WILLIAM G. BATH

EVOLUTION AND CHALLENGES IN SIGNAL PROCESSING

Signal-processing theory continues to evolve toward discrete, nonrecursive models and solutions, and toward real-time adaptation. Hardware is evolving toward larger building blocks and programmability. There are challenges remaining in the areas of throughput, inherently nonlinear problems, and cost reduction.

BACKGROUND

The evolution of signal processing consists of the joint evolution of signal-processing theory and the electronic hardware used to implement the theory. During World War II, the classic work by Wiener on filtering signals in noise[1] established the theoretical basis for using signal and noise models to deduce the optimum filter components (that is, the resistance, capacitance, and inductance values for an optimum analog circuit) for what was then the state of the art. Wiener's spectral factorization method essentially showed how to build a circuit whose transfer function (that is, spectral response) "matched" the spectrum of the signal one was trying to detect.

Since that time there have been many more rigorous mathematical treatments of continuous-time random processes. However, electronic technology overtook the mathematical evolution. Bipolar and field-effect transistors were used first as discrete components and later within integrated circuits to build filters that were more reliable and which could realize theory with much finer precision. The pioneering work of Nyquist and Shannon[2,3] established that any practical continuous-time signal could be represented with a finite number of samples obtainable by sampling at a rate at least twice the bandwidth of the analog signal, and then quantizing those samples with a sufficient number of levels. Today (with a few notable exceptions) signal processing is digital (that is, signals are represented by discrete samples in both time and amplitude). Signal-processing designers attempt to sample and quantize signals in the rawest form possible.

The evolution of signal processing brought about by digital electronics has greatly simplified the theory of signal processing, transforming the integral equations that Wiener solved into matrix equations and eliminating many thorny mathematical questions of existence and uniqueness. However, signal-processing theory and the electronic hardware continue to evolve. Applications requiring large numbers of computations per unit time and the availability of components to do such computations have both been an impetus to evolution.

Initial digital signal-processing approaches were recursive, employing feedback loops, as the analog circuits had. Those recursive approaches were very efficient in their use of digital hardware. A considerable body of theory was developed to approximate the recursive analog solutions with recursive (that is, infinite impulse response) digital solutions. However, as nonrecursive (that is, finite impulse response) solutions became technologically feasible, they rapidly supplanted recursive solutions because they were simple, unconditionally stable, and frequently better models of real-world signals and noise.

The basic ideas of signal processing that were derived from analog filtering have proven helpful in solving signal-processing problems without clear analog roots. In solving the inherently nonlinear problems of pattern recognition, sorting, and classification, one strives to do what the human eye can do, given the right display to look at. Those problems too were initially attacked recursively and have evolved to nonrecursive forms. Those problems have been a second impetus to the development of signal-processing theory and hardware.

This paper will deal entirely with digital signal processing, and principally with nonrecursive models and solutions. The principles will be presented in the time domain (as opposed to the more classical frequency-domain presentation) to emphasize the evolution toward nonrecursive solutions, and will be presented from a statistical point of view to emphasize the importance of noise models and adaptation in signal-processing solutions.

BASIC PRINCIPLES

Most signal processing can be viewed as digital matched filtering—a digital version of how Wiener viewed it. Consider the measured, sampled time signal **x** shown in Fig. 1. That signal is the sum of a signal of interest **s** and a random noise process **n**—both column vectors of length N:

$$x_i = s_i + n_i, \qquad i = 0, \ldots, N-1. \qquad (1)$$

The matched filter output y_m (a scalar) is formed by multiplying the measured signal by the complex conjugate of the signal of interest and summing over time:

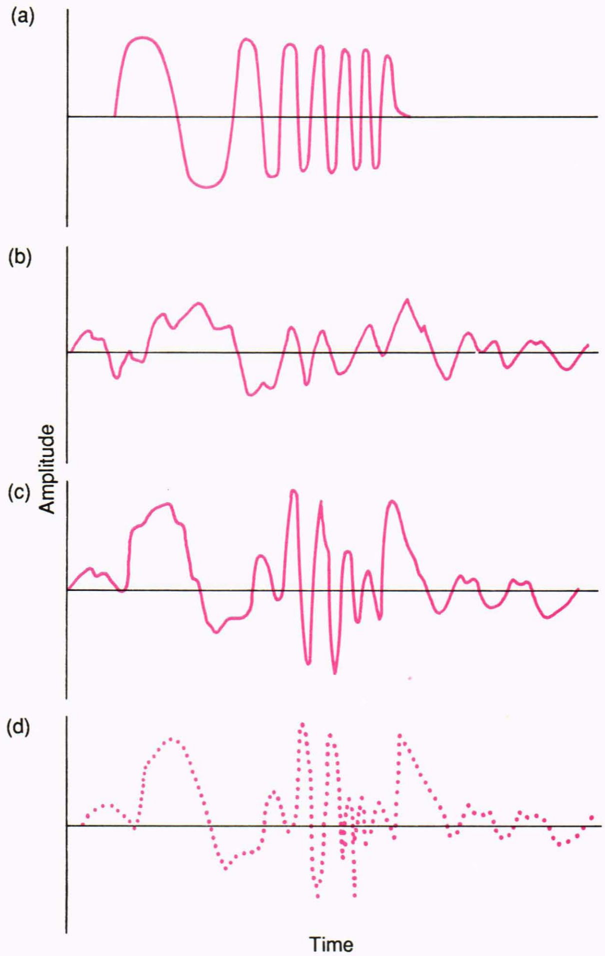


Figure 1—The basis of digital matched-filtering theory is a signal of interest (a) and noise (b) with known statistical properties. The sum of signal and noise (c) is sampled (d) and processed digitally.

