

# High accuracy computation with linear analog optical systems: a critical study

Demetri Psaltis and Ravindra A. Athale

High accuracy optical processors based on the algorithm of digital multiplication by analog convolution (DMAC) are studied for ultimate performance limitations. Variations of optical processors that perform high accuracy vector-vector inner products are studied in abstract and with specific examples. It is concluded that the use of linear analog optical processors in performing digital computations with DMAC leads to impractical requirements for the accuracy of analog optical systems and the complexity of post-processing electronics.

## I. Introduction

Analog optical processors can be used in a variety of ways to implement linear transformations and filtering useful in signal processing problems with very high throughput requirements. Among the unique features of optics exploited in such processors are (1) 1-D or 2-D parallelism, (2) ease of performing complex multiplication and addition, and (3) global and arbitrary communication between the parallel channels of computation. The most notable successes of analog optical processing have been synthetic aperture radar processing,<sup>1</sup> acoustooptic processors for rf spectrum analysis,<sup>2</sup> and correlations.<sup>3</sup> One of the most important performance parameters of these systems is the linear dynamic range defined as the ratio of the highest allowable input signal level where nonlinear distortions appear to the lowest input signal level that produces an output signal (e.g., the correlation peak or the spectrum) equal to the random noise at the output due to detector dark current, scattered light, etc. The use of heterodyning techniques in output detection has led to optical systems with 70 dB of linear dynamic range in rf signal power.<sup>4</sup>

A traditional approach to improving computational accuracy of any system is to represent measurable quantities in a digital number system in which a single large number is represented by an ordered  $n$ -tuple of

small numbers [ $a \rightarrow (a_1, a_2, \dots, a_n)$ ]. The encoding and decoding can be a complicated process determined by the particular number system used. The most popular digital representations are the binary and decimal numbers systems, even though other systems such as residue arithmetic have been used. Since in a digital number system several small numbers are used to represent a large number, the dynamic range limitations of the analog system can be overcome, and it becomes possible to represent large numbers accurately. The operations of multiplication and addition in the digital number system also break down into several small size calculations. The individual subcalculations, however, involve nonmonotonic nonlinearities and, in the cases of binary or decimal systems, involve interaction between adjacent digits in the form of carry propagation. Thus, in terms of optical processing, the use of digital number representation involves more than a simple trade-off between accuracy and parallelism (and, therefore, speed). It introduces nonmonotonic nonlinear operations, which are not easily implemented with optics.

During the past decade several research projects have been undertaken to develop nonlinear optical components and systems to perform computation. Optical logic is a growing field in which devices with improved performance are being developed continuously. The application of these logic devices to numerical computation in binary number system, however, has seen only limited development. The residue number system has also been investigated for optical realization using integrated optical switches, diffraction gratings, and liquid crystal light valves.<sup>5-8</sup> More recent developments have used holographic table look-up processors with binary as well as residue number systems<sup>9</sup> and have proposed use of threshold logic gates in implementing the truth tables for binary multiplication and addition.<sup>10</sup>

Demetri Psaltis is with California Institute of Technology, Pasadena, California 91125, and R. A. Athale is with BDM Corporation, 7915 Jones Branch Road Drive, McLean, Virginia 22102.

Received 11 March 1986.

0003-6935/86/183071-07\$02.00/0.

© 1986 Optical Society of America.

When and if numerical optical processors that utilize nonlinear optical devices will be practically useful in the near future is a very important question, which is not addressed in this paper. Instead we will investigate another approach in which linear analog optical systems are used to perform partial calculations and produce an intermediate result that can be converted to standard digital representation with appropriate nonmonotonic nonlinear operations. This approach has a long history, and the first reference to these techniques is found in the ancient Indian scriptures of the Vedas (~2000 years old) as a shortcut to large number multiplications.<sup>11</sup> In more recent times, Schwartzlander suggested it to the electronics community as a "quasi serial multiplier,"<sup>12</sup> and Whitehouse and Speiser<sup>13</sup> introduced it to the optical signal processing community under the name of "digital multiplication by analog convolution" or DMAC for short.<sup>13</sup> The first optical implementation was carried out by Psaltis *et al.*,<sup>14</sup> and schemes to incorporate it in optical linear algebra processors were forwarded by Guilfoyle<sup>15</sup> and Collins *et al.*<sup>16</sup> In the last four years numerous modifications to the basic idea have been suggested with different predicted performances.<sup>17-22</sup> In this paper we will examine in detail the trade-offs involved in performing the digital calculations by linear analog optical systems. Section II contains a general trade-off analysis for a generic system. Section III contains several specific examples of high accuracy optical processors with a common technology substrate and a common throughput performance to facilitate the extraction of basic limitations that cannot be changed by architectural ingenuity. Section IV details the conclusions of this study and makes some recommendations.

