fixed-point seeking network as in (3) it is not always possible to write $F(x) - x$ as $\nabla g(x)$ for some g. There have been attempts in the literature to convert fixed-point computations into energy minimization computations [1], [15], [34] but, in addition to added complications, they suffer from the possibility of being trapped in local minima that are not fixed points. Consider, for example, $F(x) = x + \sqrt{g(x)}$, where $\sqrt{\cdot}$ denotes the nonnegative root with $g(x) \geq 0$, $g(0) = 0$ and $g(x) > 0$ for $x \neq 0$, with $x \to g(x)$ having several local minima other than at zero. One may consider applying gradient descent to the energy function $(F(x) - x)^2 = g(x)$ which may get stuck in a local minimum that does not correspond to a fixed point of $f(x)$.

C. Discounted Markov Decision Processes

Consider a controlled Markov chain $\{X_n, n = 0, 1 \cdots\}$ taking values in a state space $Y = \{y_1 \cdots y_d\}$ and an associated control space $A = \{a_1, \cdots, a_r\}$, with a set of transition probabilities $p(y_i, y_j; a) =$ the probability of transition from $y_i \in Y$ to $y_j \in Y$ when control $a \in A$ is used. Let c: $Y \times A \to R$ be a cost function and $\beta \in (0, 1)$ a discount factor. One aims to minimize the total discounted cost $E[\Sigma_{n=0}^{\infty} \beta^n c(X_n, Z_n)]$ where $\{Z_n\}$ is the control sequence. Letting

$$V(i) = \inf E\left[\sum_{n=0}^{\infty} c(X_n, Z_n)/X_0 = y_i\right], \quad 1 \leq i \leq d$$

where the infirmum is over all admissible $\{Z_n\}$, $V = [V(1), \cdots, V(d)]^T$ is the unique solution to the dynamic programming equations

$$V(i) = \min_a \left[c(y_i, a) + \beta \sum_{j=1}^{d} p(y_i, y_j; a) V(j)\right], \quad 1 \leq i \leq d.$$

Let F: $\mathcal{R}^d \to \mathcal{R}^d$ denote the map

$$F_i(x) = \min_a \left[c(y_i, a) + \beta \sum_{j=1}^{d} p(y_i, y_j; a) x_j\right], \quad 1 \leq i \leq d.$$

Then it is easily verified that $\|F(x) - F(y)\|_\infty \leq \beta \|x - y\|_\infty$, $x, y \in \mathcal{R}^d$, and thus the solution of (3) with this choice of F converges exponentially (with time constant $(1 - \beta)^{-1}$) to $V =$ the unique fixed point of F. Given V, a standard procedure yields the optimal control policy [5].

It should be noted that extensions to more general Markov decision processes and certain other cost structures are possible. We have considered here only a simple illustrative case. Another is briefly mentioned below.

D. Stochastic SPP's

An extension of the above paradigm is a controlled Markov chain as described, but with the cost

$$E\left[\sum_{n=0}^{\tau} c(X_n, X_{n+1}, Z_n)\right]$$

where $\tau = \min_n \{n \geq 0 | X_n = y_s\}$ is the first hitting time of the state y_s. The cost is then the minimum expected total cost up to the first hitting time of y_s from the initial state, with an additional provision of explicit control dependence of the per stage cost $c(y_i, y_j, u)$. This problem leads to the dynamic programming equations

$$V(i) = \min_u \left[\sum_{j=1}^{d} p(y_i, y_j, u)(c(y_i, y_j, u) + V(j)\right]$$

for $1 \leq i < s$, with $V(s) = 0$. The solution of this set of equations is equivalent to finding the fixed point (unique in this case, see, e.g., [5]) of the ∞-nonexpansive map F: $\mathcal{R}^d \to \mathcal{R}^d$ given by

$$F_i(x) = \min_u \left(\sum_{j=1}^{d} p(y_i, y_j, u)(c(y_i, y_j, u) + x_j)\right), \quad 1 \leq i < s$$

E. Network Flow Computations

Consider a directed graph $G = (V, E), |V| = n$. Each edge (i, j) is associated with an integer a_{ij} referred to as the cost coefficient of (i, j). Let f_{ij} be the flow of edge (i, j), then the problem

$$\text{minimize} \quad \sum_{(i,j) \in E} a_{ij} f_{ij} \tag{10}$$

$$\text{subject to} \quad \sum_{j|(i,j) \in E} f_{ij} - \sum_{j|(j,i) \in E} f_{ji} = s_i, \qquad \forall i \in N \tag{11}$$

$$b_{ij} \leq f_{ij} \leq c_{ij}, \qquad \forall (i, j) \in A \tag{12}$$

is the linear network flow problem [3]. In the above a_{ij}, b_{ij}, c_{ij}, and s_i are given integers. We refer to b_{ij} and c_{ij}, and the interval $[b_{ij}, c_{ij}]$ as the flow bounds and the feasible flow range of edge (i, j), respectively. The value s_i is referred to as the supply of a node i and $-s_i$ as the demand of node i. It can be shown that the assignment, max-flow, and SPP's are special cases of (10) above. The nonlinear version of the above can be formulated by replacing $a_{ij} f_{ij}$ with nonlinear cost functions $a_{ij}(f_{ij})$. If the functions $a_{ij}(\cdot)$ are strictly convex real valued functions, it can be shown that the resulting convex network flow (CNF) problem has a unique optimal solution.