$$y_m = \mathbf{x}^T \mathbf{s}^* = \sum_{i=0}^{N-1} s_i^* x_i \ , \quad (2)$$

where the superscript T denotes transpose and * indicates complex conjugate. Complex signals arise when the analog signal that was sampled is a bandpass signal represented as a real (in-phase) component and an imaginary (quadrature) component.

One reason the application of the matched filter is so broad is that it is optimum in many senses. Two of the most important will be discussed. The first relates to detecting the presence of a signal, a common criterion used in surveillance systems. If the signal is deterministic, and the noise has a Gaussian probability distribution $f_{\mathbf{n}}(\boldsymbol{\alpha})$ with zero mean and diagonal constant covariance matrix $\boldsymbol{\Lambda}$:

$$f_{\mathbf{n}}(\boldsymbol{\alpha}) = \frac{1}{(2\pi)^{N/2} |\boldsymbol{\Lambda}|^{N/2}} \times \exp\{-½(\boldsymbol{\alpha} - \mathbf{m})^T \boldsymbol{\Lambda}^{-1} (\boldsymbol{\alpha} - \mathbf{m})^*\} \ , \quad (3)$$

where $\mathbf{m} = 0$ and $\boldsymbol{\Lambda} = \sigma^2\mathbf{I}$ ($\mathbf{I}$ is the $N \times N$ identity matrix), then the matched filter is optimum. By optimum, it is meant that comparing the scalar y_m to a threshold produces the highest probability of detection (detecting the presence of the desired signal when it is present) for a given probability of false alarm (mistakenly detecting the presence of the desired signal when it is absent). This fact is a consequence of the Neyman–Pearson lemma, which states that the optimum test statistic is the ratio of the probability densities of the measurements with and without the desired signal present:

$$y_{opt} = \frac{(2\pi)^{-N/2} |\boldsymbol{\Lambda}|^{-N/2} \exp\{-½[(\mathbf{x} - \mathbf{s})^T \boldsymbol{\Lambda}^{-1} (\mathbf{x} - \mathbf{s})^*]\}}{(2\pi)^{-N/2} |\boldsymbol{\Lambda}|^{-N/2} \exp\{-½\mathbf{x}^T \boldsymbol{\Lambda}^{-1} \mathbf{x}^*\}} \ . \quad (4)$$

A little arithmetic shows that y_{opt} and y_m are monotonically related and so they are equivalent for detection purposes. Combined with the Gaussian assumption, that is a powerful result. As simple as the matched filter is, there is no better way of processing $\mathbf{x}$.

A second sense in which the matched filter is optimum is maximizing the signal-to-noise ratio, a common criterion used in guidance or tracking processes where one is estimating signal parameters. The matched filter is the optimum linear solution in this sense, even when the noise is not Gaussian. If one combines the individual measurements linearly to produce a scalar y, then

$$y = \mathbf{w}^T \mathbf{x} = \sum_{i=0}^{N-1} w_i x_i \ . \quad (5)$$

If one assumes that the noise has zero mean but constant variance ($\boldsymbol{\Lambda} = \sigma^2\mathbf{I}$), then one can define the signal-to-noise ratio (S/N) as the ratio of the squared-mean of y (which is from the desired signal) to the variance of y (which is from the noise):

$$\text{S/N} = \frac{E[y]^2}{\text{Var}\ (y)} = \frac{\mathbf{w}^T \mathbf{s}}{\mathbf{w}^T \boldsymbol{\Lambda} \mathbf{w}^*} \ . \quad (6)$$

Taking the gradient with respect to $\mathbf{w}$ gives a maximum signal-to-noise ratio (S/N_{max}) of

$$\text{S/N}_{max} = \frac{\sum_{i=0}^{N-1} |s_i|^2}{\sigma^2} = E_s/\sigma^2 \ , \quad (7)$$

at $\mathbf{w} = \mathbf{s}^*$, which is the matched filter. The matched-filter signal-to-noise ratio is the ratio of the energy E_s in the signal to the variance of the noise, and completely determines the detectability of the signal. That is, in Gaussian noise all signals of equal energy are equally detectable, regardless of shape.