## II. General Trade-Offs

We will consider the trade-offs associated with a generic system designed to perform linear operations on binary formatted data. The binary encoding of data has three immediate consequences. The first is an increase in the number of elementary operations that need to be performed compared to an analog implementation. Each sample is represented by several bits, and we normally need to operate on each bit several times to perform the desired calculation. Thus, given the same system resources (space-bandwidth product, temporal bandwidth, and dynamic range), we end up performing fewer multiplications and additions per second with the binary encoded data and compared to the analog encoding. This loss in processing speed is generally acceptable if it can be traded off for improved accuracy. The work in applying the digital multiplication by analog convolution (DMAC) algorithm to optical linear algebra processors was motivated by exactly this hope.

Unfortunately, the decrease in computational throughput is not the only consequence of the binary encoding of data. Even though the requirement of performing nonmonotonic nonlinear operations and propagating carries can be postponed by using the

DMAC algorithm, it cannot be avoided entirely. This task falls to the electronic postprocessor that has to handle the signals coming out of the optical detector array and convert them into standard binary number. A trade-off analysis must incorporate the performance requirements and complexity of the postprocessing electronics for proper evaluation of the possible comparative advantages of the optical processor.

A third factor to be considered is the performance required of the linear analog optical processor. Although the input data are encoded in binary, the absence of quantizing and standardizing procedures within the optical system implies that the optical signals at the output will have multiple levels. The number of levels that need to be resolved with low probability of error will be governed by the number bits used in the binary representation and the number of parallel multiplications carried out by the optical processor. Since these two quantities also determine the throughput and accuracy of the processor, the dynamic range and accuracy of the analog optical processor play a crucial role in determining the ultimate performance.

### A. Description of a Generic System

We will first examine the DMAC algorithm in detail. Let  $f$  and  $g$  be two integers and  $f_i$  and  $g_i$  be the binary strings that represent the two numbers in a binary number system. Therefore,

$$f = \sum_{i=0}^{L-1} f_i 2^i; \quad g = \sum_{i=0}^{L-1} g_i 2^i. \quad (1)$$

If the product of the two numbers is denoted by  $h$ ,

$$h = f \cdot g = \sum_{i=0}^{(2L-2)} 2^i \sum_{j=0}^{(L-1)} (f_j g_{i-j}). \quad (2)$$

This simple relationship provides us with the DMAC algorithm for multiplying two binary numbers using a linear optical system. The two binary strings are convolved with each other, and the final answer is obtained by weighting each term of the convolution with an exponential factor ( $2^i$ ) and summing over all terms. We normally perform the convolution using an analog optical system to obtain the coefficients of expansion and skip direct evaluation of the summation over  $i$  in Eq. (2) to avoid the explosion in dynamic range that results from the exponential weights. We are thus evaluating the nonbinary string  $h_i$  which represents the integer  $h$ .

$$h_i = \sum_{j=0}^{L-1} f_j g_{i-j}. \quad (3)$$

The analog string can be converted into a standard binary string by using an analog-to-digital (A-D) converter and then performing binary addition with appropriate shifts of the binary strings corresponding to  $h_i$ . The block diagram of the system performing this operation is shown in Fig. 1.

This idea can be generalized to allow implementation of any linear transformation where the basic arithmetic operation involved is a sum of products. The

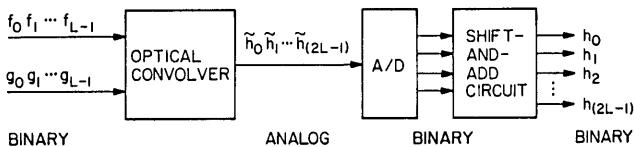


Fig. 1. Block diagram of the DMAC multiples.

