16 ANALOG VLSI PARALLEL STOCHASTIC OPTIMIZATION FOR ADAPTIVE OPTICS

R. Timothy Edwards[1], Marc Cohen[1], Gert Cauwenberghs[1],
Mikhail A. Vorontsov[2], and Gary W. Carhart[2]

[1]Department of Electrical and Computer Engineering
Johns Hopkins University, Baltimore MD 21218
{tim,marc,gert}@bach.ece.jhu.edu

[2]Intelligent Optics Laboratory
Army Research Laboratory, Adelphi MD 20783
{vorontsov,gary}@iol.arl.mil

Phase distortion in wavefront propagation is one of the key problems in optical imaging and laser optics applications. We present a hybrid VLSI and optical system for real-time adaptive phase distortion compensation. The system implements stochastic perturbative gradient descent, performing "model-free" adaptation independent of the specifics of both the distorting optical medium and the control elements. A mixed-mode VLSI system interfaces with a liquid-crystal spatial light modulator (SLM), controlling 127 parameters in parallel which adjust the wavefront phase profile. On-chip CMOS circuitry performs parallel stochastic perturbative gradient descent/ascent of an externally supplied optimization metric, e.g. *a direct measure of image quality. Parallel random perturbations of the parameters are generated locally, and the resulting differential performance measure is locally correlated with the perturbations to generate parallel parameter updates, implementing a random-direction, stochastic approximation version of gradient descent. Additional on-chip circuitry provides for liquid-crystal AC modulation and adaptive biasing of the parameter mean. Experimental results on adaptive laser beam focusing are included.*

16.1 INTRODUCTION TO ADAPTIVE OPTICS

Many optical systems, such as imaging systems or laser beam communication systems, suffer performance degradation due to distortions in the optical wavefront. As

an optical wave propagates through an optically inhomogeneous medium such as the atmosphere, differences in the index of refraction along the propagation path cause variations in the speed of light propagation, which lead to *phase distortions* (*aberrations*). The technique used to compensate these wavefront distortions is *adaptive optics*.

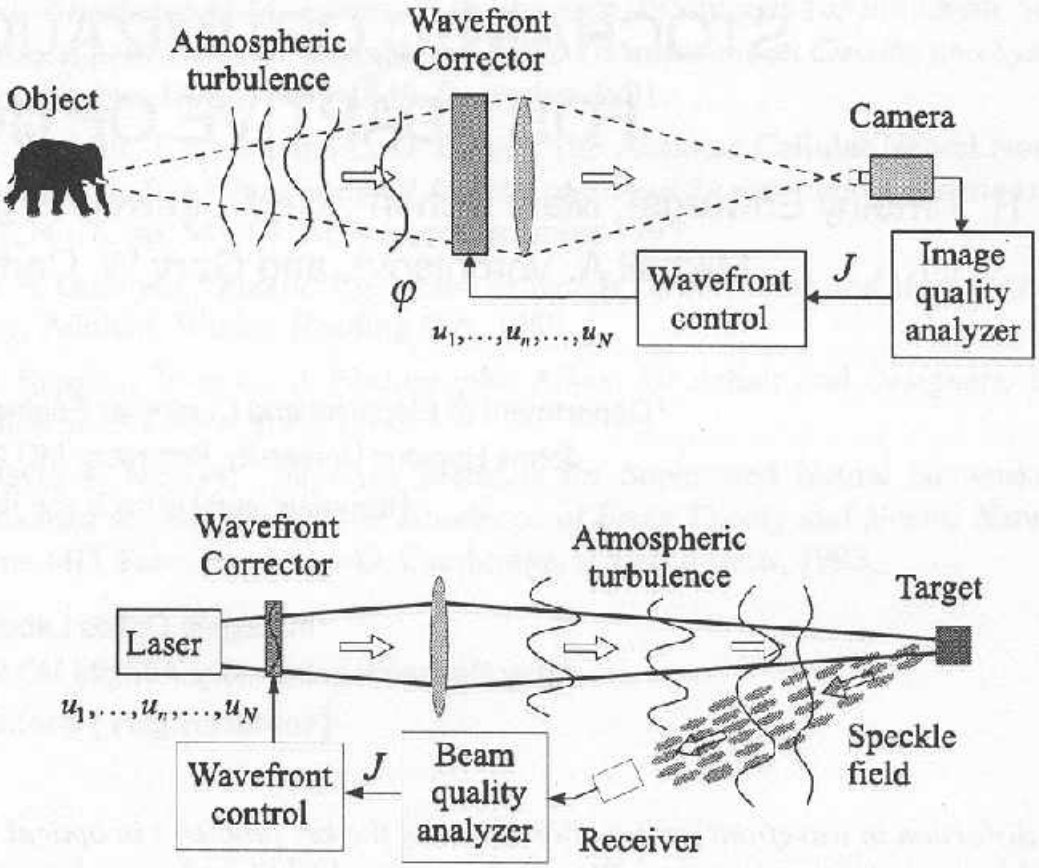

Figure 16.1 General schematics for adaptive imaging (top) and laser beam focusing (bottom) systems based on direct optimization of a system performance metric.

"Adaptive optics" is a term describing optical systems whose characteristics can be modified by dynamically changing the wavefront [1]. The fundamental ideas of adaptive optics were first formulated by Linnik [2] and Babcock [3] in the late 1950s, but practical implementation of these ideas was impeded at that time by the lack of suitable wavefront control techniques.

Since that time the situation has changed and the demands for adaptive wavefront control have increased dramatically. In addition to traditional astronomy, perhaps the primary stimulus for adaptive optics, an important drive to the development of adaptive optics techniques was the need for laser systems capable of successfully operating in the atmosphere: laser communication systems, laser lidars, active imaging, and laser beam focusing systems. Progress in laser technology, laser medicine, fiber optics, and optical information processing also required substantial improvements in laser

beam quality, and especially wavefront quality as achieved through the use of adaptive wavefront correction techniques.

The main principles of adaptive optics were formulated by studying the classical problem of phase distortion compensation of a wave originating from a point source (star) and then propagating through a phase distorting medium. The assumption of a point source is the foundation for the most widely used adaptive optics control algorithm, *phase conjugation* [4, 5, 6]. The phase conjugation algorithm requires knowledge of the wavefront aberration $\varphi(\mathbf{r}, t)$, where t is time and $\mathbf{r}$ is a vector in the plane orthogonal to the direction of wave propagation. The phase conjugation correction $u(\mathbf{r}, t) = -\varphi(\mathbf{r}, t)$ introduced by a deformable (adaptive) mirror compensates the phase aberrations to obtain an undistorted image of the point source. In some applications, most notably astronomical observations, phase conjugation works quite well and has been used with Earth-bound telescopes to greatly enhance photographic images [13, 14].

One problem with the phase conjugation technique is that the phase modulation $\varphi(\mathbf{r}, t)$ can not be directly measured and must be reconstructed from intensity data measurements obtained from a wavefront sensor. Retrieval of phase information from wavefront sensor data requires extensive and time-consuming calculations that significantly increase adaptive system cost and complexity. Another problem is the assumption of a point source. There are a number of important applications where the assumptions upon which phase conjugation is based are not valid and other wavefront control strategies should be considered: for example, atmospheric imaging of extended objects, laser beam focusing on targets with rough surfaces, and the adaptive compensation of nonlinear effects in the propagation of high-power laser beams.

There is an alternative and more general approach that does not require either determining phase information from the intensity, or a point source image. In an adaptive system based on direct optimization of a system performance metric, the control algorithm can be made independent of the system model ("model-free" or "blind" optimization [7]). A schematic illustration of this concept for atmospheric imaging systems and laser beam focusing systems is shown in Figure16.1. The measured quality metric $J = J(\mathbf{u})$ is a function of the control parameters $\mathbf{u} = \{u_1, \ldots, u_n, \ldots, u_N\}$ of the wavefront corrector. The parameters u_n could be, for instance, voltages applied to electrodes of a deformable mirror or a spatial phase modulator.

To optimize metric J, various methods approximating gradient descent based on physical measurements can be applied [8]. For instance, in a sequential perturbation method the partial derivatives $\partial J/\partial u_n$ are estimated in sequence, by applying small perturbations Δu_n to the control parameters u_n one at a time and measuring corresponding changes ΔJ_n to the quality metric [9]. The critical issue with such techniques is the adaptation speed. The convergence time of the sequential optimization process increases more than linearly with the number of control elements N.

The increase in adaptation time caused by sequential perturbations can be reduced using the multi-dither technique [10]. In this method perturbations in the form of harmonic signals having small amplitudes and different dithering frequencies are simultaneously applied to all wavefront corrector electrodes. Estimates of the gradient

components can be obtained by performing synchronous signal detection in each control channel. With this technique the increase in N requires a corresponding increase in the system frequency bandwidth, typically limited by system hardware (primarily by wavefront corrector frequency bandwidth) [4]. Accordingly, the number of control channels for the multi-dither technique does not exceed 10^2.

Among model-free optimization techniques that have recently appeared stochastic parallel gradient descent is perhaps the most promising for adaptive optics applications [11, 12]. The greatest potential benefit for adaptive optics applications comes from the fact that the parallel perturbation technique is well-suited for analog VLSI implementation [20, 21, 22, 28]. The technique has been demonstrated in analog VLSI, both in sequential [23] and parallel [24, 25, 26] forms.