When the noise is not white, then it is straightforward to show that the optimum "colored" filter (in both senses of optimality) is

$$y_c = \mathbf{x}^T(\mathbf{\Lambda}^{-1}\mathbf{s}^*) \ . \tag{8}$$

Such a filter can be implemented in two stages—first a pre-whitening stage,

$$\mathbf{x}_I = \mathbf{\Lambda}^{-1}\mathbf{x} \ , \tag{9}$$

followed by the usual matched filter,

$$y_c = (\mathbf{x}_I)^T\mathbf{s}^* \ . \tag{10}$$

This, of course, requires one to know the covariance matrix, $\mathbf{\Lambda}$, of the noise (or, equivalently, either its discrete spectrum or the power spectral density of the analog noise sampled to produce it).

In some applications the noise is not Gaussian, or instead of being added to the signal it is combined with the signal in a more complicated way. Theoretical solutions exist to such problems and those solutions involve nonlinear combinations of the data. In most practical applications, those nonlinear solutions are approximated as adjustments to the linear-matched-filter solution. For example, if the signal and noise have been multiplied together (instead of added) prior to measurement, then one may operate on the logarithm of measurements with a matched filter (cepstrum processing). The logarithm makes signal and noise contributions additive. If the noise has a probability density function with longer tails than a Gaussian density function (for example, the noise occasionally has extremely large values) then a nonlinear process may be used to delete or de-emphasize outliers that deviate greatly from an estimated signal.

UNKNOWNS AND ADAPTATION

If the simplicity of the matched (and colored) filters seems inconsistent with the huge computational resources frequently required in signal processors, the reason is that, in general, many of the desired signal and noise parameters are not known. When such is the case, there are two alternatives—enumeration and estimation. One enumerates all possible solutions by building matched filters for all possible values of the unknown signal, passing the measured signal through each, and selecting the largest output (or, equivalently, thresholding all the outputs). One classical unknown desired signal characteristic is time, in which case one must try each possible time and the matched filter becomes a finite impulse response filter convolved with the measurements:

$$y_k = \sum_{n=0}^{N-1} x_{k-N+n}s_n^* \quad \text{for each possible } k \ . \tag{11}$$

A second classical unknown desired signal characteristic is the frequency, f, of a sinusoidal signal sampled at times nT,

$$s_n = \exp(-j2\pi fnT), \quad n = 0, 1, \ldots, N-1 \ . \tag{12}$$

In that case, if one spaces the frequencies in steps of $(NT)^{-1}$ then the matched filters become the discrete Fourier transform,

$$y_k = \sum_{n=0}^{N-1} x_n \exp(-j2\pi nk/N), \quad k = 0, 1, \ldots, N-1 \ , \tag{13}$$

and the calculation of the N matched filters can be made via the fast-Fourier-transform technique.

The second option for dealing with unknown parameters is estimation, which is usually the case for the noise covariance $\mathbf{\Lambda}$. If one can obtain a look at the noise uncontaminated by the signal or with minimal contamination (for example, a stationary noise sample with length much greater than N) then the covariance matrix can be estimated by estimating the $N(N+1)/2$ unique variance and covariance elements of which it is comprised (Fig. 2a).[7] For example, if one obtains M "snapshots" of the noise, $\mathbf{m}_i$, each of length considerably greater than N, then the classical estimate of the covariance matrix of N noise samples $\mathbf{\Lambda}_M$ is

$$\hat{\mathbf{\Lambda}}_M = \frac{1}{M}\sum_{i=0}^{M-1} \mathbf{m}_i^*\mathbf{m}_i^T \ . \tag{14}$$

Such a process of estimating the noise properties and then adapting to them is termed adaptive filtering, and is employed in some degree in almost all signal-processing applications. In many cases one simplifies the problem and stabilizes the computations by allowing the covariance matrix to have only a small (smaller than N) number of degrees of freedom. This is done by assuming a model, such as an autoregressive moving average model, with less than $N(N+1)/2$ parameters. The estimation of covariance matrix then consists of fitting the model to the measurements $\mathbf{m}_i$ by methods such as maximum entropy[5] or maximum likelihood.[6]

In some applications, the noise is colored in a simple way with a few degrees of freedom. Further, one can obtain a look at noise $\mathbf{m}$ that is minimally contaminated by the signal and not only has similar statistics to the colored portion of the noise $\mathbf{n}$, but is highly correlated with that portion. This situation is common when an array of sensors operating together as an antenna aperture experiences colored noise in the form of a small number of interfering plane waves. Then one can use this auxillary noise $\mathbf{m}$ to reduce the noise power of $\mathbf{n}$ by subtracting the correlated portion producing $\mathbf{x}_c$, a cancelled version of $\mathbf{x}$,