sum of two binary encoded numbers can be obtained by simply adding without carries the equivalent bits pairwise with an analog processor and then performing the operation of A-D conversion and binary addition with shifts that we described in connection with the multiplication. Now we can merge these techniques of digital multiplication and digital addition by linear analog optical systems combined with a digital electronic postprocessor and perform the canonical operation of an inner product between two  $n$ -element column vectors  $\mathbf{f}$  and  $\mathbf{g}$ . The elements of the vector are represented by binary strings. Therefore,  $f_{i,k}$  and  $g_{i,k}$  correspond to the  $i$ th bits of the  $k$ th elements of vectors  $\mathbf{f}$  and  $\mathbf{g}$ , respectively. The inner product between  $\mathbf{f}$  and  $\mathbf{g}$  is a scalar  $h$  given by

$$h = \mathbf{f}^T \mathbf{g} = \sum_{k=1}^N f_{k} g_{k} = \sum_{k=1}^N \sum_{i=0}^{(2L-2)} 2^i \sum_{j=0}^{L-1} f_{j,k} g_{(i-j),k}. \quad (4)$$

This equation can be rewritten by rearranging the order in which the three summations are performed:

$$h = \sum_{i=0}^{2L-2} 2^i \left\{ \sum_{k=1}^N \sum_{j=0}^{L-1} f_{j,k} g_{(i-j),k} \right\}. \quad (5)$$

The term in the bracket is 2-D linear operation that is implemented with an analog optical processor, and the result is then digitized using the sequence of operations described earlier to produce the binary string representing  $h$  while avoiding the explicit evaluation of the exponential weights. Again the decomposition suggested here is not unique. One could choose to add fewer bits optically by digitizing after only one of the summations or partial sums is performed. As we will see shortly, the number of bits that are accumulated to form each output sample is a crucial parameter that directly affects the accuracy of the processor.

A schematic diagram of a generalized optical processor that performs linear operations using the DMAC algorithm is shown in Fig. 2. The system has  $N_1$  parallel spatial channels at the input, and each channel accepts binary bits at a rate  $B_1$ . Each channel may accept bits every clock cycle from a separate external information source or from the adjacent channel. The information in each channel is multiplied by  $M$  separate bits in the optical processor. We call  $M$  the fan-out factor since it is generally equal to the number of output channels that are illuminated by light emitted from each input channel. The system has  $N_2$  parallel output channels each having temporal bandwidth  $B_2$ . The binary products that are optically formed are

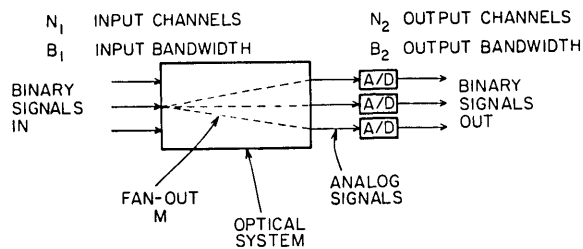


Fig. 2. Schematic diagram of a generalized optical processor.

accumulated (added) at the output plane through either spatial or temporal integration to form a linear transformation of the binary input data. If  $N_1 M > N_2$ , the system performs spatial integration since multiple bit products are detected at the same spatial location. If  $B_1 > B_2$ , the system is time integrating. Both conditions can hold simultaneously, in which case the linear transformation is performed through a combination of temporal and spatial integrations. The signal detected at each output channel is electronically converted to the binary representation by the A-D converter and a shift-and-add circuit.

Having defined these parameters we can characterize the performance of any specific architecture without further knowledge of the details of the implementation. Thus we will be able to derive some guidelines that are generally applicable. The number of bit multiplications that the processor performs per unit time is equal to  $N_1 B_1 M$ . The number of bit multiplications that are required to realize one multiplication between two integers with DMAC is  $aL^2$ , where  $a$  is a constant between 1 and 4, depending on the efficiency of the specific implementation, and  $L$  is the number of bits that are used to represent each number. The processing power  $P$  of the overall system is

$$P = N_1 B_1 M / aL^2 \text{ multiplications/s.} \quad (3)$$

Clearly,  $P$  is a number we wish to maximize. Also note that Eq. (3) demonstrates the trade-off between accuracy and processing speed referred to earlier. The number of analog-to-digital conversions that need to be performed per unit time is

$$C = N_2 B_2 \text{ A-D conversions/s.} \quad (4)$$

Again it is clear that  $C$  is a number we wish to minimize to keep the complexity of the electronics at a minimum. Unfortunately,  $P$  and  $C$  cannot be independently chosen. To appreciate this fact, we define the ratio

$$R = P/C = \left( \frac{N_1 B_1 M}{N_2 B_2 aL^2} \right) \text{ multiplications/A-D conversion.}$$