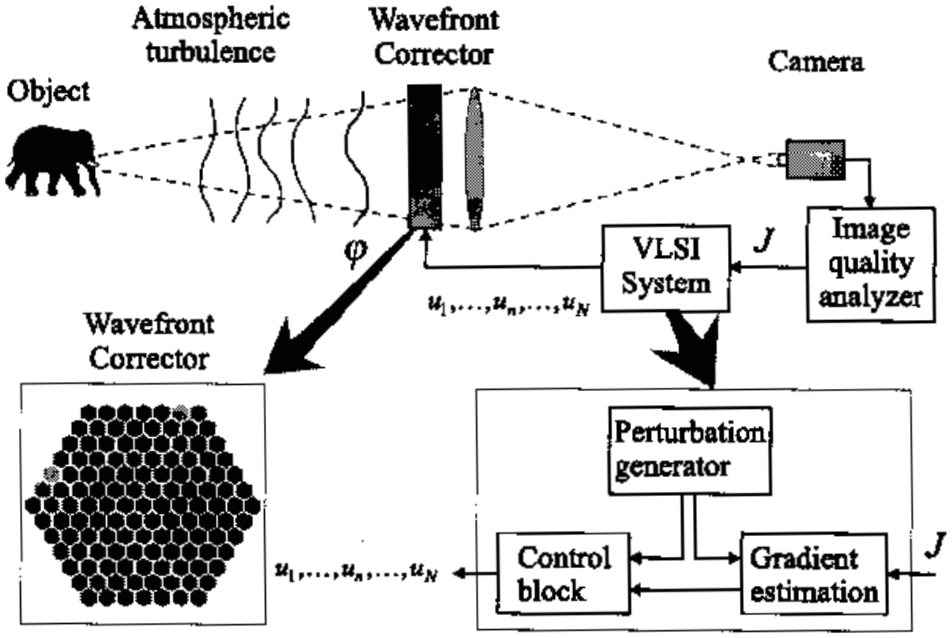

Figure 16.2 Block diagram for an adaptive hybrid VLSI-optical imaging system based on image quality metric optimization using stochastic parallel gradient descent.

In this chapter we report on the first VLSI hardware implementation of a system for adaptive optics based on this method. Figure 16.2 illustrates our approach. Section 16.2 briefly reviews gradient descent in relation with model-free techniques, and presents the parallel stochastic perturbative gradient descent equations. Section 16.3 gives a detailed description of an efficient implementation of the algorithm in analog VLSI hardware. Section 16.4 combines the analog hardware with optics to create an adaptive laser beam focusing system. In Section 16.5, we characterize performance of adaptive wavefront distortion correction using experimental results obtained from this system.

16.2 MODEL-FREE STOCHASTIC PARALLEL OPTIMIZATION

16.2.1 Gradient Descent

One possible technique to adaptively optimize the quality metric $J = J(\mathbf{u})$ is by *gradient descent* of J in the control parameters u_n of the wavefront corrector (see Figure 16.1).

Gradient descent algorithms incrementally adjust the parameters u_n based on instantaneous estimates of the gradient components $\partial J/\partial u_n$ at each iteration m:

$$u_n^{(m+1)} = u_n^{(m)} - \gamma \frac{\partial J}{\partial u_n} \tag{16.1}$$

where γ (not necessarily constant) determines the size of the step in the direction of the steepest gradient from the current position on the surface of the function $J(\mathbf{u})$.

In principle, the gradient components can be calculated from a model of the control system. Often such a model is not available, or too complex to estimate or evaluate. This is particularly true in the case of optical phase aberration, for which a precise model would have to account for the index of refraction profile over the entire propagation path. Even if this information is available, computing the gradient from such a model would be computationally too expensive. This motivates the use of *model-free optimization* (also known as *blind optimization*) techniques [7].

16.2.2 Model-Free Optimization

Model-free stochastic optimization, loosely inspired by principles of adaptive neural computation [7], provides a means to perform parallel descent of any scalar function without *a priori* knowledge about the internal workings of the control system. The technique treats the system as a "black box" that allows access to its internal parameters u_n so they can be tweaked, or *perturbed*, and returns an estimate of the performance metric J on demand (Figure 16.2).

The general principle is rooted in theory of *stochastic approximation* [16, 17, 18, 19], which formulates conditions under which a stochastic procedure, applying incremental parameter adjustments, converges to a minimum of a scalar optimization function based only on discrete and noisy evaluations of that function. The technique is *model-free*, since it requires no knowledge of how each internal control parameter independently affects the system output. Instead, it relies on direct scalar measurements of the system output under perturbation of the parameter(s), plus local information on how (*i.e.*, in which direction) the parameter vector was perturbed, to estimate the gradient component in the direction of the parameter perturbation.

The stochastic perturbative optimization operates on the same basic principle as standard gradient descent (16.1), except that a *perturbation* Δu_n is applied to each parameter. The perturbation and the resulting response of the system, ΔJ, are used as approximations to the true gradient components.

The convergence rate of a method that uses scalar function evaluations of course cannot be as fast as pure gradient descent, which makes use of vector information as provided by the true gradient. However, *parallel stochastic perturbation* of the

parameters yield a convergence rate significantly faster than *sequential perturbation methods*. In fact, for mutually uncorrelated perturbations the gain in speed is a factor $\sqrt{N}$, where N is the number of parameters [20]. For optical aberration compensation, the rate of convergence can be further improved by modulating the correlation between the perturbations of adjacent control parameters in a phase modulator [12].

16.2.3 Parallel Stochastic Perturbative Gradient Descent

In parallel stochastic optimization, a random ensemble of perturbations is applied to all N control parameters simultaneously. The original state of the control system is restored after each perturbation, and the parameters adapt along the direction of the perturbation vector by an amount proportional to the measured ΔJ value, which can be positive or negative to indicate improvement or deterioration of the optimization function in that direction.

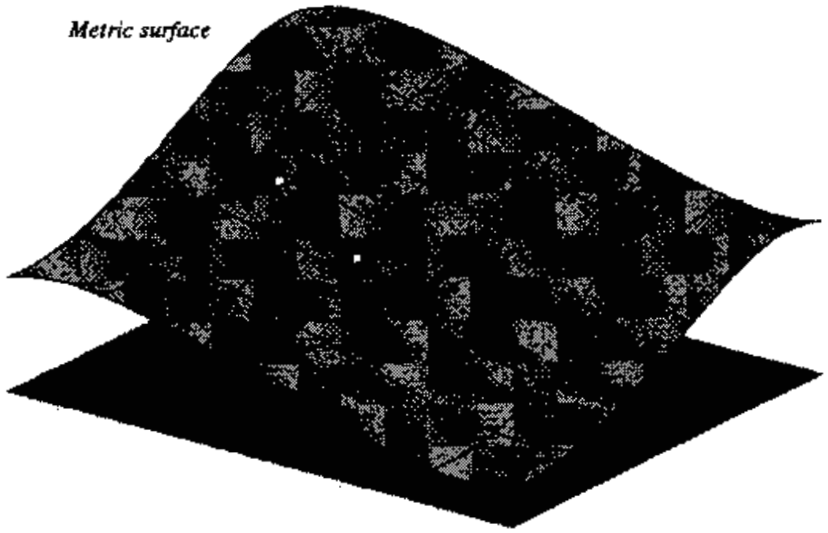

Figure 16.3 Stochastic approximation gradient descent explores the performance metric surface J at a few points around the current parameter state $\mathbf{u}$ and chooses the direction in the subspace resulting in steepest descent of J.

In the simplest scenario, the perturbed metric ΔJ is given in terms of the perturbations Δu_n as

$$\Delta J = J(u_1 + \Delta u_1, u_2 + \Delta u_2, \ldots, u_N + \Delta u_N) - J(u_1, u_2, \ldots, u_N) \qquad (16.2)$$

from which, by Taylor expansion,

$$\frac{\Delta J}{\Delta u_n} \approx \frac{\partial J}{\partial u_n} + \sum_{\ell \neq n}^{N} \frac{\partial J}{\partial u_\ell} \cdot \frac{\Delta u_\ell}{\Delta u_n}. \qquad (16.3)$$

Under certain conditions, when the perturbations Δu_n are random and statistically independent, the second term of Equation (16.3) reduces in expectation to zero, yielding

a good approximation to the true gradient for large N. This is true in particular when the the perturbations are Bernoulli distributed, of fixed amplitude but random sign, $\Delta u_n = \pm\sigma$ [20, 21], but also valid for other, continuous-valued distributions of random perturbations [18]. More accurate results are obtained with a two-sided approximation to the gradient, using a differential measurement under complementary perturbations $\pm\frac{1}{2}\Delta u_n$, yielding the following simple implementation:

$$\begin{aligned} u_n^- &= u_n^{(m)} - \sigma\pi_n^{(m)}, \quad \forall\, n, m \\ u_n^+ &= u_n^{(m)} + \sigma\pi_n^{(m)} \\ \Delta u_n &= u_n^+ - u_n^- = 2\sigma\pi_n^{(m)} \\ \Delta J &= J^+ - J^- \\ u_n^{(m+1)} &= u_n^{(m)} - \gamma\frac{\Delta J}{\Delta u_n} \end{aligned} \tag{16.4}$$

where the perturbation signals $\pi_n^{(m)}$ are generated from a Bernoulli random distribution:

$$\pi_n^{(m)} = \pm 1, \quad \Pr(\pi_n^{(n)} = +1) = 0.5, \quad \forall\, n, m \tag{16.5}$$

with uncorrelated statistics across parameters and over time:

$$E(\pi_n^{(m)}\pi_p^{(q)}) = \delta_{np}\,\delta_{mq}, \quad \forall\, n, m, p, q\,. \tag{16.6}$$

Considering that the perturbation amplitudes $|\Delta u_n|$ are identical (2σ) for all perturbations, the specification of the update (16.4) is further simplified:

$$u_n^{(m+1)} = u_n^{(m)} - \gamma'\pi_n^{(m)}\Delta J, \tag{16.7}$$

where the constant $\gamma' = \gamma/2\sigma$ absorbs both learning rate and perturbation strengths.