A fixed-point iteration for solving (10) can be set up in the domain of price vectors (which are Lagrange multipliers) using the map $\mathcal{F}$: $\mathcal{R}^n \to \mathcal{R}^n$ detailed in (pp. 457–460, [3]). This map remains nonexpansive for the (CNF) problem and hence (3) is applicable for both the linear and nonlinear network flow computations.

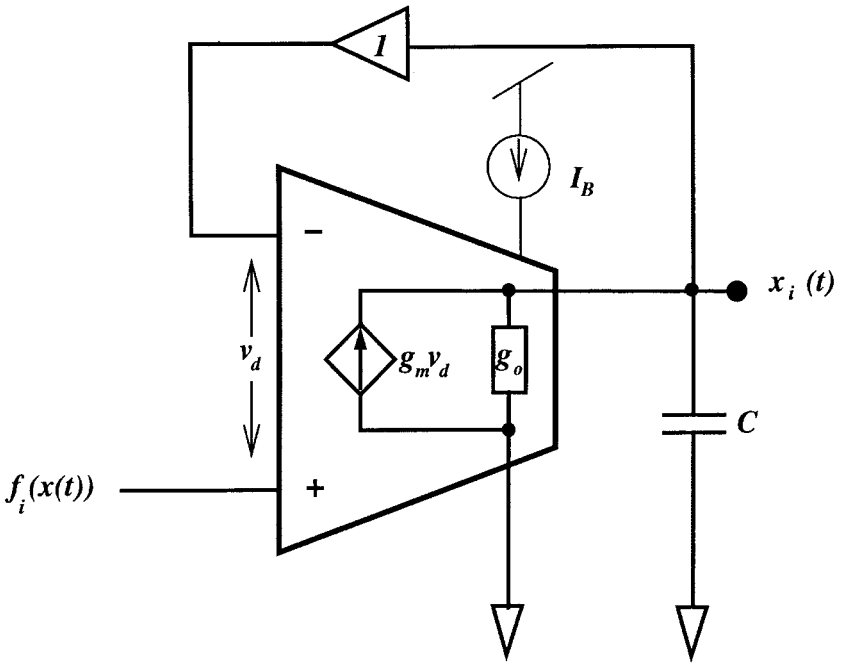


Fig. 3. Voltage mode dynamic operator implementation.

F. Dynamic Programming (SPP's)

Let $G = (V, E), |V| = n$ be a simple weighted graph with a set of edge weights $c_{ij} > 0$ where c_{ij} is the cost of edge (ij) $\forall i, j \in E$, $i \neq j$. The problem is to find the shortest path costs x_i, from nodes $v_i \in V, i = 1, \ldots, n-1$ to a single destination(or source, the problem is often referred to as the single-source SPP $v_0 \in V$. Then the $x = [x_1 \ldots, x_{n-1}]$ can be shown to be the unique fixed points of the ∞-nonexpansive map F_i given by

$$F_i(x) = \min(\min_{1 \leq j \leq n-1}[c_{ij} + x_j], c_{i0}), \quad 1 \leq i \leq n-1. \tag{13}$$

Hence, the solutions $x(\cdot)$ of (3) converge to x^* corresponding to the shortest paths. The above is the classic dynamic programming formulation of SPP, due to Bellman, and is often referred to as Bellman's equation.

G. Voltage Mode Dynamic Operator Implementation

All the applications described in the foregoing sections require a dynamic element to implement the integration implied in (1). A transimpedance integrator provides a convenient method of realizing the term $F_i(x(t))) - x_i(t)$ as the input to the dynamic element that computes the state variable $x_i(t)$. If an ideal transconductor $(= g_m)$ is used in conjunction with a capacitor C a lossy integrator with finite time constant $\tau = g_m/C$ results. The ideal weighted solution for (1), in a small signal sense, is given by

$$\mathcal{X}_i(s) = \frac{-g_m}{Cs}\left[\mathcal{F}_i(s) - \mathcal{X}_i(s)\right] \tag{14}$$

with the proviso that $f_i(\cdot) - x_i(\cdot)$ remains within the linear range of the integrator (which is typically a few tens of millivolts).