The ratio  $R$  increases monotonically as either  $P$  increases or  $C$  decreases. Therefore, we want to make  $R$  as large as possible.  $R$  provides a direct comparative estimate of the optical implementation vs an all electronic implementation. If, for example,  $R$  were equal to one, only one binary multiplication is being performed by the optical system per A-D conversion. Since it is about equally difficult to perform multipli-

Table I. Design Parameter Common to all Architectures

Vector/matrix dimensions	$N = 32$
Input accuracy	$M = 16$ bits
Input accuracy (per element)	$B = 50$ Mbits/s
Overall throughput	$C = 50$ MOPS
Output accuracy	21–22 bits

cations and A–D conversions electronically, this would be a strong indication that optics offers no advantage in this case. To determine the maximum value for  $R$  we consider the characteristics of the output stage of the processor. The number of bits that are being generated per unit time within the optical processor is equal to  $(N_1 B_1 M)$ . The number of samples that are being transferred out of the optical processor per unit time is  $N_2 B_2 (= C)$ . The ratio  $(N_1 B_1 M)/(N_2 B_2)$  is, therefore, equal to the maximum number of bits that are accumulated (through either spatial or temporal integration) to form each output sample. Therefore, this ratio cannot exceed the accuracy of the system  $DR_2$ , which is defined as the number of distinguishable signal levels that can be produced reliably by the detector and A–D converter. This gives us a very simple upper bound for  $R$ :

$$R < DR_2 / aL^2. \tag{6}$$

For example, if  $L = 10$  and  $a = 1$ , we require the number of distinguishable levels at the output to be at least 100 for  $R$  to be equal to 1. It is important to note that the output levels must be sufficiently well defined so that the A–D converter can detect all of them with very low probability of error.  $DR_2$  is much smaller than what is conventionally called the detector dynamic range. We estimate that it would require very sophisticated engineering to obtain  $DR_2 = 100$ , and it does not appear that  $DR_2 = 1000$  is practically feasible in the foreseeable future.

The pessimistic result from the above discussion regarding the viability of DMAC-based optical algebraic processors is based on the assertion that an analog-to-digital conversion is approximately equally costly as an electronically implemented binary multiplication. We can accept this without further qualification if the two operations are performed at the same speed and with the same number of bits. Through some specific examples we will consider in the following section, we will see that this is not always the case. It is appropriate then to ask whether we can derive a possible advantage by replacing fast digital multipliers with a larger number of slower ADCs that perhaps also have a smaller word length. The answer to this question is a qualified no. The reason is that we can generally find a digital implementation that also uses more but slower multipliers that can solve the same problem at the same overall processing rate. The qualification is that in problems where the input binary data are available at a very high data rate, a time-integrating optical processor can accept the input and deliver it at slower rates to the ADCs. In this rather specialized situation optics may help avoid the need

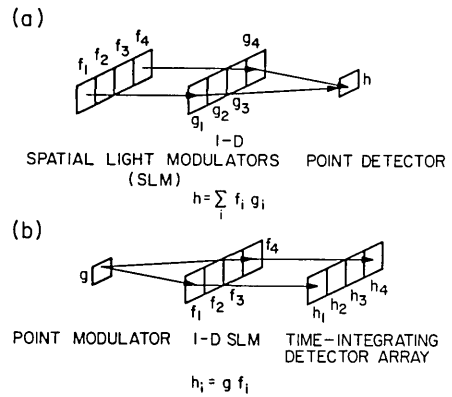


Fig. 3. Schematic diagrams of optical systems for performing a vector–vector inner product (a) and scalar–vector product (b). A vector–matrix multiplication is performed by repeatedly performing either operation.

for high speed multipliers. In general, however, there is not a clear advantage in DMAC-based optical processors over electronic systolic arrays, and thus we do not expect these systems to have a broad significant impact in signal processing.

### III. Specific Examples

In this section we elaborate on the conclusions of the previous section through specific examples. The particular linear operation chosen is a vector–matrix multiplication, since a large number of more complex operations can be decomposed in terms of this operation and also since it is useful by itself. Acoustooptic Bragg cells are selected as the active optical devices, since they represent the most mature technology and since most of the previous work in this area was also based on this technology. The architectures to be considered here handle all the elements of an input vector in parallel and hence can be thought of as 1-D processors. However, since each element is represented by several bits, the optical system has to use the second spatial dimension as well, thus physically giving a system that is 2-D in nature. Table I shows the performance parameters common to all the architectures to be analyzed here. These values were chosen in view of the current state-of-the-art in device technology. The same computational throughput will be shown to require different levels of system performance for different architectures.