16.2.4 Implementation

First, a random or pseudorandom binary sequence generator creates a vector of values $\pi_n^{(m)} = \pm 1$. The controller uses the random vector to perturb each of the parameters a distance $\pm\sigma$ from its current state. This new state $S = -1$ is the *negative-phase perturbation*. The controller then waits until the perturbed state propagates through the system and produces a valid output J^-. After the perturbation, the system returns to its original state $u_n^{(m)}$ and then is perturbed in the opposite direction to the first perturbation. This new state $S = +1$, is the *positive-phase perturbation*. Again, the controller waits for the perturbation to propagate through the system and produce a valid output J^+. At this point, two points on the performance metric surface are known, and a gradient estimate in the direction of the random perturbation vector can be computed. The parameters are adapted in this direction by an amount proportional to the gradient, ΔJ scaled by a gain factor γ', and multiplied by the perturbations π_n. The multiplication operation is straightforward because perturbations are constant-amplitude, $\pi_n = \pm 1$.

16.3 MIXED-MODE VLSI HARDWARE

Stochastic perturbative gradient descent works well with analog control hardware where nonidealities make the system difficult to model and the exact gradient impossible or hard to compute [23, 24, 25]. Analog components are both nonlinear and subject to mismatch. The equations governing the behavior of the circuit depend on factors such as local temperature which are neither well-controlled nor easily measured. A model of such a system ultimately becomes much more complicated than the system itself. Model-free adaptation is not merely a convenience; it is a necessity.

Before considering the application of stochastic gradient descent to adaptive optical phase correction, we will examine the basic design of a mixed-mode VLSI system which performs perturbative learning. The design makes use of a compact and accurate charge pump design [27] augmented with perturbation and update circuitry [25, 26]. The general design methodology is described in [28]. The system consists of five main parts:

1. an analog memory circuit which maintains each control parameter u_n
2. a circuit which perturbs the parameter
3. a circuit which adapts the parameter
4. an output driver specific to the application
5. a pseudorandom vector sequence generator

The first four parts form a module which is duplicated for each parameter of the system, while the random number generator provides input to each module in parallel. This is depicted in Figure 16.4.

16.3.1 Parallel Random Perturbation Generator

The first step in the algorithm is to generate the random vector $\pi_1 \ldots \pi_N$ which determines the direction of perturbation of each parameter. A pseudorandom sequence suffices, and the period of the sequence does not have to be especially long; however, it makes sense to maximize the periodicity for the number of parameters available. A typical way to implement a pseudorandom bit generator is a Linear Feedback Shift Register (LFSR) of a specific length, from which the values of the MSB (Most Significant Bit) and one other bit at cycle n are exclusive-or'd together to generate the LSB (Least Significant Bit) value for cycle $n + 1$. Tables are widely available which show what register lengths and taps yield *maximal-period* sequences (sequences which cycle through all 2^N possible combinations). A shift register implementation alone, however, does not suffice, because the sequences over space are correlated over time. One solution to decorrelate the bit streams is given in [29], and a simple solution using two independent counter-propagating LFSRs is given in [26]. A slightly more efficient alternative is to fold a longer LFSR in half and compute the exclusive-or of each bit of the top and bottom halves of the register, as shown in Figure 16.5. The figure shows 19 parameter outputs, but because a length-38 register cannot form a maximal-period

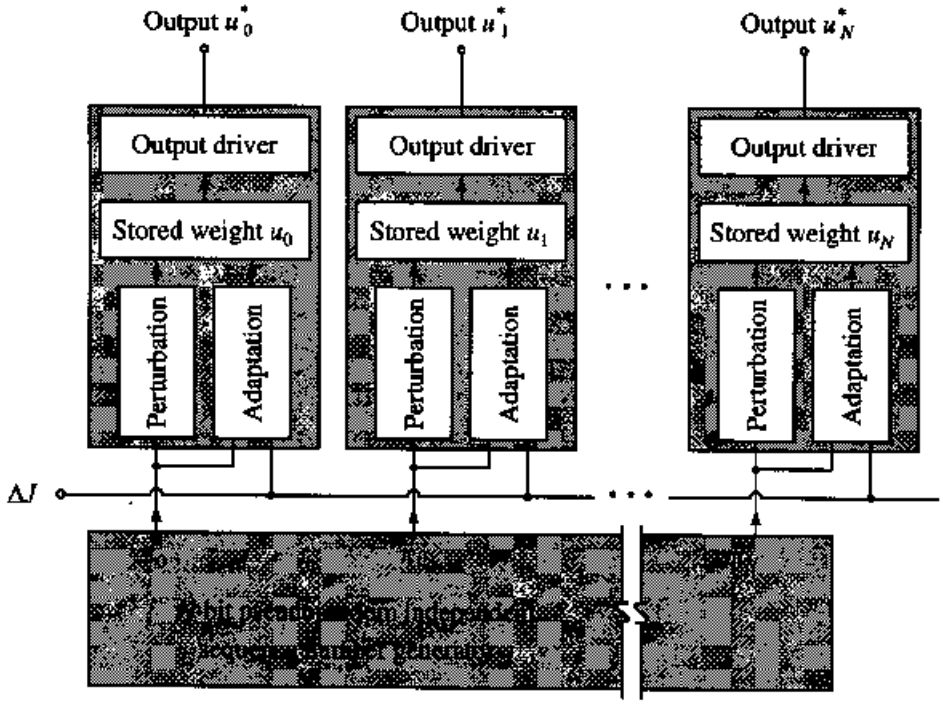

Figure 16.4 System block diagram.

sequence, the register length is increased by one, forming a length-39 register with a period of 2^{39}, or 549,755,813,887.

The multiplexer at the input stage allows a random seed to be loaded into the system from an external source SR_in, when SR_override is selected. The design also allows to bypass the generated random perturbations with a uniform identical perturbation, as controlled by SR_All. This feature is particularly helpful for parameter renormalization in the adaptation process as motivated and described in Section 16.4.1.

16.3.2 Parameter Storage

The model-free aspect of the system alleviates most of the concerns normally associated with analog memory. The analog memory does not have to be calibrated or even particularly accurate. Its value may leak or drift as long as it does so at a rate slower than the rate of adaptation. Since adaptation is continuous, long-term storage is not required, and the memory can be implemented as a charge held on a capacitor, so that each control parameter is encoded as a voltage. In Figure 16.6, capacitor C_1 stores the parameter u as a voltage y, which is a high-impedance node buffered from the output node by amplifier A_1.

16.3.3 Parameter Perturbation

To explore the state space around its present position, the memory value y must be able to be perturbed, but more importantly, it must be able to return to its original state after the perturbation is removed. This requirement is, in fact, easy to satisfy by applying

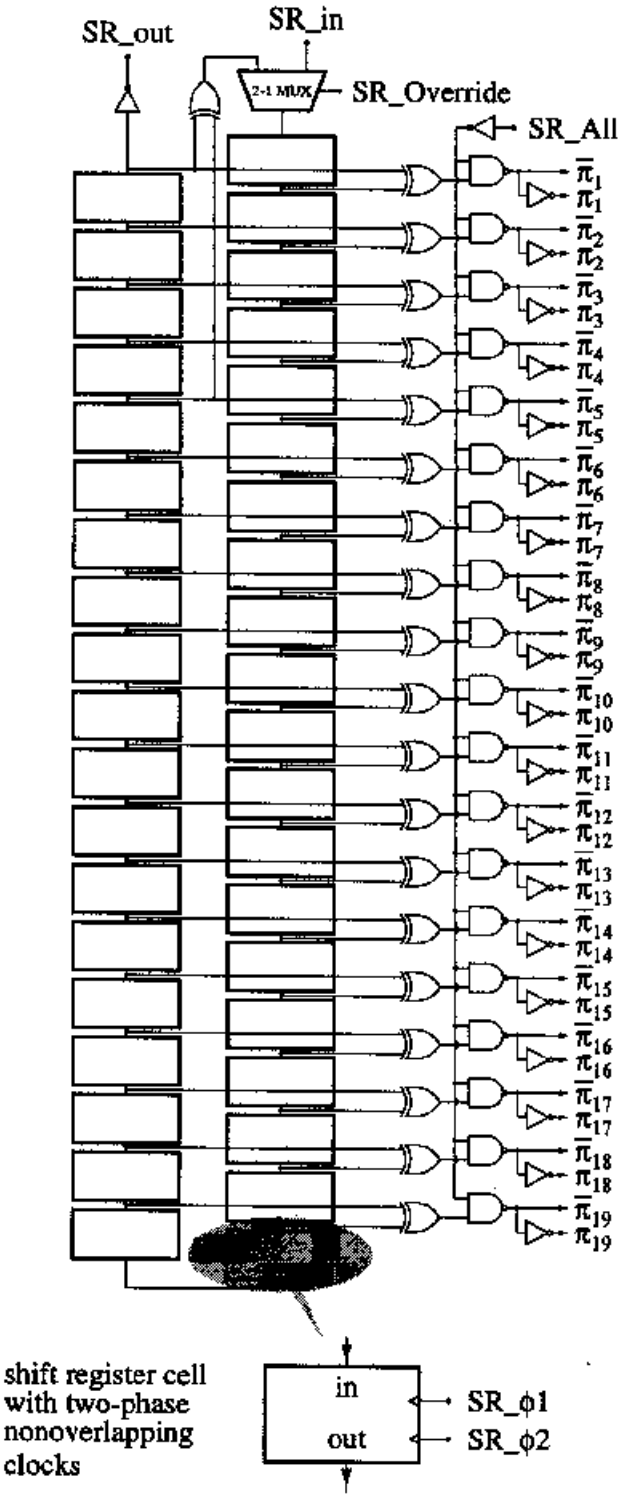

Figure 16.5 Folded, decorrelated pseudorandom bit vector sequence generator, with load and override functions.

the necessary perturbation through *capacitive coupling*. The AC coupling ensures that the perturbation has no permanent effect on the parameter.