Fig. 3 shows a model of a nonideal transimpedance integrator in a typical CMOS implementation built using an OTA with an associated bias current I_B and a finite output conductance g_o. The above model assumes that the input conductance of the OTA (primarily caused by the gate-to-source capacitance and associated parasitics) is negligible compared to g_m and g_0 at the frequencies of interest. The resulting small signal response is

$$\mathcal{X}_i(s) \approx -\frac{A_0\omega_0}{s+\omega_0}[\mathcal{F}_i(s) - \mathcal{X}_i(s)] \tag{15}$$

where $A_0 = g_m/g_0$ is the dc open loop gain, $\omega_0 = g_0/C$ is the dominant pole, and g_0 is the output conductance of the integrator.

The single pole integrator model in Fig. 3 can be used to derive an expression that quantifies the effects of the nonideal behavior of the dynamic element when used in implementations of our fixed-point seeking system. We have

$$\mathcal{X}_i(s) \approx \left[\frac{\omega_m}{s+1} + \frac{\omega_m(1-\omega_m)}{s^2 + (\omega_m+1)s + \omega_m}\right]\mathcal{F}_i(s) \tag{16}$$

where $\omega_m = g_m/C$. Since the ratio of g_m to g_0 is typically several hundred for standard CMOS processes, we have assumed in the foregoing that $\omega_m \gg \omega_0$. The first term in the above is the ideal weighted solution, while the second represents the distortion associated with the nonidealities of the transimpedance integrator. The model described by (16) was validated by comparison with circuit simulations (using level two models) of a standard two-stage OTA design in a 2-μ p-well process. The simulations confirmed that the model was robust over the frequency range where the single pole approximation is valid.

The responses predicted by the above model indicate that in addition to a finite bandwidth of operation, OTA based transimpedance integrators are unable to accurately describe

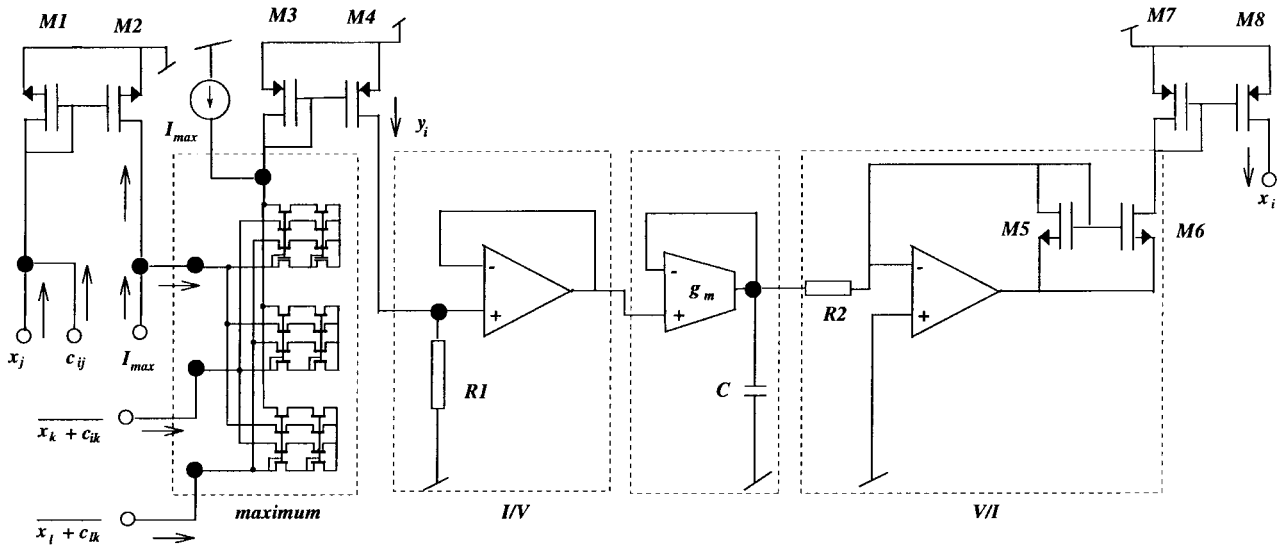


Fig. 4. Unit representing each node in the network.

the low frequency steady state solutions of the fixed-point seeking system. The distortion can be minimized by reducing ω_0. For a given ouput conductance, the available design parameter is value of the capacitor C, which is typically restricted by the size limitations of most VLSI designs to a few picofarads. The second term in (16) provides a robust and computationally efficient method for determining the input voltage, phase distortion, and the noise limitations of these dynamical systems. It can also be used to design predistortion elements for compensation when appropriate.

IV. Analog VLSI Implementation of Shortest Path Computation

A design study of a VLSI implementation, in a 2-μ process, of the above shortest path fixed-point computation has been conducted. The design is an implementation of a system for computing the shortest paths from a source node on a fully connected eight-node network. The circuit uses $n-1$ units, each unit representing a node in the network other than the source node. The source node is represented by the ground plane. Consequently, the output currents of each unit x_i correspond to scaled analogs of the shortest path distances from the source node. Each unit implements the computation of the term $x_i(t)$ implied in (13), using three basic elements:

1) a current-mode summer to calculate $x_j + c_{ij}$;
2) a current-mode minimum circuit based on a fuzzy-logic module [23];
3) a voltage mode integrator to realize the final steady-state shortest path values $x_i(t)$.

