Digital Signal Processing:

A DSP Primer

Although digital signal processing is fast becoming the 'hottest' growth area in electronics, it's still something of a mystery even to many people working in the industry. Here's an explanation of the basic concepts of DSP, and a look at some of its applications...

by TED DINTERSMITH
Analog Devices, Inc.

Digital signal processing or *DSP* means the processing of 'analog' signals using digital techniques – i.e., digital hardware and software.

It should perhaps be noted here that *any* electrical signal is by nature an analog signal, even if it represents a digital '1' or '0'. This can be understood if one pictures what happens to a chain of 1's and 0's returned from beyond the orbit of Jupiter, buried in cosmic noise: the signal must be received, amplified, converted and processed (digitally) to reconstrt the original digital information; but until that digital information has been identified, the signal is to all intents and purposes a purely analog one.

While *any* digital processing of signals that originate in the analog world, and have at some point been converted into digital form, would qualify for this broad definition, the term DSP has come to be used in a much more specific way. Nowadays, DSP is the the application of fast, specialised hardware, sophisticated algorithms and the appropriate software for the purpose of manipulating large amounts of data associated with extracting and processing analog-based information in essentially *real time*.

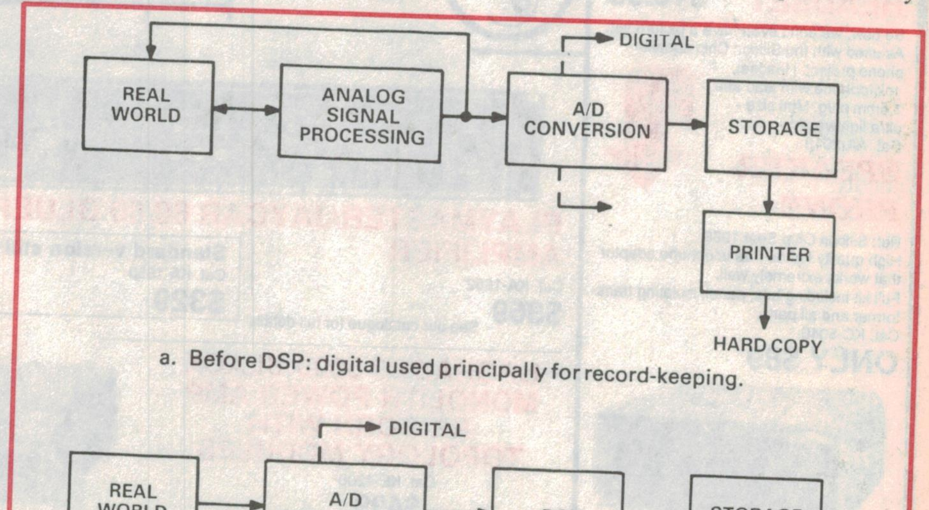
The emergence of DSP hardware is changing the role of analog-to-digital conversion in today's signal processing systems. In early days, all processing of a signal, with the goal of obtaining results with sufficient speed to be useful in real time, was of necessity handled by analog components. The principal destinations for analog signals converted to digital format, after substantial analog processing, were off-line computation, data storage, and hard-copy tabulation, rather than real-time instrumentation, computation, and control.

Now, however, system designers have an incentive to perform the signal conversion as early in the loop as possible (see Fig.1). The reason for this is that much or all of the required signal processing can be handled by fast, flexible digital components that allow high-performance DSP routines to be implemented more accurately, reliably, and flexibly than with analog circuitry, yet, in many cases, with sufficient speed to interact in real time.

This article first reviews basic signal processing tasks, giving emphasis to the general role played by DSP. Two key DSP algorithms are examined in some detail – digital filters and spectral analysis. The basic hardware required to perform DSP is described. Finally, some applications that exemplify DSP's advantages are reviewed.