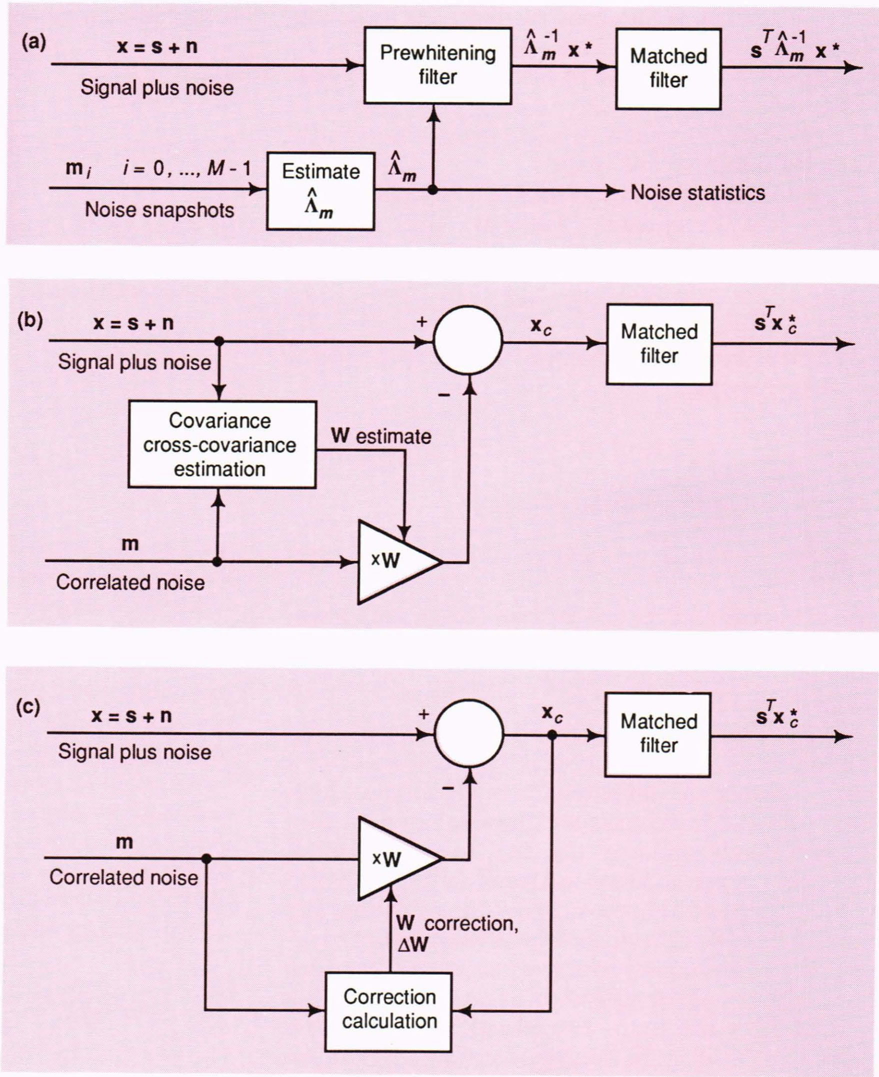


Figure 2—Three methods of adaptation in digital signal processing are (a) nonrecursive adaptation to noise covariance, (b) nonrecursive cancellation of correlated noise, and (c) recursive (feedback) cancellation of correlated noise.

$$\mathbf{x}_c = \mathbf{x} - \mathbf{W}\mathbf{m} \ , \tag{15}$$

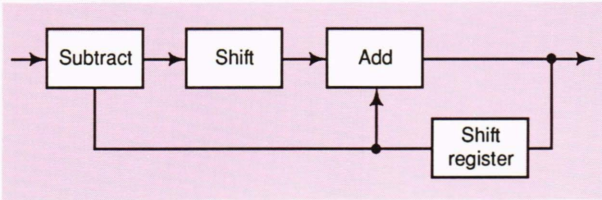
where the gain matrix **W** is estimated either in a nonrecursive configuration by estimating the cross-covariance matrix between **x** and **m** (Fig. 2b) or in a recursive configuration by adjustment of **W** via the Widrow–Hoff algorithm[7] modified-steepest-descent method (Fig. 2c).

Adaptive filtering in any form is a delicate process. One is inverting a covariance matrix (either directly or indirectly) that is frequently nearly singular, then using the answer to construct the filter coefficients. Errors of a few degrees of phase angle or a fraction of a decibel of amplitude can be catastrophic, a consideration that influences the design of the signal extraction and analog-to-digital conversion design, and which frequently requires many checks of reasonableness and stability that are tailored to the specific application. Selecting the window over which adaptation occurs (for example, the number of samples used in a feed-forward and prewhitening adaptation or the time constant used in a feedback adaptation) is a compromise between precision and the ability to adapt to nonstationary noise.

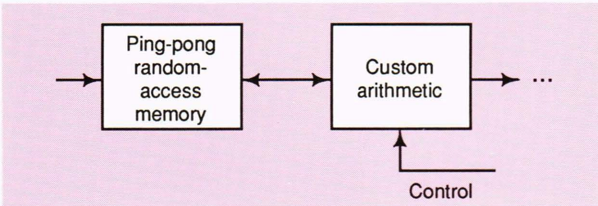
IMPLEMENTATION AND COST

The hardware used to implement digital signal processing has evolved toward larger building blocks and toward firmware and software rather than hard-wired circuits. Registers were used for storage (Fig. 3a) when memory was at a premium. Multiplications were conserved through recursive implementations and shifts when possible. With the advent of less expensive high-speed random-access memory, recent designs (Fig. 3b) store data in random-access memory and manipulate it prior to passing the answer to the next stage of the pipeline. High-speed multiplier–accumulators make nonrecursive implementations practical. Coefficients and addresses can be variable—either downloaded or selected from programmable read-only memories via tailored logical operations.