The schematic for a single unit is shown in Fig. 4. The inputs to each unit are the interconnection costs c_{ij} which are brought in off chip as buffered current inputs. In addition to the simplicity of summing operations with currents, the use of current-mode computations also results in an area-efficient minimum finding circuit. This element is based on a maximum circuit which uses regeneratively connected current amplifiers (mirrors) to rapidly converge to the max value of a set of input currents. The maximum circuit is modified to form a minimum circuit by computing

$$\mathrm{MIN}(x_j(t)) = I_{\max} - \mathrm{MAX}(-x_j(t) + I_{\max})$$

where $I_{\max}$ is the maximum possible value (largest representable value of x_j) of the unit output currents. This minimum circuit uses much less die area and is typically faster and more accurate than a minimum circuit comprised of a tree of voltage comparators [23]. Recent developments indicate that, using alternative topologies, it is possible to build well-matched maximum circuits that span several die, thus increasing the number of nodes that can be processed reliably [31]. The circuit settles to the maximum of the input currents provided the dc loop gain around each pair of current mirrors is ≥ 1. Fig. 5 shows a schematic of a three-node maximum finding circuit. The complement operation required by the above equation is implemented at the output and input of the maximum circuit by using the reference current $I_{\max}$ and n-channel current sink mirrors. By using an OTA-C integrator, the subtraction called for by the system in (2) is carried out in the same step as the integration. The integrators, buffers, and converters are all designed around split-pole frequency compensated two-stage OTA's.

The edge costs can be brought in off chip as buffered current inputs to the individual units. Fig. 6 shows the interconnection scheme for a four-node fully connected network using the unit elements described above. The fourth (source) node is represented by the ground plane. The system was designed to compute path lengths in the range of 50–700 μA. The OTA-C integrator has a linear range of 40 mv and the I/V and V/I converters perform the necessary compression–decompression. In the case of the V/I converter, the output stage has been designed to sink the maximum allowable current of 700 μA. The resistors used in the V/I and I/V converters are not available in a standard double poly-CMOS process and were modeled as ideal elements for the simulations. The reference current $I_{\max} = 700\ \mu A$ is generated using p-channel current mirrors.

The accuracy of the computational elements is primarily determined by the current mirror matching accuracy. This

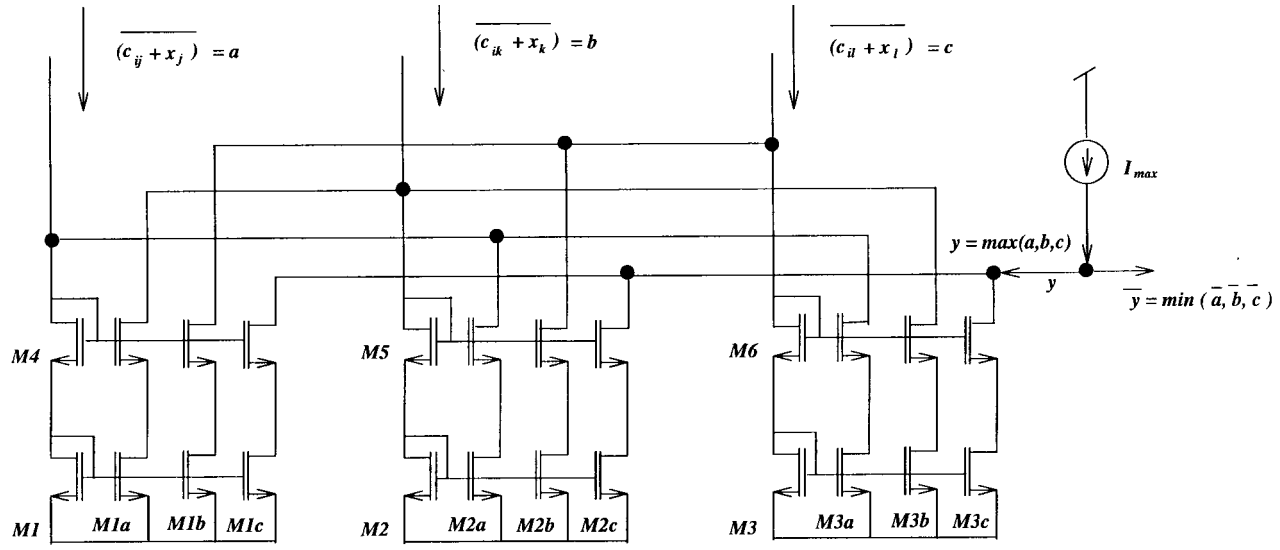


Fig. 5. Three-node maximum finding circuit.