DSP basics

Signal processing revolves around two basic tasks – digital filtering and spectral analysis. *Filtering* smoothes, removes noise from, selects particular signal components from, or predicts future values of an incoming signal. A time-domain signal can be interpreted as a weighted combination of purely sinusoidal spectral components; *spectral analy-*



a. Before DSP: digital used principally for record-keeping.

b. DSP era: digital responds to and operates on real-world phenomena.

***Fig.1:** As DSP technology has developed, designers have been able to convert to digital earlier in the signal processing chain.*

sis determines the weights corresponding to each frequency in the spectrum.

Signal-processing applications span many areas, including speech analysis and synthesis, telecommunications, instrumentation, radar and sonar, and – using multi-dimensional techniques – graphics and imaging. For example, filtering is used to minimize high-frequency noise and the low-frequency hum in telephone-line transmission. Spectral analysis is used to determine the format content of incoming speech for recognition. Two-dimensional filtering improves the clarity of a satellite image.

Filtering and spectral analysis have traditionally been implemented with analog components. Filtering is carried out by passing the signal through a circuit consisting of resistors, capacitors, op-amps, and/or inductors; the precise configuration of these components and the relationship of the magnitudes of their parameters determine the filter's characteristics. Multiple analog filters – each passing energy in a narrow band – can be cascaded for sharpness and banked together to perform spectrum analysis.

Analog-based signal processing has numerous advantages, including low component cost, the ability to handle wide bandwidths in real time, the availability of pre-packaged modules and ICs, and a large existing base of knowledge. However, analog components introduce noise at each stage; and filter characteristics – requiring effort to tune initially – are sensitive to the effects of temperature and aging. In addition, multi-stage filters pose subtle design challenges. Because coefficients and configurations – once established – tend to be inflexible, signal-processing hardware using analog parts is generally restricted to performing a narrow, dedicated task.

In response to the limitations of analog-based processing, the digital processing of signals has emerged as an alternative. The next section demonstrates how signal-processing tasks – including filtering, spectral analysis, and a host of others – can be carried out with digital arithmetic operating on digitised data.

Recent advances in VLSI (very large-scale intergration) now make is feasible to perform real-time digital signal processing with just a handful of ICs. The advantages conferred upon a system by such DSP hardware are dramatic – substantially improved performance, stability, and flexibility. Just as digital computers supplanted analog computers two decades ago in general-purpose computing applications, DSP is strongly challenging analog circuit configurations in real-time processing.

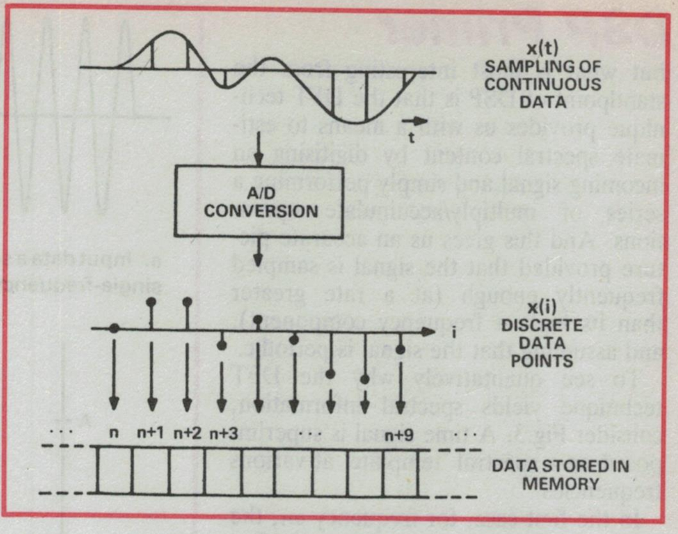


Fig.2: In DSP, continuous data is replaced by sampled data and continuous time by discrete time.

Our discussion of spectral analysis and digital filtering will benefit from a brief discussion of DSP nomenclature (there is also a brief glossary at the end of this article). Following Fig.2, an incoming analog signal is digitised, with the sampled data output points denoted x(i). The index, i, corresponds to the discrete sampling time. This sampled data is stored in a buffer and operated on by DSP hardware. The DSP algorithm determines the sequence in which data and coefficients are accessed and how they are processed. In the cases below, the computational outputs are spectral weights or filtered sampled data.