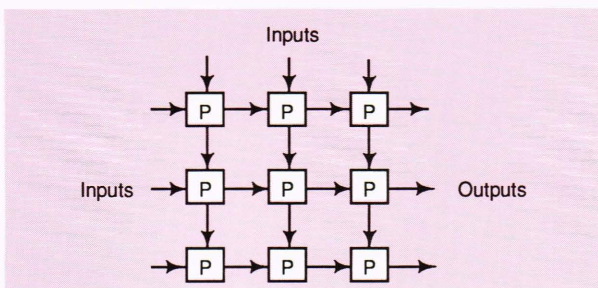
In many applications the gains of signal processing justify a large investment financially as well as in size, weight, and power consumption. Despite continued evolution to progressively larger building blocks, a standard signal processor—while frequently discussed as a way to economize—has proven elusive. One reason is that



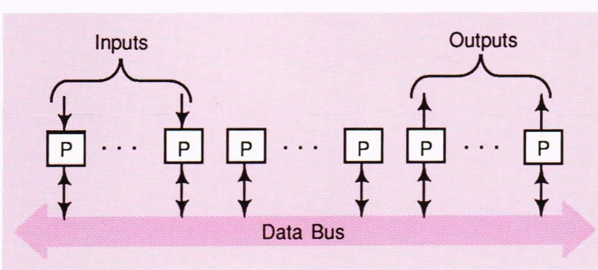
(a) One stage of a digital recursive filter using discrete components.



(b) One stage of a modern digital signal processor. Custom circuit multiplexes one or more high-speed multiplier/accumulators plus controlled coefficient read-only memories and tailored logic.



(c) An array of processors, each with its own random-access memory, programmable arithmetic logic unit, and firmware.



(d) A bussed array of processors.

Figure 3—Evolution of signal processing implementations.

in almost all practical examples there are application-specific functions that pervade classical processing. For example, in a military radar one is likely to have distributed throughout the signal processor various checks to identify deception jammers; such checks are specific to the application. A similar reason deals with extreme deviations of the noise from the model assumed in formulating the optimum filter. For example, in radar, radio-frequency interference (from friendly sources) and some environmental reflections can be many orders of magnitude larger in amplitude than the signal of interest, so as to make the signal undetectable. The methods for recognizing and confining the effects of such problems are specific to the application as well. These application-specific functions can require a significant portion of the total signal-processing power and are not easily standardized.

Standard building blocks may be a more realistic way to economize. Arrays of processors (Fig. 3c), with each processor being a standard mass-produced (and therefore, in theory, more economical) item have been extensively studied in recent years. Some processing algorithms (such as matrix multiplication, matrix inversion, and ordering lists of numbers) have an inherent degree of parallelism, which leads to efficient ways to divide the tasks among a large number of processors with relatively few interconnections per processor. Research into systolic arrays,[8] regular iterative algorithm arrays,[9] wave-front processors,[10] and neural networks try to improve efficiency, minimize connectivity, and broaden the classes of algorithms that can be attacked in this manner.

An alternative to ease the connectivity problem is "soft" connection via a data bus[11] (Fig. 3d). With a data bus, every processor can communicate with every other processor. The total volume of communications is fixed by the bus hardware, speed of the memory from which and to which data is taken off and put on the bus, and the method of bus arbitration.

A SIGNAL PROCESSING CHALLENGE—ALL-DIGITAL RADAR WITH THREE-DIMENSIONAL MATCHED FILTERING

An example of a challenging signal-processing problem that is technologically unwieldy today is that of a complex radar system having a large number of array elements, a large signal bandwidth, and open-loop adaptation in which all processing is done digitally. Conceptually, one could organize the measurements in a three-dimensional array such as that in Fig. 4. One dimension represents space (N_A array elements or subarray outputs), another dimension represents time or range (N_R range elements), and the third dimension represents iterations (for example, N_I coherent dwells). One could then view the length of a matched filter as the entire three-dimensional array $N = N_R N_I N_A$ and perform matched filtering across all three dimensions simultaneously. That has been done with simple experiments or in real high-frequency radar systems in which both N_A and the signal bandwidths are small. Today's high-bandwidth radar developments generally decouple the problem into, at most, three uncoupled or loosely coupled problems—beam forming (analog in the spatial dimension), pulse compression (analog or digital in the range dimension) and Doppler filtering (digital in the iteration dimension). Coupling, when it occurs, is very structured; for example, extra pulse-compression channels to correct for range–Doppler coupling effects in phase-coded or nonlinearly frequency-coded signals. Significant coupling of even two dimensions, such as the range–Doppler coupling in synthetic-aperture radar, is a challenge today in real time. The three-dimensional problem would require orders of magnitude more complex multiplies per second than one can currently obtain. The dig-