The operation of vector–matrix multiplication can be implemented on a 1-D processor using two different strategies: (1) space-integrating architecture based on a vector–vector inner product; (2) time-integrating architecture based on a scalar–vector product. The schematic diagrams of these two systems are shown in Fig. 3. In the space-integrating architecture, the vector  $\mathbf{g}$  and rows of the matrix  $\mathbf{F}$  [ $f^{(i)}$  is the  $i$ th row of  $\mathbf{F}$ ] are input in parallel, while the output vector  $\mathbf{h}$  is calculated sequentially one element at a time. In the time-integrating architecture, on the other hand, the vector  $\mathbf{g}$  is entered sequentially, and columns of  $\mathbf{F}$  [ $f^{(j)}$  is the  $j$ th column of  $\mathbf{F}$ ] are input in parallel, while the output

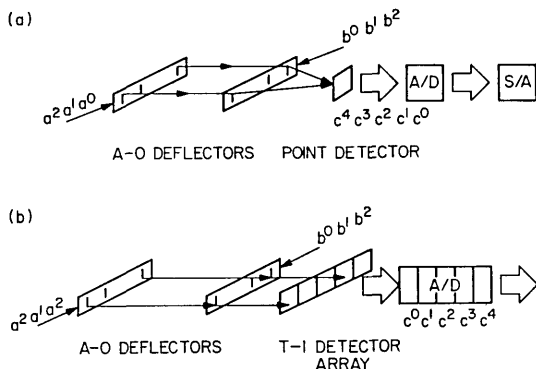


Fig. 4. Two ways of performing digital multiplication via linear analog optical systems: (a) space-integrating convolver; (b) time-integrating convolver.

vector  $\mathbf{h}$  is accumulated in parallel in the time-integrating detector array.

The convolution operation needed to implement the digital multiplication can similarly be performed using space integration or time integration. The schematic diagrams of these two systems are shown in Fig. 4. It should be noted that one of the binary sequences has to be reversed with respect to the other in the time-integrating convolver.

The two choices each for the matrix operation and the digital multiplication can be combined to produce four architectures for high accuracy vector-matrix multiplication. These are

- (1) inner product/space-integrating convolution;
- (2) scalar-vector product/time-integrating convolution;
- (3) inner product/time-integrating convolution;
- (4) scalar-vector product/space-integrating convolution.

Figures 5-8 depict these four processors. The multitransducer acousto-optic Bragg cells shown in the figures are assumed to have thirty-two parallel channels (the size of the vector) with a time-bandwidth product of  $(2L - 1)$ , where  $L$  is the number of bits (sixteen in this case). The requirements for the active devices in these four systems are widely different. Although all the active devices need to be considered for a complete system design, we will concentrate on the complexity of the postprocessing electronics and the analog accuracy of the optical processor since we have already established that this part of the processor is the most crucial in determining the practicality of the system. Table II contains a list of the relevant parameters for all the processor architectures. In what follows, we discuss each system briefly and explain the origin of the parameter values obtained in each case.

(1) Inner product/space-integrating convolution: In this architecture (Fig. 5) space integration is used for the summation of the vector elements and for the convolution. The output vector  $\mathbf{h}$  is thus produced in a bit as well as element sequentially fashion. Since each analog bit of each element of  $\mathbf{h}$  involves summa-

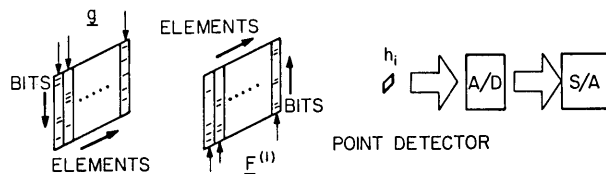


Fig. 5. Schematic diagram of a system employing a vector-vector inner product with space-integrating convolution.

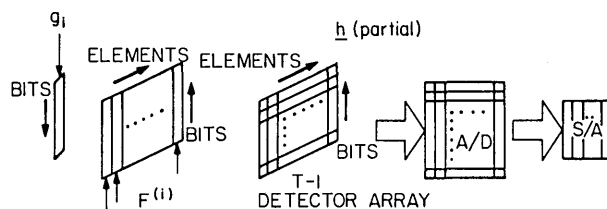


Fig. 6. Schematic diagram of a system employing a scalar-vector product with time-integrating convolution.