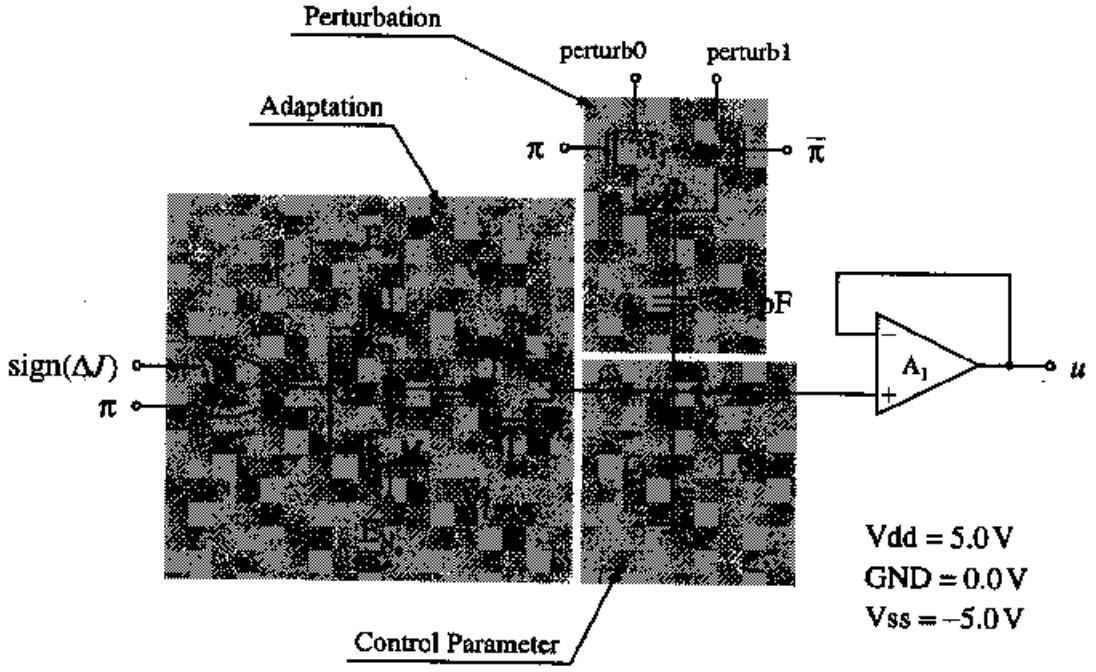

Figure 16.6 The circuit for encoding, perturbing, and adapting a single parameter [28].

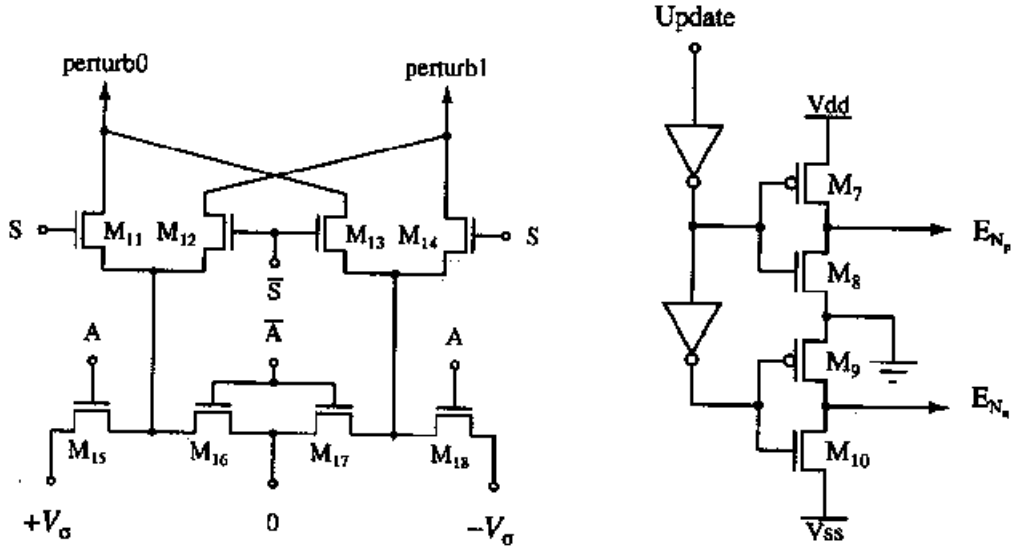

Figure 16.7 Global control signals applied to all channels. *Left:* Cycle control for system perturbation. *Right:* Update controller for adaptation circuit.

A	S	π	Φ
0	X	X	0
1	-1	-1	$+V_\sigma$
1	-1	$+1$	$-V_\sigma$
1	$+1$	-1	$-V_\sigma$
1	$+1$	$+1$	$+V_\sigma$

Update	sign(ΔJ)	π	w
0	X	X	0
1	-1	-1	Vss
1	-1	$+1$	Vdd
1	$+1$	-1	Vdd
1	$+1$	$+1$	Vss

Table 16.1 *Left:* Perturbation voltage Φ as a function of the algorithmic cycle and the perturbation component π. *Right:* Charge pump source-switched voltage w as a function of π and the differential metric ΔJ. "X" indicates a "don't-care" condition.

16.3.4 Differentially Perturbed Metric Evaluation

The stochastic perturbative gradient descent algorithm is applied each update cycle over a sequence of four phases as determined by control values A (for perturbation amplitude) and S (for perturbation phase sign). The global control values A, S, and the local random bit π determine the perturbation voltage Φ applied to the bottom of the coupling capacitor C_2 according to Table 16.1 (left), which is implemented by the array of MOS switches shown in Figure 16.7 (left) and Figure 16.6.

During the first phase of the clock cycle $A = 0$, $S = -1$, the pseudorandom sequence generator is clocked to produce a new vector π. On the next phase, the *negative-phase perturbation*, S remains -1 and $A = 1$, such that the parameter voltage u is shifted by a value

$$u_n^- = u_n^{(m)} + \frac{C_2}{C_1 + C_2}\Phi^- \tag{16.8}$$

where

$$\Phi^- = -\pi\, V_\sigma. \tag{16.9}$$

At this point in the algorithm, the effect of the perturbation is allowed to propagate through the system and produce a metric value J^-. This metric value is captured via A/D or sample-and-hold. After capturing the metric, the controlling process sets $S = 1$ and the system performs the *positive-phase perturbation*. On this cycle,

$$u_n^+ = u_n^{(m)} + \frac{C_2}{C_1 + C_2}\Phi^+ \tag{16.10}$$

where

$$\Phi^+ = \pi\, V_\sigma. \tag{16.11}$$

Again, the effect of the perturbation is allowed to propagate through the system, and the new metric value is captured and the first metric value subtracted from it.

This gradient estimate ΔJ, scaled by γ', is split into *sign* and *magnitude* components which are distributed separately to all adaptive parameter elements in parallel. The sign component is correlated locally (through exclusive OR digital logic) with the perturbation value as $\pi \cdot \text{sign}(\Delta J)$ to determine the direction of the parameter update. The magnitude component $\gamma' \, |\Delta J|$ determines the size of the update.

16.3.5 Parameter Update

Adaptation is performed by the *charge pump* circuit of Figure 16.6 and the update controller of Figure 16.7 (right) [28]. The charge pump operates in one of three conditions: If node w is coupled to Vdd, M_3 becomes a current source and leaks current onto the parameter voltage node, driving the parameter voltage up. If node w is coupled to Vss, M_4 becomes a current sink and leaks current off of the parameter voltage node, driving the parameter voltage down. When w is held at system ground, halfway between Vdd and Vss, then both M_3 and M_4 are disabled and the parameter voltage node maintains a high-impedance, low-leakage state. *Source-switching* the transistors in this manner practically eliminates charge injection into node y [27] and also reduces subthreshold leakage in the off-state.

The single-transistor current sources M_3 and M_4 are not linear with respect to their control voltages V_{UPD_p} and V_{UPD_n}. This complicates the task of applying the magnitude $|\Delta J|$ to the adaptation through voltage control. Instead, if voltages V_{UPD_p} and V_{UPD_n} are kept *fixed*, then the current through each is constant during adaptation, and the parameter voltage drops at a constant rate per unit time. This fact allows the magnitude of the update to be applied linearly by making the length of the "Update" pulse proportional to $|\Delta J|$. The time will of course be quantized to the resolution (master clock frequency) of the controlling processor, but this quantization can be made arbitrarily small by proper adjustment of V_{UPD_p} and V_{UPD_n}. The charge-pump design easily allows quantization to microvolt resolutions [28].