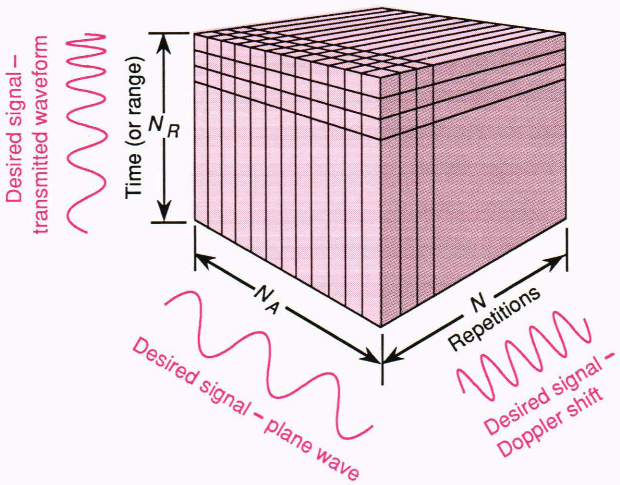


Figure 4—Conceptual data array for all-digital, fully coupled, radar signal processor.

ital beam-forming problem itself is formidable. Consider a 1000-element array with a 10-MHz signal bandwidth. The number of complex multiplies for the most straightforward implementation would be 10^{10} per second. This problem is likely to be attacked gradually, first with the develpment of practical subarray digital beam forming and then with progressively more complex coupling of the spatial processing with the range and Doppler dimensions and with progressively more degrees of freedom in the estimation of the noise covariance. The spatial dimension is potentially amenable to distributed processing in the form of a processor at each array element with interconnections to adapt and couple with other dimensions.

As with most of today's signal-processing problems, the theoretical foundations are well established. Many of the challenges are in systems engineering, which develops the goals and constraints on the signal-processing problem and with the components used to build the signal processor.

A SIGNAL-PROCESSING CHALLENGE—NONRECURSIVE MATCHED FILTERING OF EVENTS

Considerable challenges remain in signal processing that is inherently nonlinear. When the signal of interest is a set of events that must be sorted from a larger collection of signal and noise events, the theory is more complex but many of the same principles apply. Figure 5 shows a one-dimensional event signal buried in noise. Most people quickly identify it with their eye after being told in very general terms what constitutes a desired signal. Here it is a line of dots (rising from left to right at the top). Developing and building a processor to do the same may at first seem unrelated to matched filtering, but the event-matched filtering problem can be formulated and solved in a similar (though not a linear) manner.

In general, the axis x is a vector and contains all the information measured about an event occurring at time t_i. Thus a signal (event chain) of interest can be represented by a set of ordered pairs $s = \{(\mathbf{s}_i, t_i) : i = 0, ..., N - 1\}$ indicating the event coordinates and time. The measured set of events, x_i, at each t_i is the union of randomly detected desired-signal events with coordinate noise added and extraneous events:

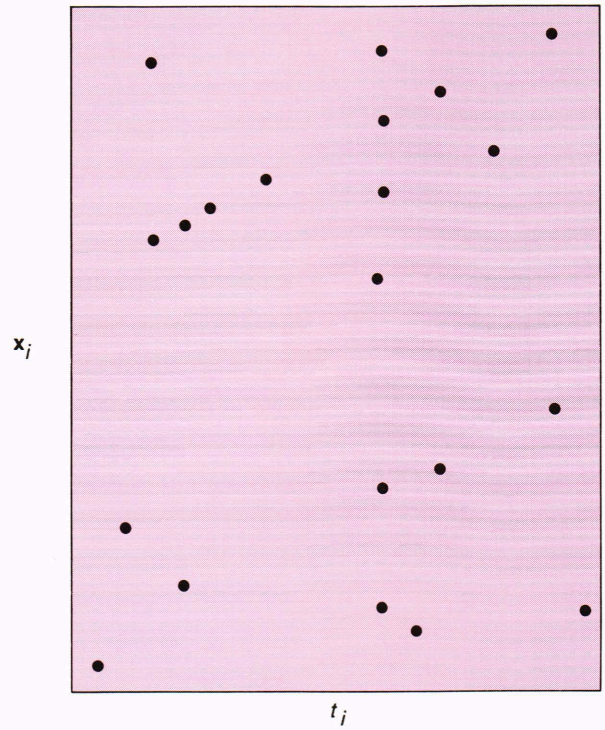


Figure 5—An event signal in noise (one-dimensional coordinate).

$$x_i = \begin{cases} \{\mathbf{x} : \mathbf{x} = \mathbf{s}_i + n\} \cup \{r_j : j = 0, ..., N_{ri}\} \\ \qquad\qquad \text{with probability } p \\ \{r_{ij} : j = 0 ..., N_{ri}\} \text{ with probability } 1 - p \ , \end{cases} \tag{16}$$

Where n is noise with zero mean and covariance matrix Λ, and r_i are random coordinates.