Spectral analysis

Digital spectral analysis essentially derives from the principle of *Fourier transformation*. Without going into the mathematics here, this is a mathematical technique of taking a signal's time-domain representation and resolving it into its equivalent frequency-domain spectral weights - called Fourier coefficients.

Since the basic Fourier equations linking the time and frequency domains require continuous signals, they have only indirect bearing on digital processing. However under certain circumstances, a sampled (digitised) signal can be related faithfully to its Fourier coefficients through the so-called *discrete Fourier transform* (DFT).

Again we needn't go into the maths,

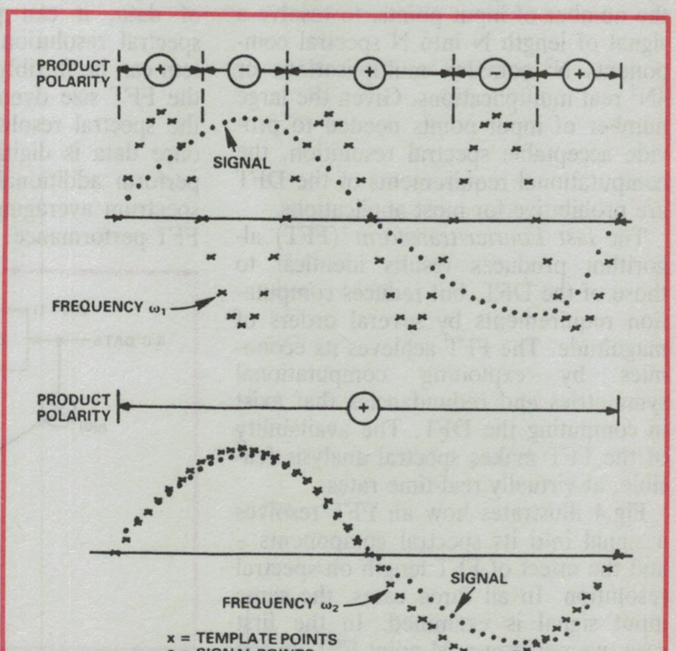


Fig.3: Comparing a signal with a sinusoidal 'template', to measure its frequency content. When the template signal is at nearly the same frequency, the polarity of the product is continuously positive.

but what is most interesting from the standpoint of DSP is that the DFT technique provides us with a means to estimate spectral content by digitising an incoming signal and simply performing a series of multiply/accumulate operations. And this gives us an accurate picture provided that the signal is sampled frequently enough (at a rate greater than its highest frequency component), and assuming that the signal is periodic.

To see qualitatively why the DFT technique yields spectral information, consider Fig.3. A time signal is superimposed on a spectral 'template' at various frequencies.

In the first case, for frequency ω_1, the input signal and the spectral template have little relationship; as a result, the positive products are more or less cancelled out by negative products. The net effect is that the summation of the product between the two in the DFT process indicates little spectral energy of frequency ω_1 in the input signal.

In the second case, however, a reinforcing pattern emerges; the signal and the template tend to be positive or negative concurrently – producing a positive product nearly everywhere. Thus, the sum of the products will be a large positive number, indicating that the incoming signal has significant energy of frequency, ω_2.

Unfortunately, the large number of multiplications required by the DFT process limits its use in real-time signal processing. The computational complexity of the DFT grows with the square of the number of input points; to resolve a signal of length N into N spectral components N^2 complex multiplications or $4N^2$ real multiplications. Given the large number of input points needed to provide acceptable spectral resolution, the computational requirements of the DFT are prohibitive for most applications.