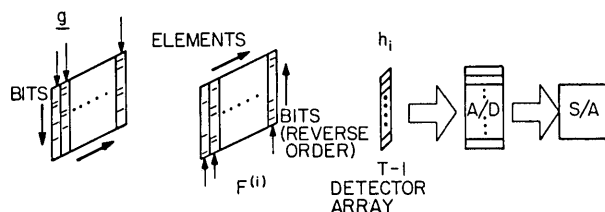


Fig. 7. Schematic diagram of a system employing a vector-vector inner product with time-integrating convolution.

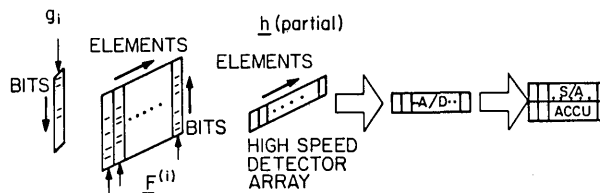


Fig. 8. Schematic diagram of a system employing a scalar-vector product with space-integrating convolution.

Table II. Parameters for Processor Architectures

Architectures	I	II	III	IV
No. of detectors/A-D	1	1024	31	32
Bandwidth per channel	50 MHz	50 kHz	1.5 MHz	50 MHz
Accuracy per channel	9 bits	9 bits	9 bits	4 bits
Additional postprocessing	Shift-add (32 bits)	Shift-add (32 bits)	Shift-add (32 bits)	Shift-add + accumulator (32 bits)

tion over sixteen bits of the input vector element as well as summation of thirty-two elements of the input vector, the single detector and ADC will be required to resolve 512 levels (or nine bits). A new bit is calculated at the output for every clock cycle of the input device,

and hence the detector and ADC bandwidth will be 50 MHz.

(2) Scalar-vector product/time-integrating convolution: In this architecture (Fig. 6) time integration is used for the vector-element summation and the convolution. The output vector  $\mathbf{h}$  is thus produced in a bit- and element-parallel fashion at the end of the integration period. The number of levels to be resolved by the individual detector element and the ADC associated with it still remains at 512, since that is totally determined by the size of the problem and is independent of the method (space or time integration) used to produce the final answer. Since all the bits of all the elements of the vector  $\mathbf{h}$  are accumulated in parallel a  $32 \times 31$  2-D time-integrating detector array is needed in the output with an ADC associated with each detector element. The bandwidth per channel is now reduced to  $\sim 50$  kHz.

(3) Inner product/time-integrating convolution: In this architecture, space integration is used for the vector index summation, and time integration is used for the convolution. The output vector  $\mathbf{h}$  is thus produced in a bit parallel and element sequential fashion. Each element of  $\mathbf{h}$  is fully calculated (all bits accumulated in parallel) after all the bits of a row of matrix are input to the processor, i.e., after thirty-one clock cycles of the input devices. The 1-D time-integrating detector array and associated ADC containing thirty-one elements will now have a bandwidth of approximately  $\sim 1.6$  MHz. The number of levels to be resolved still remains at 512.

(4) Scalar-vector product/space-integrating convolution: In this architecture, time integration is used for the vector-index summation and space integration for convolution. The output vector  $\mathbf{h}$  is thus produced in bit sequential and element parallel fashion. The optical system is not utilized in performing the time integration for the vector-index summation. That task is delegated to thirty-two digital accumulators (one for each element of the output vector  $\mathbf{h}$ ). Since the output of one channel of the acoustooptic space-integrating convolver corresponds to an analog bit stream representing one element of the scalar-vector product  $[g_i f^{(i)}]$ , it will have only sixteen resolvable levels (4 bits). Hence the 1-D high bandwidth detector array and associated A-D array will need to resolve only sixteen levels. Since a new sample in the output is produced for each new bit in the input, the bandwidth per channel will be 50 MHz. A shift-and-add circuit is needed behind each A-D converter to produce the time binary representation for each element of the scalar-velocity product. The binary words resulting from successive scalar-velocity products will then be added in an accumulator array, which will have to be 32 bits wide and operate at  $\sim 1.6$  MHz.