The event-matched filter is

$$y_e = \sum_{i=0}^{N-1} \mathbf{s}_i \wedge \chi_i \ , \tag{17}$$

where y_e is a scalar and $\wedge$ is the proximity sort operation. The $\wedge$ operator looks through the entire set χ_i for any coordinate matching the desired signal coordinate $\mathbf{s}_i$. A common form of the proximity sort is a statistical distance sort,

$$\mathbf{s}_i \wedge \chi_i = g\left[\min_{\text{all } \mathbf{x} \in \chi_i} (\mathbf{s}_i - \mathbf{x})^T \Lambda^{-1} (\mathbf{s}_i - \mathbf{x})^*\right] , \tag{18}$$

where $g(x)$ is an influence function selected to weight the individual contributions. That sort can be shown (again through the Neyman–Pearson lemma) to be the most powerful (in a probability-of-detection and false-alarm sense) proximity sort when the random coordinates are uniformly distributed over some range and $g(x)$ is chosen to be the identity function.

An event-matched filter requires a number of computations that are quadratic in the size of χ_i. The coefficient of the quadratic term can be minimized by presorting the data into sectors. Generally, each sector constitutes a linked list through the data base.

As in the linear matched filter, most of the complexity of event filtering comes from not knowing the signal **s** in advance. Referring to Fig. 5, consider all the possible signals that the human eye could recognize when they are immersed in random events. As in the linear case, the two basic approaches are enumeration and estimation. One case of considerable practical interest is an event with coordinates varying linearly with time,

$$\mathbf{s}_i = \mathbf{a}t_i + \mathbf{b} \ , \tag{19}$$

where the t_i are equally spaced, and where **a** is constrained in D dimensions,

$$|a_i| < \beta_i \qquad i = 0, \ldots, D - 1 \ , \tag{20}$$

and where s_i is constrained to lie in a region of D-dimensional space with volume G, and the expected number of random events at any t_i is N_R.

This problem has been extensively studied in its recursive formulation. Recursive solutions try to identify candidate signal starts, then bootstrap the problem by estimating derivatives with respect to time, predicting the future coordinates using those derivatives, and associating coordinates with those predictions (Fig. 6a). Many real-time processing systems to automate the detection of signal events have been built on such principles.

As with the linear filtering problems, processing technology has made nonrecursive solutions practical. A nonrecursive solution takes the enumeration approach rather than the estimation approach (Fig. 6b). One can think of constructing event-matched filters for all possible circumstances. With the event coordinates varying linearly with time, this would require **s** vectors covering all allowable slopes and intercepts with granularity matched to the noise covariance **Λ**. One would sort the data with respect to each possible **s** vector to find and threshold the largest y_c. This is an enormous processing task. Fortunately, the sparseness of the matrix in Fig. 5 can be exploited to make the problem practical. R. J. Prengaman of the Applied Physics Laboratory developed the retrospective processing technique[12] that solves this massive sorting problem in an efficient manner. Initial real-time implementation on a Motorola 6800 microprocessor[13] demonstrated the efficiency and practicality of the nonrecursive solution.

The recursive and nonrecursive solutions can be shown to be nearly equivalent when the density of extraneous events is small enough.[14] As the density of extraneous events increases, the recursive solution begins to experience ambiguities in the association process and must be made more complex by propagating multiple hypotheses. It is in these situations that the nonrecursive solutions are more powerful, because decisions do not have to be made piecemeal.

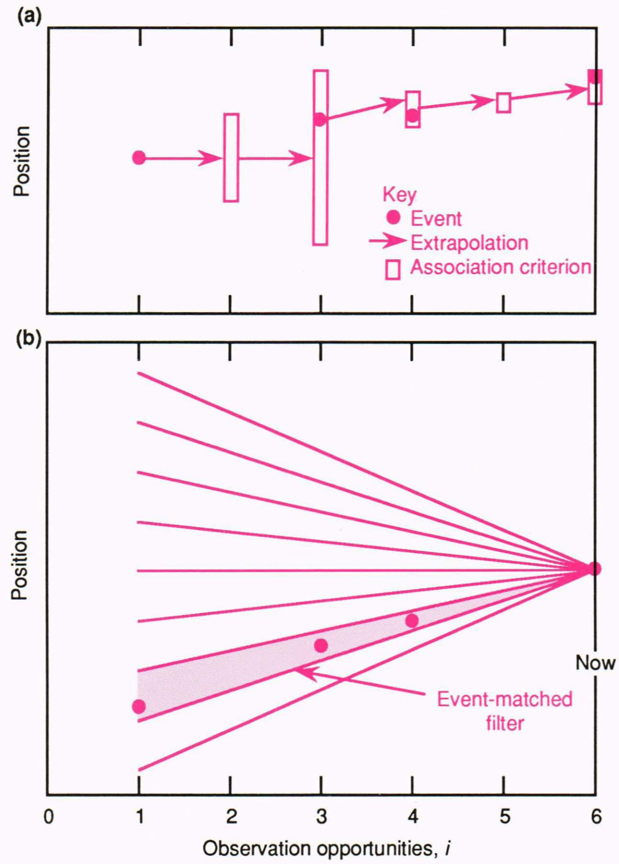


Figure 6—Solutions to the event-detection problem using the proximity-sort operator. A recursive solution (a) bootstraps the problem. A nonrecursive solution (b) implements all possible event matched filters (at least functionally).