After the update is finished, the controlling processor sets S and A back to their initial values ($S = -1$, $A = 0$), and the cycle repeats.

16.4 ADAPTIVE PHASE DISTORTION COMPENSATION

16.4.1 Liquid crystal spatial light modulator control

The mixed mode VLSI chip implementing the parallel stochastic perturbative gradient optimization algorithm was manufactured through MOSIS. A micrograph of the "AdOpt" chip is given in Figure 16.9. The system contains seven VLSI chips, each providing stochastic parallel gradient descent optimization of 19 control parameters (see Figure 16.8, bottom left).

The system was designed to control a multi-electrode liquid crystal HEX-127 spatial light modulator (SLM) from Meadowlark Optics, Inc. The SLM consists of two glass plates with conductive layers (transparent electrodes) on one side, and a thin parallel aligned layer of nematic liquid crystal between glass plates. One conductive layer has a form of an array with 127 independently addressed hexagonal-shaped electrodes. The liquid crystal geometry is shown at the bottom left corner of Figure 16.2. Each cell is 1.15mm in diameter with 36μm spacing. Since liquid crystal is birefringent, the applied voltage induces a reorientation of the crystal's optical axis resulting in modulation of the incident light wavefront or polarization rotation. If the polarization of the incident light is parallel to the liquid crystal optical axis, the output light experiences pure phase modulation.

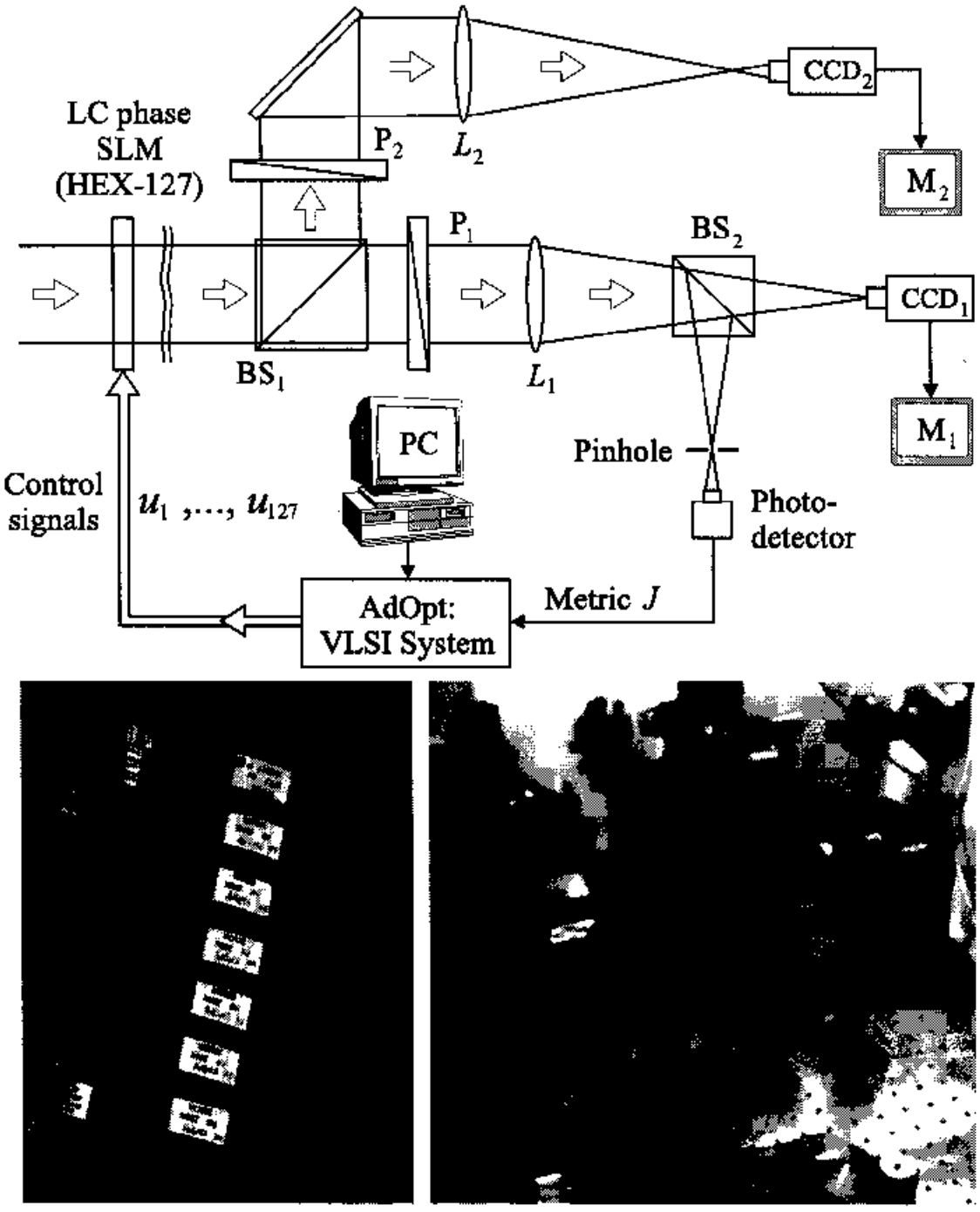

Figure 16.8 Schematic (top) and photograph (bottom right) of the experimental setup for adaptive laser beam focusing using the mixed-mode VLSI system, designated "AdOpt" (bottom left).

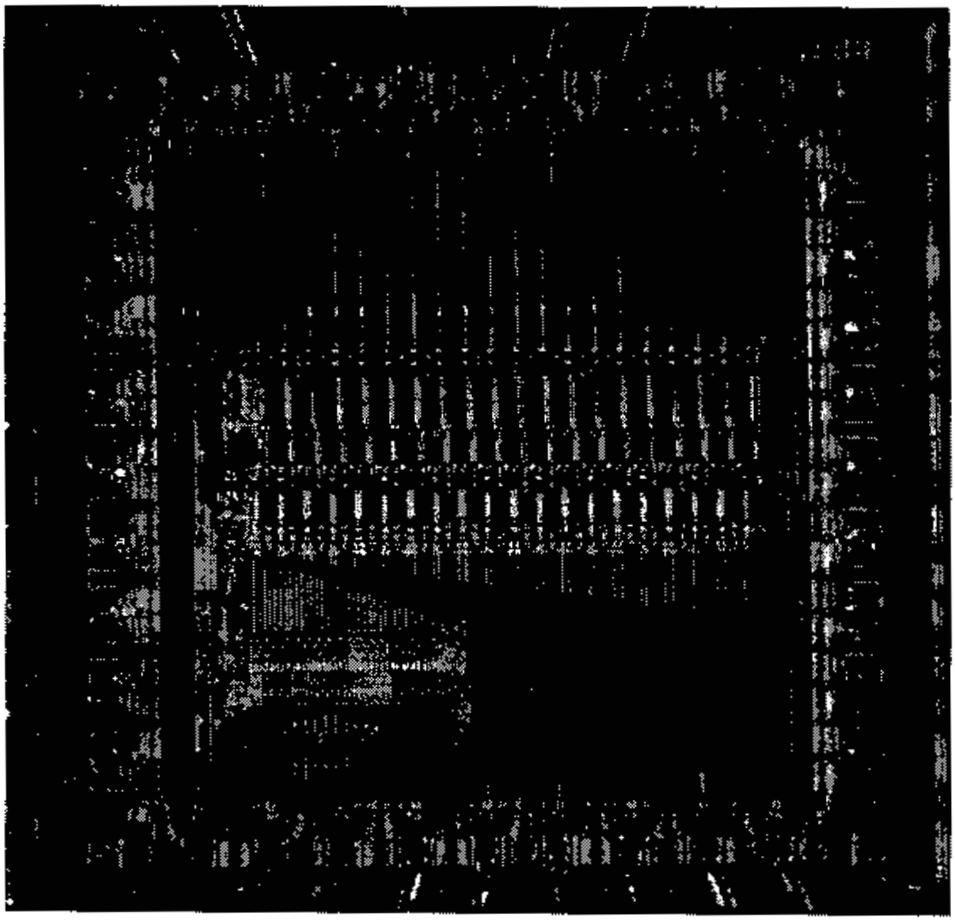

Figure 16.9 Micrograph of the 19 parallel channel mixed-mode VLSI stochastic gradient descent optical controller, a 2.2 × 2.25 sq. mm chip fabricated in 1.2 μm CMOS technology.

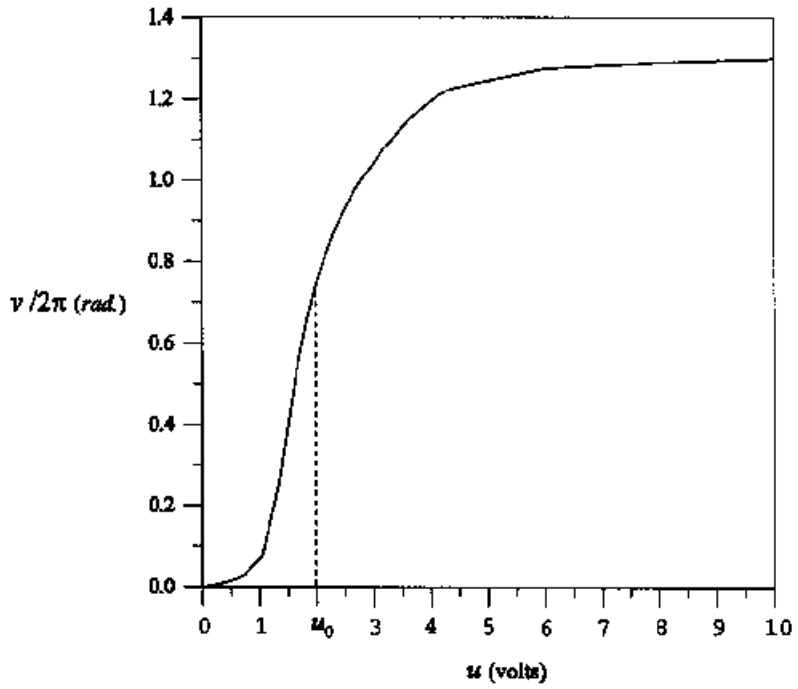

Figure 16.10 Phase-modulation characteristic of the HEX-127 SLM.