These four examples serve to illustrate the different trade-offs among the bandwidth, number of channels, and levels per channel associated with the postprocessing electronics including the detectors. The product of these three parameters is invariant among these four architectures and equal to  $2.56 \times 10^{10}$  levels/s.

This is also equal to the total input data rate for the processor (32-element vector, 16 bits/elements, 50-MHz bit rate/channel), which is to be expected since the vector-matrix multiplication is a linear transformation and does not involve any data reducing operations. The first three architectures demonstrate a trade-off between the bandwidth and number of channels while leaving the number of levels produced per channel constant. The fourth architecture reduces the number of levels per channel at the expense of increasing the number of channels and the bandwidth per channel. Since the number of levels to be reliably resolved by a detector is limited by the analog accuracy of the optical processor, this may seem to provide the best solution. A close examination reveals, however, that this trade-off is only achieved by performing the accumulation digitally, thus reducing the computation performed by the optical system. Thus the last architecture, although most practical, will suffer when compared with an all electronic architecture.

There are numerous other variations on these architectures for high accuracy optical processors that involve using outer products for digital multiplication<sup>23</sup> or using systolic or engagement architectures for vector-matrix multiplications. They will affect the device requirements and data-flow characteristics of the processor. But they will not change the overall picture concerning the required sophistication for the postdetection electronics.

The conclusion of this study of four specific architectures is that the number of levels per second that the optical processor needs to generate and the electronic postprocessor needs to handle is totally fixed by the computational throughput of the system. Thus the only way of achieving a very high throughput is to have a very high performance electronic postprocessor and a very high accuracy analog optical system. Both of these requirements negate the basic goal of employing digital encoding to build a high-throughput, high-accuracy optical processor that is far superior to an all-electronic implementation. Another secondary conclusion is that the only way of reducing the requirements on the analog accuracy of the optical system is to reduce the amount of computations performed by it (multiplication without the summation).

#### IV. Conclusion

It is quite apparent from the previous discussion that the use of linear analog optical processors in processing digitally encoded data leads to unacceptable requirements on the analog accuracy of the optical system and on the complexity and performance of the electronic postprocessor. As with other situations in life, a difficult task that cannot be avoided all together becomes progressively more difficult when it is postponed further. The nonmonotonic nonlinear operations are an integral part of processing digitally encoded data. The more operations one performs without this step, the more difficult the nonlinear operations become. On the other hand, if the electronic nonlinear operation is performed frequently, the role

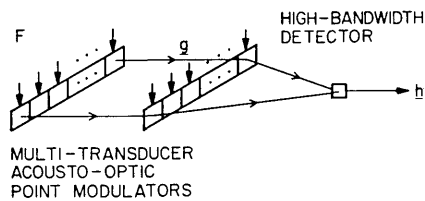


Fig. 9. Space-integrating analog processor for performing vector-matrix multiplication.

of optics diminishes compared to the electronics, putting in question the reason for using optics at all!

It is sometimes suggested that the use of a base larger than 2 for the fixed-radix digital number representation will require fewer channels for representing a large number and hence will lead to a more efficient optical processor. This will indeed be the case if we are only interested in minimizing the space-bandwidth requirement of the optical system to perform operations with given accuracy. As we saw in previous sections, however, the number of levels that need to be reliably produced by an optical system will also have to be minimized for a practical system. Therefore, the cost function will involve total levels required to represent a number with a given accuracy and can be chosen to be equal to (number of digits)  $\times$  (base-1). The minimum cost for a given accuracy has been shown to occur for base = 3 and is only 5% below the cost for base = 2.<sup>25</sup> Therefore, it is apparent that little will be achieved by going to a higher value for the base.

It will be instructive to compare the throughput of a purely analog optical system that uses the same technology as the four architectures described in Sec. III. Figure 9 shows a space-integrating inner/product processor built with thirty-two-channel point modulator arrays and a high speed photodetector. If we assume 50-MHz analog bandwidth/channel and 9-bit accuracy for the optical system, the computation throughput will be  $1.6 \times 10^9$  multiply adds/s at 9-bit accuracy. The output of the detector can be used in further stages of computation without additional postprocessing electronics. This throughput is highly attractive, especially when obtained with an optical processor that could be very compact and require low power.

The linear analog optical processors thus seem best suited for applications that do not demand high accuracy but put a premium on high computational throughput in a small volume with low power consumption.

The authors would like to thank R. C. Williamson of Lincoln Laboratory for pointing out the central role of the total number of levels produced by an optical system and for numerous insightful comments.