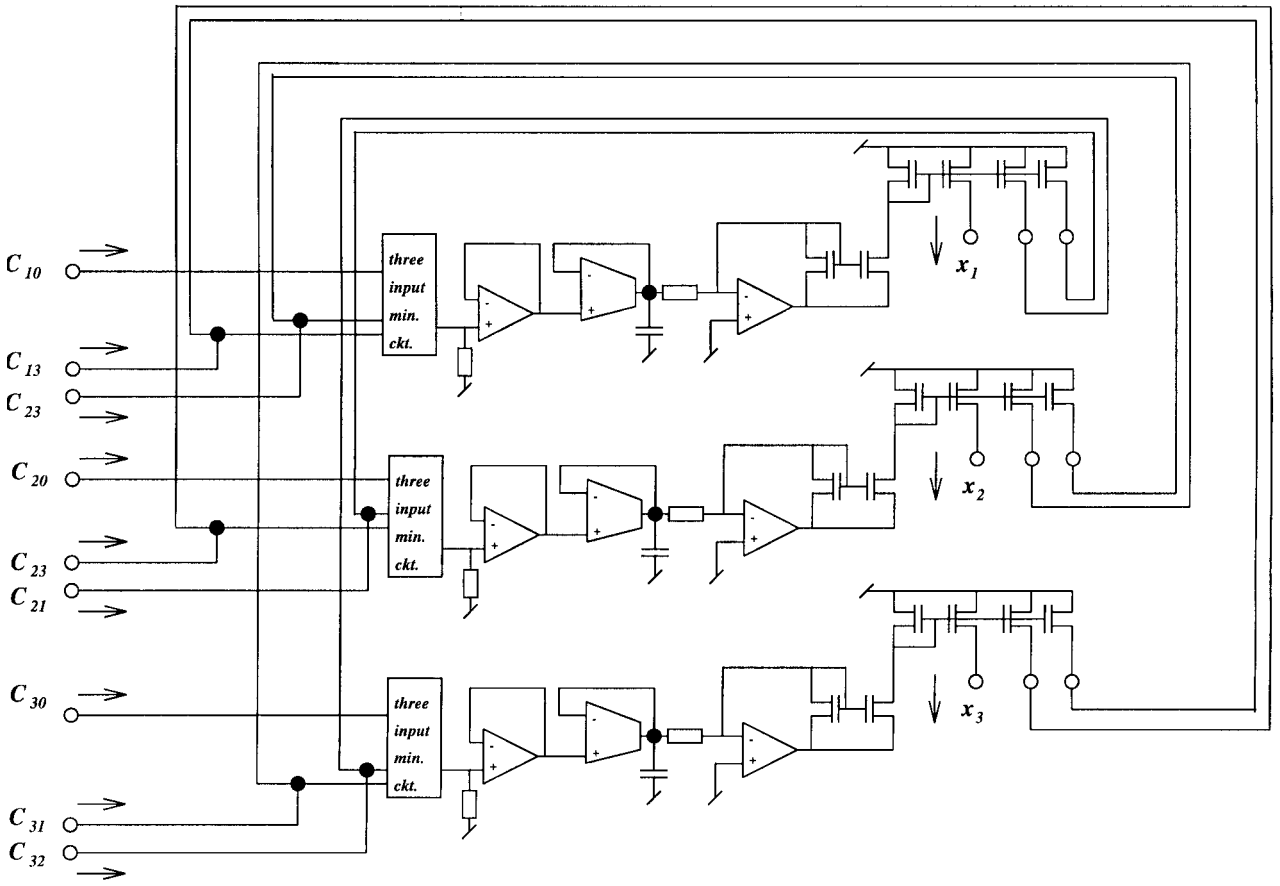


Fig. 6. Shortest path computational structure for four-node fully interconnected network. Source node is assumed to be at ground.

accuracy is a function of the output conductance of the mirrors used. The relatively poor matching accuracy (10 μA) of the given process necessitated the use of cascoded current mirrors at the outputs of the V/I convertors, input stages and in the minimum finding circuit. The mirrors were sized optimally to produce mirrors with a matching accuracy of approximately 1 μA.

The settling time of the maximum finding circuit is a function of the quiescent current and the number of nodes (Fig. 7). The 8 node settling time is seen to be less than 10 ns, which is well under the integration time constant of 1 μs. The settling times of the current mirrors are functions of quiescent current and the input step size. Figs. 8 and 9 show that the worst case settling time is also less than 10 ns. The settling time of the circuit is consequently dominated by the integration time constant.