The VLSI system has 127 outputs which directly control the cells of the HEX-127 modulator. Each output is an amplitude-modulated 2kHz square wave u^* (the signal required to drive the liquid crystal cells) which is generated by the switched capacitor driver circuit shown in Figure 16.11 such that $u = |u^*|$. The dependence of the amplitude of phase modulation ν produced by a liquid crystal cell in response to the applied periodic waveform u^* with zero-to-peak voltage amplitude u is shown in Figure 16.10. This *modulation characteristic* has a narrow region of operation (1V $< u <$ 5V) in which the phase amplitude changes rapidly through 2π radians. Inside this operational range the modulation characteristic can be approximated by a linear dependence $\nu = \alpha u$.

Note that during the adaptation process individual cells in the SLM could migrate outside the desirable operational range towards the modulation characteristic saturation region, resulting in significant decrease of the performance metric perturbation amplitude ΔJ and hence decreasing the rate of the control algorithm convergence. The effect of saturation can be reduced by accounting for the fact that in most adaptive optics applications the system performance metric J is not sensitive to a constant wavefront shift by an arbitrary mean value ν_0, that is, $J(\nu_1, \ldots, \nu_N) = J(\nu_1 - \nu_0, \ldots, \nu_N - \nu_0)$. To prevent drift of the aperture averaged mean phase value during adaptive operation, the control update algorithm is modified to include an additional penalty term to the metric J accounting for the drift of the mean as follows [11, 12]:

$$u_n^{(m+1)} = u_n^{(m)} - \gamma' \pi_n^{(m)} \Delta J - \eta(\overline{u}^{(m)} - u_0), \tag{16.12}$$

where the control parameter $\overline{u}^{(m)} \equiv \frac{1}{N} \sum_{n=1}^{N} u_n^{(m)}$ corresponds to the aperture averaged phase, and u_0 is a constant voltage. The reference voltage $u_0 = 2V$ corresponds

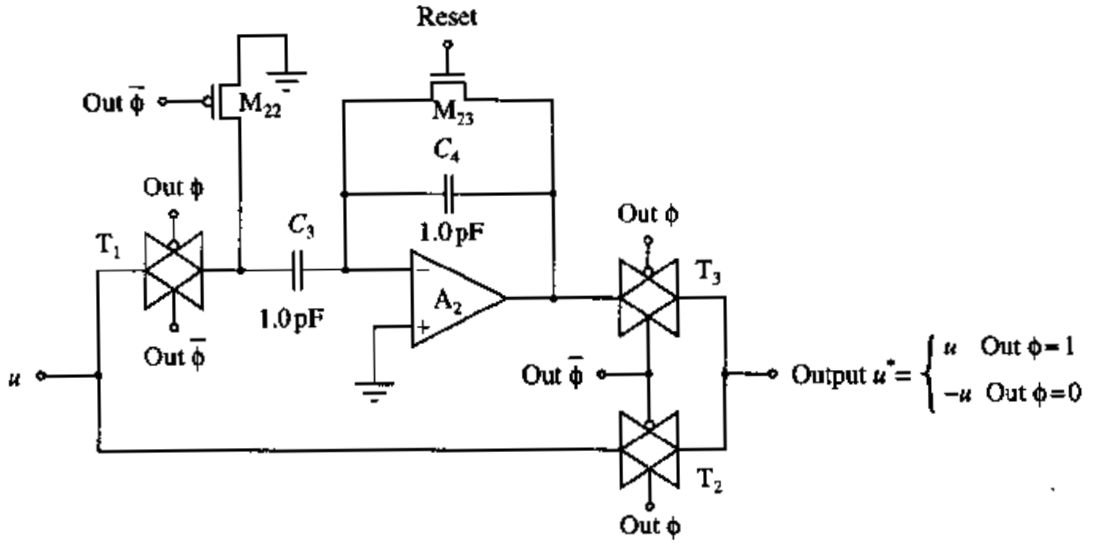

Figure 16.11 This switched capacitor circuit generates the square wave which drives the liquid crystals of the SLM. On alternating cycles, the parameter voltage u and its negative are directed to the output.

to the middle of the liquid crystal phase characteristic in Figure 16.10. The control parameter $\overline{u}^{(m)}$ is calculated on the VLSI system. Typically, $\eta \ll \gamma'$, best implemented by performing the average-value adaptation only once per $\lfloor \gamma'/\eta \rfloor$ cycles of the stochastic perturbative gradient descent adaptation, while using the same values of V_{UPD_p} and V_{UPD_n} and the same unit length "Update" pulse. The "SR_All" signal (Figure 16.5) temporarily overrides the random bit pattern and forces all values π_n to -1, to perturb all parameters unidirectionally. The instantaneous phase bias $\overline{u}$ is estimated by the switched capacitor circuit of Figure 16.12, which is distributed over all channels.

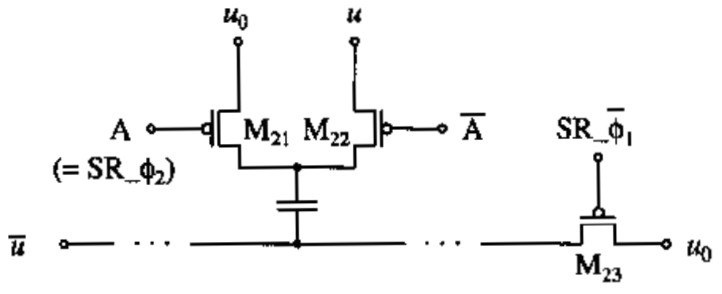

Figure 16.12 Switched capacitor circuit which estimates the mean phase value of the SLM by computing the average of u_n over all control parameters.

16.4.2 Control System Architecture

The VLSI system requires both analog and digital control signals that were generated by a desktop PC with two analog/digital input/output cards (Computer Boards CIO-DAS1602/12 and CIO-DAS08). Using these cards the PC generated the timing signals that drive the VLSI system, performed the criteria measurements and provided control of the algorithm parameters: perturbation amplitude, the sign and the magnitude of the coefficient γ' in (16.12), and the reference voltage u_0. The choice of the VLSI system optimization mode, *i.e.*, maximization/minimization of the performance metric, was implemented by changing the sign of the coefficient γ'. The use of a computer to control parameters of the VLSI system allowed implementation of flexible secondary parameter control for adaptively changing the perturbation amplitude and/or the value of the coefficient γ'.

The computer controller may be replaced by an on-board PIC microcontroller for specific applications requiring less flexibility in control, programmed in non-volatile memory with the adaptation algorithm needed. The use of the microcontroller makes the entire control system compact, fast (within the limits of the microcontroller's A/D conversion), and relatively power efficient. The microcontroller board architecture is shown in Figure 16.13.

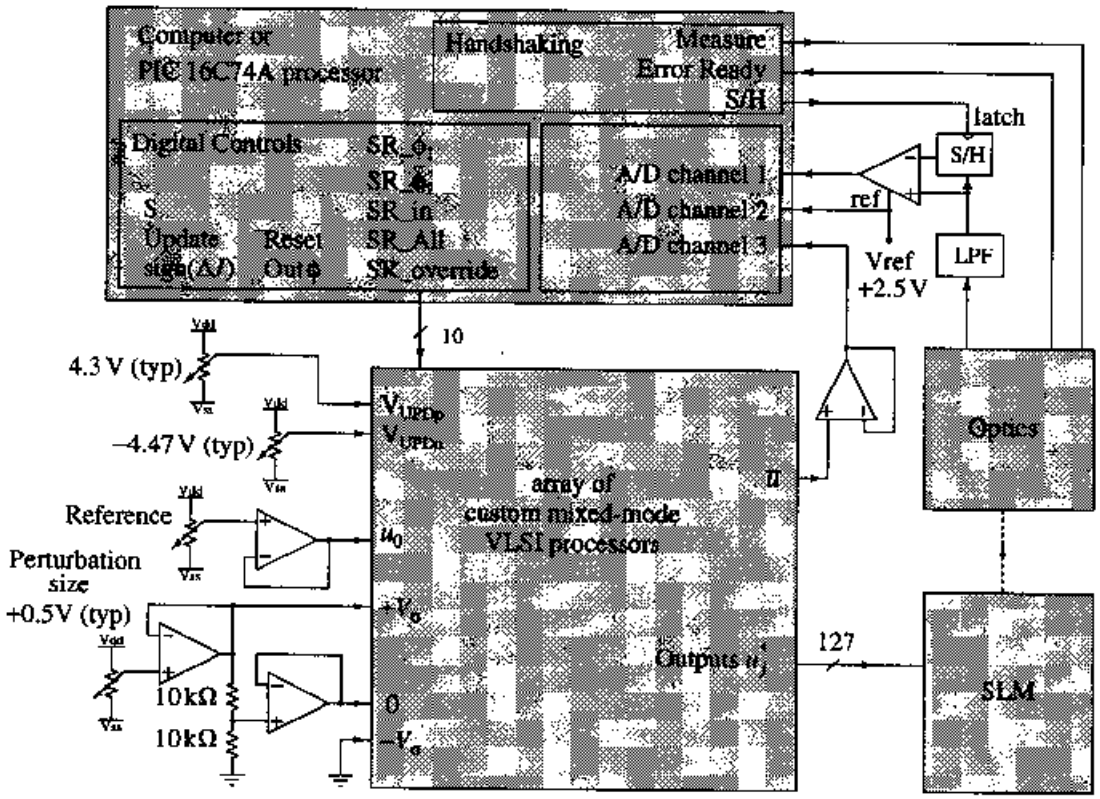

Figure 16.13 Adaptive system, showing supporting electronics hardware in addition to the custom VLSI hardware.