## References

1. L. J. Cutrona, E. N. Leith, L. J. Porcello, and W. E. Vivian, "On the Application of Coherent Optical Processing Techniques to Synthetic Aperture Radar," *Proc. IEEE* **54**, 1026 (1966).
2. T. M. Turpin, "Spectrum Analysis Using Optical Processing," *Proc. IEEE*, **79** (1981).
3. Special Issue on Acoustooptics, *Proc. IEEE* (Jan. 1981).
4. N. J. Berg, J. N. Lee, M. W. Casseday, and E. Katzen, in *Ultrasonics Symposium Proceedings*, IEEE Catalog No. 78 CH 134-1SU (1978), p. 91.
5. A. Huang, Y. Tsunoda, J. W. Goodman, and S. Ishihara, "Optical Computation Using Residue Arithmetic," *Appl. Opt.* **18**, 149 (1979).
6. A. Tai, I. Cindrich, J. R. Fienup, and C. C. Aleksoff, "Optical Residue Arithmetic Computer with Programmable Computation Modules," *Appl. Opt.* **18**, 2812 (1979).
7. D. Psaltis and D. Casasent, "Optical Residue Arithmetic: A Correlation Approach," *Appl. Opt.* **18**, 163 (1979).
8. S. A. Collins, Jr., "Numerical Optical Data Processing," *Proc. Soc. Photo-Opt. Instrum. Eng.* **128**, 313 (1977).
9. C. C. Guest and T. K. Gaylord, "Truth-Table Look-up Optical Processing Utilizing Binary and Residue Arithmetic," *Appl. Opt.* **19**, 1201 (1980).
10. R. Arrathoon and M. N. Hassoun, "Optical Threshold Logic Elements for Digital Computation," *Opt. Lett.* **9**, 143 (1984).
11. "Vedic Mathematics," Shankaracharya of Govardhana Pitha, Motilal Banarsidass Pub., New Delhi, ISBN: 0-89581-416-1 (1965).
12. E. E. Schwartzlander, Jr., "The Quasi-serial Multiplier," *IEEE Trans. Comput.* **C-22**, 317 (1973).
13. H. J. Whitehouse and J. Speiser, "Linear Signal Processing Architectures," in *Aspects of Signal Processing with Emphasis on Underwater Acoustics*, Vol. 2, G. Tacconi, Ed. (Reidel, Hingham, MA, 1977).
14. D. Psaltis *et al.*, "Accurate Numerical Computation by Optical Convolution," *Proc. Soc. Photo-Opt. Instrum. Eng.* **232**, 151 (1980).
15. P. S. Guilfoyle, "Systolic Acousto-optic Binary Convolver," *Opt. Eng.* **23**, 20 (1984).
16. W. C. Collins, R. A. Athale, and P. D. Stilwell, "Improved Accuracy for Optical Iterative Processor," *Proc. Soc. Photo-Opt. Instrum. Eng.* **352**, 59 (1983).
17. R. P. Bocker, "Optical Digital RUBIC (Rapid Unbiased Bipolar Incoherent Calculator) Cube Processor," *Opt. Eng.* **23**, 26 (1984).
18. K. Wagner and D. Psaltis, "A Space-integrating Acousto-optic Matrix Multiplier," *Opt. Commun.* **52**, 173 (1984).
19. A. P. Goutzoulis, "Systolic Time-Integrating Acoustooptic Binary Processor," *Appl. Opt.* **23**, 4095 (1984).
20. S. Cartwright and S. C. Gustafson, "Convolver-based Optical Systolic Architectures," *Opt. Eng.* **26**, 59 (1985).
21. C. M. Verber, "Integrated Optical Architectures for Matrix Multiplications," *Opt. Eng.* **24**, 19 (1985).
22. J. Jackson and D. Casasent, "Optical Systolic Array Processor Using Residue Arithmetic," *Appl. Opt.*, **22**, p. 2817 (1983).
23. M. S. Mort, "Modified Quasi-Serial Multiplier," *Appl. Opt.* **24**, 1396 (1985).
24. R. A. Athale, W. C. Collins, and P. D. Stilwell, "High Accuracy Matrix Multiplication with Outer Product Optical Processor," *Appl. Opt.* **22**, 368 (1983).
25. S. L. Hurst, "Multiple Valued Logic—Its Status and Its Future," *IEEE Trans. Comput.* **C-33**, 1160 (1984).