In most practical event-filtering problems, the coordinates do not fit a linear model. This poses little increase in complexity if one knows what model the coordinates do fit. More often, one knows less and the coordinate constraint contains several unknown parameters. For example, in Fig. 5, suppose the signal could be any smooth curve, where one can postulate various definitions of smooth that would produce a number of unknown parameters between 2 and N. The number of event-matched filters immediately grows by several orders of magnitude. Practical nonrecursive event filtering with many unknown signal parameters is a challenging problem that will require further evolution of both theory and implementations.

If the coordinate constraint is locally linear, and the density of false alarms low enough, one can use a cascaded processor[14] that consists of a short-time-base nonrecursive processor (working on a linear constraint) followed by a recursive processor (estimating unknown parameters and extrapolating).

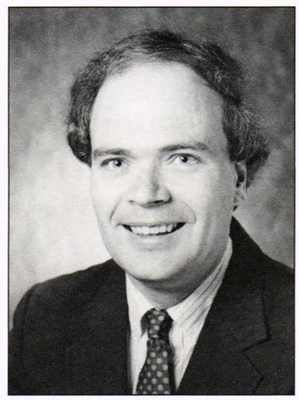
SUMMARY

The declining cost, weight, size, and power consumption of digital processing and random-access memory, combined with signal-processing applications in which the desired signal and interfering-noise characteristics must be estimated in real time, have caused signal processing to evolve toward digital, nonrecursive solutions. These solutions are typically based on *a priori* or estimated statistical models. Signal-processing implementations have evolved toward larger building blocks and toward firmware and software rather than hard-wired circuits. Considerable challenges remain in obtaining the digital throughput required for high-bandwidth, many-channel systems; in developing approaches for inherently nonlinear problems; in further reducing cost by standardizing building blocks; and in the systems engineering necessary to match signal-processing technology to requirements.

REFERENCES

[1] N. Wiener, *Extrapolation, Interpolation and Smoothing of Stationary Time Series*, Wiley, New York (1949).
[2] H. Nyquist, "Certain Topics in Telegraph Transmission Theory," *Trans. Am. Inst. Electr. Eng.* **47**, 617–644 (1928).
[3] C. E. Shannon, "Communication in the Presence of Noise," *Proc. Inst. Radio Eng.* **37**, 10–21 (1949).
[4] S. P. Applebaum, "Adaptive Arrays," *IEEE Trans. Antennas Propag.* **AP-24**, 585–598 (1976).
[5] J. P. Burg, "Maximum Entropy Spectral Analysis," in *Proc. 37th Meeting Society of Exploration Geophysicists* (1976).
[6] J. Capon, "High-Resolution Frequency-Wavenumber Spectral Analysis," *Proc. IEEE* **57**, 1408–1418 (1969).
[7] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive Antenna Systems," *Proc. IEEE* **55**, 2143–2159 (1967).
[8] H. T. Kung and C. E. Leiserson, "Systolic Arrays for VLSI," in *Sparse Matrix Proc.*, Society of Industrial and Applied Mathematics, pp. 245–282 (1978).
[9] S. K. Rao and T. Kailath, "Regular Iterative Algorithms and Their Implementation on Processor Arrays," *Proc. IEEE* **76**, 259–269 (1988).
[10] S. Y. Kung, "Wavefront Array Processors," *Computer* **20**, 18–33 (1987).
[11] H. M. South, "Digital Recording and Signal Processing Systems for Hydrophone Arrays," *The Johns Hopkins APL Tech. Dig.* **4**, 212–218 (1983).
[12] R. J. Prengaman, R. E. Thurber, and W. G. Bath, "A Retrospective Detection Algorithm for Extraction of Weak Targets in Clutter and Interference Environments," in *Proc. 1982 IEE Internat. Radar Conf.*, London, pp. 341–345 (1982).
[13] W. G. Bath, L. A. Biddison, and R. E. Thurber, "Noncoherent Subclutter Visibility through High-Speed Contact Sorting," in *Proc. 1984 Military Microwaves Conf.*, London, pp. 621–624 (1984).
[14] W. G. Bath, M. E. Baldwin, and W. D. Stuckey, "Cascaded Spatial Temporal Correlation Processes for Dense Contact Environments," *Proc. 1987 IEE Internat. Radar Conf.*, London, pp. 125–129 (1987).

THE AUTHOR



WILLIAM G. BATH received B.E.S. and Ph.D. degrees in electrical engineering from The Johns Hopkins University in 1974 and 1980, respectively. Since 1974, Dr. Bath has been employed by APL, working in the areas of systems analysis, signal-processing techniques, and information-processing algorithms for Navy radars. His experience includes the development of automatic detection and tracking algorithms, evaluation of the effectiveness of various deployed Navy radar systems, development and testing of techniques for multiple platform radar integration, and research into coherent and noncoherent signal processing to enhance detection in clutter and interference. He is supervisor of the Engineering Analysis Group at APL.