The eight-node network was simulated using level two SPICE models to account for the finite ouput impedances of

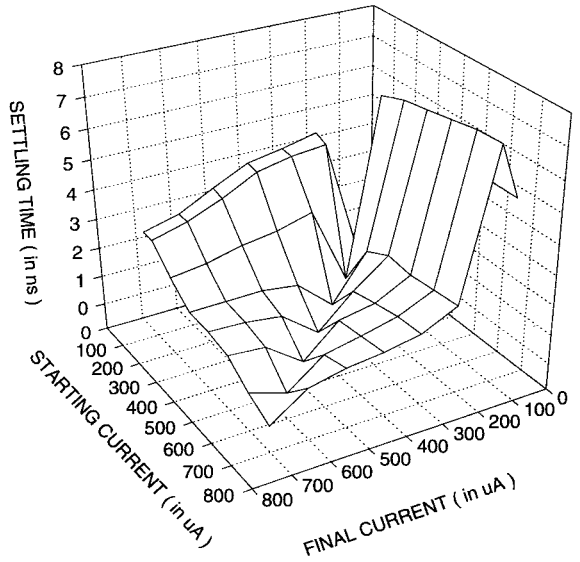


Fig. 7. Settling time for a cascoded comparator as a function of final current and number of inputs.

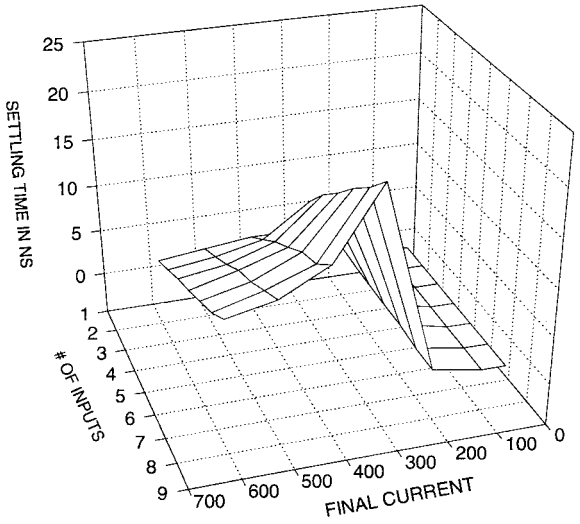


Fig. 8. Settling times for a cascoded two-input comparator.

the active devices. The outputs were accurate to within 10% of the expected values in the worst case. This result is available in 100 ns in all cases. The maximum accuracy is insensitive to path length (i.e., number of edges) and occurs in the middle of the solution range when the current mirrors operate with maximum headroom.

V. Conclusion

Several applications of an analog parallel network for fixed-point computation are discussed, in particular for: 1) solving systems of linear equations; 2) nonlinear programming; 3) dynamic programming; and 4) network-flow computations. Schematic circuits are proposed for representative cases and implementation issues are discussed. A robust computationally efficient model for assessing the accuracy of the dynamic element in implementing any of the above schemes is presented. A new approach to the SPP that always converges to the exact solution in all cases, based on the above fixed-point

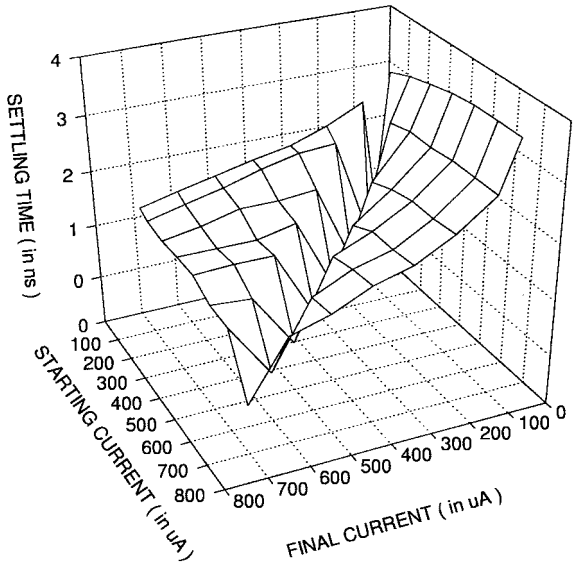


Fig. 9. Settling times for a cascoded p-channel current mirror.

formulation is proposed. A circuit that can be used for an area-efficient analog VLSI implementation for the single-source SPP on a fully connected eight-node network is described in detail. The proposed distributed implementation is able to operate in real time on dynamic networks, when the edge costs vary at a rate slower than the settling time of the system. Finally, comparisons of the convergence rate of the dynamical system with discrete-time systems in general, and with digital implementations of the SPP in particular, are subjects for future research. This research will also include comparisons with alternative analog implementations that use current-mode integrators and switched-capacitor realizations.