16.5 EXPERIMENTAL RESULTS

16.5.1 Adaptive optical experimental setup

We characterized performance of the VLSI systems using a simple adaptive laser focusing system shown in Figure 16.8. The beam from an Argon laser (λ = 514nm) was expanded to a diameter of 30mm and then passed through the HEX-127 phase modulator. The optical axis of the SLM was set at an angle $\pi/4$ with respect to the direction of the input beam polarization. A beamsplitter BS_1 equally divided the input wave between two optical channels. Two polarizers P_1 and P_2 were placed behind the beamsplitter. To separate the phase-only modulated component of the input wave the optical axis of the polarizer P_1 was set parallel to the liquid crystal optical axis. This phase modulated wave passed through the lens L_1 and the beamsplitter BS_2. A small pinhole of 25μm diameter was placed in the focal plane of the lens L_1 (with a focal length F_1 = 14in.) A photodetector placed behind the pinhole measured the laser beam power inside the pinhole. The VLSI system used the photodetector voltage output (filtered with a simple antialiasing lowpass prior to sampling) as the performance metric J. A camera CCD_1 registered the intensity distribution of the laser beam in the focal plane of the lens. The video image was displayed on a monitor.

A second optical channel containing a polarizer P_2, lens L_2, and camera CCD_2 was used to image the HEX-127 spatial light modulator and make visible the wavefront change occurring during the adaptation process. The optical axis of the polarizer P_2 was set orthogonal with respect to the direction of the input beam polarization and at an angle $\pi/4$ with the liquid crystal optical axis. This allowed visualization of the liquid crystal cell phase modulation as an intensity pattern shown in Figure 16.14(a,c). The lens L_2 imaged the intensity pattern of the HEX-127 SLM onto the CCD_2 camera, displayed on a second monitor.

The results of adaptive system performance are presented in Figs. 16.14 and 16.15. The adaptive system was exercised with a repeating sequence of performance metric maximization and minimization. During the first half of the cycle, or 512 steps, the system performed beam quality metric maximization ($\gamma' > 0$), followed by 512 iterations of beam quality metric minimization. During the minimization stage the adaptive system created random phase distortions resulting in the laser beam focal plane intensity spreading out. The characteristic pictures of the HEX-127 intensity patterns, and the focal plane intensity distributions for both adaptation regimes are shown in Figure 16.14.

The dependencies $J(m), m = 1, \ldots, 1024$ (adaptation evolution curves) were averaged over 100 adaptation cycles. The normalized averaged evolution curve (metric values $\langle J(m) \rangle$) is shown in Figure 16.15 (a). The normalized standard deviation of the system performance metric $\sigma_J(m)$ and the standard deviation of the metric perturbation $\sigma_{\Delta J}(m)$ are shown in Figure 16.15 (b). The normalized standard deviations were calculated using the following expressions:

$$\sigma_J(m) = \frac{\left\langle (J(m) - \langle J \rangle)^2 \right\rangle^{\frac{1}{2}}}{\langle J \rangle} \tag{16.13}$$

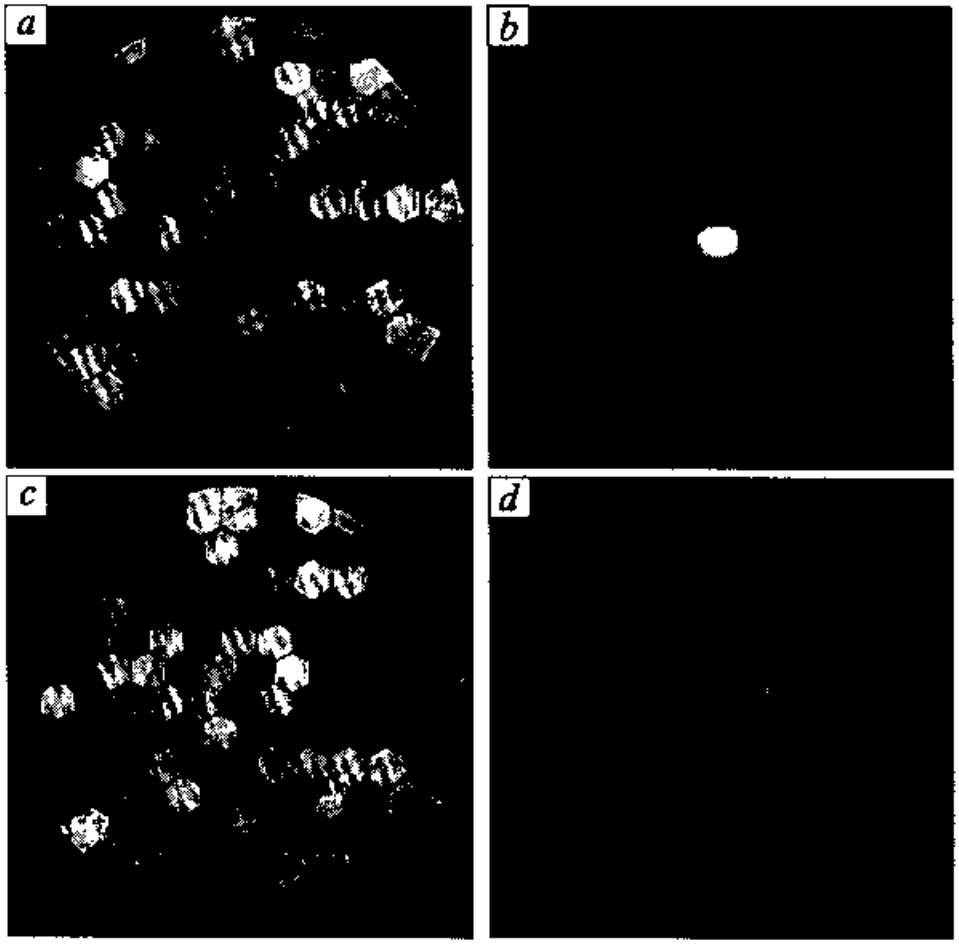

Figure 16.14 Experimental results of wavefront control in the adaptive laser beam focusing system with the VLSI controller, for performance metric maximization (a),(b) and minimization (c),(d). The intensity modulation patterns for HEX-127 (a) and the focal plane intensity distribution (b) correspond to the iteration step $m = 500$. The patterns (c) and (d) correspond to $m = 1000$.

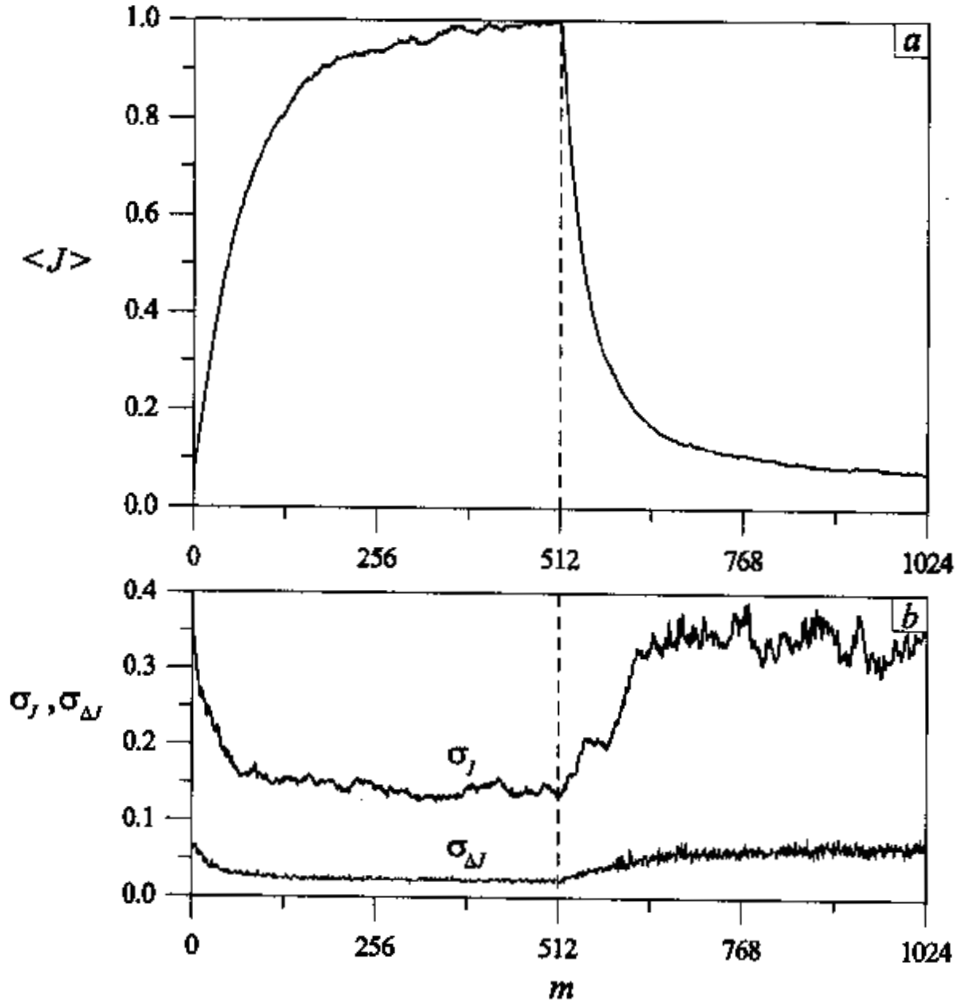

Figure 16.15 Averaged laser beam quality metric (a) and standard deviations for values J and ΔJ (b) during the sequence of system performance metric J maximization ($m < 512$) and minimization ($m > 512$).