Appendix

Proof of Theorem 3.1

Rewrite (6) as

$$\frac{d}{dt}(x(t)-\overline{x}) = (P^T - I)(x(t)-\overline{x}.)$$

Thus

$$x(t)-\overline{x} = \exp((P^T - I)t)x(0) - \overline{x}.$$

Since P is an irreducible matrix, it has a simple eigenvalue one with left and right eigenvectors $(x^*)^T$ and e^T, respectively, and the real part of all other eigenvalues is strictly less than one ([3]). Thus, $(P^T - I)$ has a simple eigenvalue zero with left and right eigenvectors e and x^*, respectively and the real part of all other eigenvalues is strictly negative. Hence, (4) becomes

$$x(t)-\overline{x} = (x^*e + \Phi_t)(x(0)) - \overline{x} = \Phi_t(x(0))$$

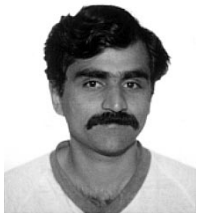
with Φ_t being a matrix with exponentially decaying terms. Thus $x(t)-\overline{x} \to 0$ exponentially at a rate dictated by $\min(|\mathrm{Re}\ \lambda|)$ as λ varies over nonzero eigenvalues of $P^T - I$.

ACKNOWLEDGMENT

K. Soumyanath wishes to thank S. Ritter and J. Von Arx for help in simulations and layout. The authors also wish to thank Dr. S. Zaccheo for several interesting discussions on integrators.

REFERENCES

[1] M. K. M. Ali and F. Kamoun, "Neural networks for shortest path computation and routing in computer networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 931–940, Nov. 1993.
[2] B. Linares Barranco, E. Sanchez-Sinencio, A. Rodriguez-Vazquez, J. L. Huertas, "A modular T-mode design approach for analog neural network hardware implementations," *IEEE J. Solid-State Circuits*, vol. 27, pp. 701–713, May 1992.
[3] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation.* Englewood Cliffs, NJ: Prentice-Hall, 1989.
[4] D. Bertsekas and R. Gallager, *Data Networks.* Englewood Cliffs, NJ: Prentice-Hall, 1992.
[5] V. S. Borkar, "Topics in controlled Markov chains," in *Pitman Research Notes in Mathematics. No. 240.* Harlow, U.K.: Longman, 1991.
[6] V. S. Borkar and K. Soumyanath, "An analog scheme for fixed point computations part I: Theory," *IEEE Trans. Circuits Syst. I*, vol. 44, pp. 351–354, Apr. 1997.
[7] A. Bouzerdoum and T. R. Pattison, "Neural network for quadratic optimization with bound constraints," *IEEE Trans. Neural Networks*, vol. 4, pp. 293–304, Mar. 1993.
[8] R. W. Brockett, "Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems," *Linear Algebra Appl.*, vol. 146, pp. 79–91, 1991.
[9] ———, "Least squares matching problems. Least squares matching problems," *Linear Algebra Appl.*, vols. 122/123/124, pp. 701–777, 1989.
[10] R. W. Brockett and W. S. Wong, "A gradient flow for the assignment problem," in *New Trends in System Theory, Progress in System and Control Theory* G. Conte, A. M. Perdon, and B. Wyman, Eds. Basel, Switzerland: Birkhauser Verlag, 1991, pp. 170–177.
[11] C. Mead, *Analog VLSI and Neural Systems*, Addison Wesley, 1989.
[12] L. O. Chua and L. Yang, "Cellular neural networks: Theory and applications," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 1257–1290, 1988.
[13] L. O. Chua and T. Roska, "The CNN universal machine: An analogic array," *IEEE Trans. Circuits Syst.*, vol. 40, pp. 163–173, Mar. 1993.
[14] L. O. Chua and G. N. Liu, "Non-linear programming without computation," *IEEE Trans. Circuits Syst.*, vol. 31, pp. 210–214, Feb. 1984.
[15] C. Chiu, C. Y. Maa, and M. A. Shanblatt, "Energy function analysis of dynamic programming neural networks," *IEEE Trans. Neural Networks*, vol. 2, pp. 418–426, 1991.
[16] M. T. Chu, "Chu on the continuous realization of iterative processes," *SIAM Rev.*, vol. 30. no. 3, Sept. 1988.
[17] A. Cichocki and R. Unbehauen, "Neural networks for solving systems of linear equations: Part II—Minimax and least absolute value problems," *IEEE Trans. Circuits Syst. II*, vol. 39, pp. 619–633, Sept. 1992.
[18] J. Cronin, *Differential Equations: Introduction and Qualitative Theory*, 2nd ed. New York: Dekker, 1994.
[19] C. B. Garcia and W. I. Zangwill, *Pathways to Solutions, Fixed Points and Equilibria.* Englewood Cliffs, NJ: Prentice–Hall, 1981.
[20] B. Gilbert, "A four quadrant analog divider/multiplier with 0.01% distortion," *ISSCC Dig. Tech. Papers*, Feb. 1983, pp. 248–249.
[21] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems.* London, U.K.: Springer Verlag, 1994.
[22] M. P. Kennedy and L. O. Chua, "Neural networks for nonlinear programming," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 231–242, Mar. 1990.
[23] M. Sasaki, T. Inoue, Y. Shirai, and F. Ueno, "Fuzzy multiple input maximum and minimum circuits in the current mode and their analysis using bounded difference equations," *IEEE Trans. Comput.*, vol. 39, pp. 768–774, June 1990.
[24] E. Polak, *Optimization.* New York: Springer–Verlag, 1997
[25] A. Rodriguez-Vazquez, S. Espejo, R. Dominguez-Castro, J. L. Huertas, and E. Sanchez-Sinencio, "Current-mode techniques for the implementation of continuous and discrete-time cellular neural networks," *IEEE Trans. Circuits Syst. II*, vol. 40, pp. 132–146, Mar. 1993.
[26] A. Rodriguez-Vazquez, R. Dominguez-Castro, A. Rueda, Jose L. Huertas, and E. Sanchez-Sinencio, "Nonlinear switched-capacitor 'neural' networks for optimization problems," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 384–398, Mar. 1990.
[27] G. R. Sell, *Topological Dynamics and Ordinary Differential Equations.* London, U.K.: Von Nostrand Reinhold, 1971.
[28] S. E. Ritter and K. Soumyanath, "An analog parallel distributed solution to the shortest path problem," in *Proc. ICCD '90*, Cambridge, MA, Sept. 1990, pp. 130–134.
[29] K. Soumyanath and J. Von Arx, "An analog parallel processor for the dynamic programming paradigm," *Fifth Ann. IEEE Int. ASIC Conf. Exhibit*, Rochester, NY, 1992, pp. 557–560.
[30] S. Sudarshanan and M. Sunderashan, "Exponential stability and a systematic analysis of a neural network for quadratic minimization," *Neural Networks*, vol. 4, no. 5, pp. 599–613, 1991.
[31] T. Serrano-Gotarredona and B. Linares-Barranco, "A high precision current mode WTA-MAX circuit with multi-chip compatibility," *IEEE J. Solid-State Circuits*, vol, 33, pp. 280–286, Feb. 1998.
[32] T. Yoshizawa, "Stability theory of Liapunov's second method," *Math. Soc. Japan*, 1996.
[33] J. L. Wyatt Jr., *et al.*, "Analog VLSI systems for image acquisition and fast early vision processing," *Int. J. Comput. Vision*, vol. 8, no. 3., pp. 599–613, 1992.
[34] L. Zhang and S. C. A. Thomopoulos, "Neural network implementation of the shortest path algorithm for traffic routing in communication networks," in *Proc. Int. Joint Conf. Neural Networks*, June 1989, pp. II 591.



K. Soumyanath (M'95) received the B.E. degree in electronics and communication engineering from the Regional Engineering College, Tiruchirappalli, India, in 1979, the M.S. degree in electronics from the Indian Institute of Science, Bangalore, India, in 1985, and the Ph.D. degree in computer science from the University of Nebraska, Lincoln, in 1993.

He was a member of the faculty at Tufts University Medford, MA, until 1995, where he served as the Director of the ARPA-supported Program in Mixed Signal IC design for Defense Engineers In this capacity he collaborated on mixed-signal IC projects with several Boston-area companies, including Raytheon, Analog Devices, Loral, and Textron. He was also a consultant for Sipex Corporation, where he was involved in the design of several precision, monolithic A/D convertors. Since 1996 he has been with Intel Corporation, Hillsboro, OR, where he manages the high-performance circuits research program in the Circuits Research Laboratory. His research interests are in mixed-signal circuit design, graph theory, and CAD algorithms and classical Tamil poetry.

Dr. Soumyanath served as Chair of the Design Sciences Task Force for the Semiconductor Research Corporation in 1998.

Vivek S. Borkar received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Mumbai, India, in 1976, the M.S. degree in systems and control from Case Western Reserve University, Cleveland, OH, in 1977, and the Ph.D. degree in electrical engineering and computer science from the University of California, Berkeley, in 1980.

After working with the Tata Institute for Fundamental Research, Bangalore Center, Bangalore, India, for many years, he joined the Indian Institute of Science, Bangalore, where he is currently an Associate Professor in the Department of Computer Science and Automation. He has held visiting positions at T.H.T., the Netherlands, The Massachusetts Institute of Technology, Cambridge, MA, the University of Maryland, College Park, and the University of California, Berkeley. His research interests are in stochastic control and optimization algorithms.

Dr. Borkar is a member of the American Mathematics Society, a life member of the Indian Society for Probability and Statistics and the Operations Research Society of India, and a Fellow of the Indian Academy of Sciences and the Indian National Science Academy. He was the recipient of the Homi Bhabha Fellowship in 1994–1995 and the S. S. Bhatnagar Award in 1992 and was a cowinner of the Best Transactions Paper Award of the IEEE Control Systems Society in 1982.