and

$$\sigma_{\Delta J}(m) = \frac{\langle (\Delta J(m) - \langle \Delta J \rangle)^2 \rangle^{\frac{1}{2}}}{\langle J \rangle} . \tag{16.14}$$

The evolution curves in Figure 16.15 a show the existence of two characteristic phases of the adaptation process: a relatively rapid convergence during first 100 to 150 iterations followed by a decrease in the convergence rate, a behavior also observed in numerical simulations [12]. The convergence occurs approximately 1.5 times faster for the metric minimization than for metric maximization (Figure 16.15 (*a*)). This confirms a natural expectation that it is easier to create a phase distortion than compensate one. The adaptation behavior reflects the fact that the number of system states corresponding to a highly distorted beam (metric local minimum) is greater then the number of states corresponding to the local maximum. There are more ways to destroy quality of the beam then correct it. The presence of noise (laser beam intensity and photocurrent fluctuations) has more impact on the minimization phase of the adaptation process because the laser beam intensity is low. Both factors—higher noise level and larger number of local extrema of the performance metric—result in a significantly higher level of the performance metric fluctuations ($\sigma_J(m)$ in Figure 16.15 (*b*)) for the minimization than for the maximization process. The value of the standard deviation for the performance metric perturbation ($\sigma_{\Delta J}(m)$) was approximately 3% of the metric averaged value for the maximization and about 7% for the minimization process.

16.6 CONCLUSIONS

We have designed, built and tested a mixed-mode VLSI system for adaptive wavefront correction using parallel stochastic perturbative gradient descent. Although we have demonstrated its operation for a phase SLM with 127 elements, our design allows for expansion by addition of VLSI modules. The speed of operation of our system is presently limited by the dynamic response of the SLM (in the millisecond range). Our continued research in this area investigates alternative SLM technologies including high-speed and large-scale MEMS, coupled with CMOS to integrate control parameters along with the adaptive processing elements.

This is the first demonstration of a fully-integrated system for adaptive phase wavefront correction using parallel stochastic optimization. Since the implemented optimization is model-free and the objective measure can be arbitrarily specified, the results carry over to a large class of adaptive optics applications such as adaptive imaging through a turbulent atmosphere, or adaptive focusing for medical applications. The parallel architecture is extendable to higher resolutions ($N \sim 10^3$ to 10^6 parameters) which is one of the most challenging problems in adaptive optics.

References

[1] J.W. Hardy, "Active optics: a new technology for the control of light," *Proc. IEEE*, **66**, pp. 651–697, 1978.

[2] V.P. Linnik, "The possibility for reducing the influence of atmospheric turbulence on a star image," *Sov. Opt. Spectrosc.*, **4**, p. 401, 1957.

[3] H.W. Babcock, "Deformable optical elements with feedback," *J. Opt. Soc. Am.*, **48**, p. 500, 1958.

[4] R.K. Tyson, *Principles of Adaptive Optics*, Academic Press, Boston, 1991.

[5] M. A. Vorontsov and V. I. Shmalhauzen, *Principles of Adaptive Optics*, Nauka, Moscow, 1985.

[6] B.Y. Zeldovich, N.V. Pilipetsky, and V.V. Shkunov, *Principles of Phase Conjugation*, No. 42, Springer Series in Optical Sciences, Berlin Springer-Verlag, 1985.

[7] A. Dembo and T. Kailath, "Model-Free Distributed Learning," *IEEE Transactions on Neural Networks*, vol. **1** (1), pp 58-70, 1990.

[8] M. Avriel, *Nonlinear Programming: Analysis and Methods*, Prentice-Hall, Englewood Cliffs, N.J., 1976.

[9] M.A. Vorontsov, G.W. Carhart, D.V. Pruidze, J.C. Ricklin and D.G. Voelz, "Adaptive Imaging System for Phase-Distorted Extended Source/Multiple Distance Objects," *Appl. Optics*, **36**, pp. 3319–3328, 1997.

[10] T.R. O'Meara, "The multi-dither principle in adaptive optics," J. Opt. Soc. Am. 67, 315-325, 1977.

[11] M.A. Vorontsov, G.W. Carhart and J.C. Ricklin, "Adaptive phase-distortion correction based on parallel gradient-descent optimization", Opt. Lett., 22, 907-909, 1997.

[12] M. A. Vorontsov and V. P. Sivokon, "Stochastic parallel gradient descent technique for high-resolution wavefront phase distortion correction," *J. Opt. Soc. Am.* A, **15**, pp. 2745–2758, 1998.

[13] J. C. Shelton and S. L. Baliunas, "Results of Adaptive Optics at Mount Wilson Observatory," *Smart Structures and Materials 1993: Active and Adaptive Optical Components and Systems II* M. A. Ealey, ed., Proc. SPIE, **1920** (371), 1993.

[14] R. G. Dekany, "The Palomar Adaptive Optics System," *Adaptive Optics*, OSA 1996 Technical Digest Series **13** (40), 1996.

[15] M. A. Vorontsov, G. W. Carhart, D. V. Pruidze, J. C. Ricklin, and D. G. Voelz, "Image quality criteria for an adaptive imaging system based on statistical analysis of the speckle field," *J. Opt. Soc. Am.* A **13**, pp. 1456–1466, 1996.

[16] H.J. Kushner, and D.S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, New York, NY: Springer-Verlag, 1978.

[17] H. Robins and S. Monro, "A Stochastic Approximation Method," *Annals of Mathematical Statistics*, vol. **22**, pp 400-407, 1951.

[18] J.C. Spall, "A Stochastic Approximation Technique for Generating Maximum Likelihood Parameter Estimates," Proceedings of the 1987 American Control Conference, Minneapolis, MN, 1987.

[19] M.A. Styblinski and T.-S. Tang, "Experiments in Nonconvex Optimization: Stochastic Approximation with Function Smoothing and Simulated Annealing," *Neural Networks,* vol. **3** (4), pp 467-483, 1990.

[20] G. Cauwenberghs, "A Fast Stochastic Error-Descent Algorithm for Supervised Learning and Optimization," in *Advances in Neural Information Processing Systems,* San Mateo, CA: Morgan Kaufman, vol. **5**, pp 244-251, 1993.

[21] J. Alspector, R. Meir, B. Yuhas, and A. Jayakumar, "A Parallel Gradient Descent Method for Learning in Analog VLSI Neural Networks," in *Advances in Neural Information Processing Systems,* San Mateo, CA: Morgan Kaufman, vol. **5**, pp 836-844, 1993.

[22] B. Flower and M. Jabri, "Summed Weight Neuron Perturbation: An $\mathcal{O}(n)$ Improvement over Weight Perturbation," in *Advances in Neural Information Processing Systems,* San Mateo, CA: Morgan Kaufman, vol. **5**, pp 212-219, 1993.

[23] M. Jabri and B. Flower, "Weight Perturbation: An Optimal Architecture and Learning Technique for Analog VLSI Feedforward and Recurrent Multilayered Networks," *IEEE Transactions on Neural Networks,* vol. **3** (1), pp 154-157, 1992.

[24] D. Kirk, D. Kerns, K. Fleischer, and A. Barr, "Analog VLSI Implementation of Gradient Descent," in *Advances in Neural Information Processing Systems,* San Mateo, CA: Morgan Kaufman, vol. **5**, pp 789-796, 1993.

[25] G. Cauwenberghs, "A Learning Analog Neural Network Chip with Continuous-Recurrent Dynamics", in *Advances in Neural Information Processing Systems,* San Mateo, CA: Morgan Kaufman, vol. **6**, pp 858-865, 1994.

[26] G. Cauwenberghs, "An Analog VLSI Recurrent Neural Network Learning a Continuous-Time Trajectory," *IEEE Transactions on Neural Networks,* vol. **7** (2), March 1996.

[27] G. Cauwenberghs, and A. Yariv, "Fault-Tolerant Dynamic Multi-Level Storage in Analog VLSI," *IEEE Transactions on Circuits and Systems II,* vol. **41** (12), pp 827-829, 1994.

[28] G. Cauwenberghs, "Analog VLSI Stochastic Perturbative Learning Architectures," *J. Analog Integrated Circuits and Signal Processing* **13** (1–2) pp. 195–209, 1997.

[29] J. Alspector, J.W. Gannett, S. Haber, M.B. Parker and R. Chu, "A VLSI-Efficient Technique for Generating Multiple Uncorrelated Noise Sources and Its Application to Stochastic Neural Networks," *IEEE T. Circ. Syst.*, vol. **38** (1), pp 109-123, 1991.