The *fast Fourier-transform* (FFT) algorithm produces results identical to those of the DFT, but reduces computation requirements by several orders of magnitude. The FFT achieves its economies by exploiting computational symmetries and redundancies that exist in computing the DFT. The availability of the FFT makes spectral analysis feasible, at virtually real-time rates.

Fig.4 illustrates how an FFT resolves a signal into its spectral components – and the effect of FFT length on spectral resolution. In all three cases, the same input signal is examined. In the first case, we perform a 64-point FFT on the first 64 sample points; the second and third cases perform 256 and 1024-pointFFTs on the first 256 (1024) data points. The differences observed in spectral resolution underscore a key principle – the longer the time period in which a signal is observed, the sharper the spectral resolution that can be attained.

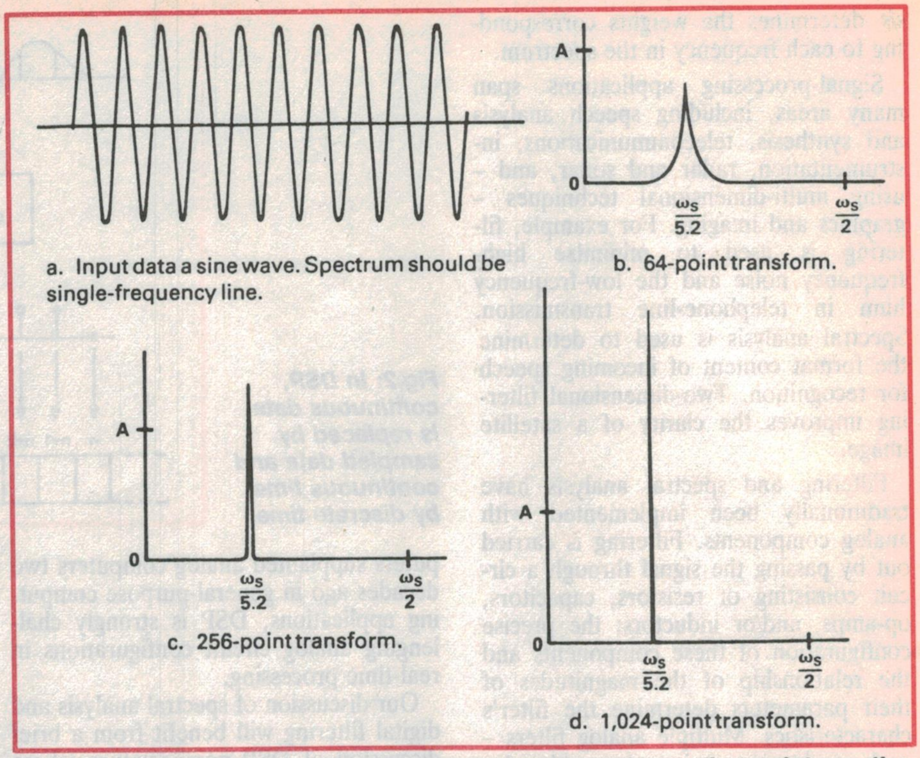


Fig.4: Fourier transform – the effect of the number of sampling points on the computed spectrum.

Bringing digital hardware to bear on a spectral-analysis task has numerous advantages. With a long-enough window of data, it can provide very precise spectral resolution. Moreover, the system can be flexibly programmed to vary the FFT size dynamically, according to the spectral resolution needed. Finally, once data is digitised, it is possible to perform additional DSP tasks, such as spectrum averaging, to further improve FFT performance.

Digital filtering

Digital filters have performance attributes similar to those of analog filters – ripple in the passband and attenuation in the stopband. What distinguishes digital filters is their ability to provide arbitrarily high performance. For example, the rolloff slope (i.e., the rate at which the filter makes a transition from the passband to the stopband) can be made virtually as steep as is desired. In general, it is straightforward to design a digital filter that easily out-performs the most complicated analog designs.

Without going into the maths involved, digital filtering essentially involves *convolving*, or adding together on a continuous basis, the products of various signal samples (representing the signal at different points in time), and

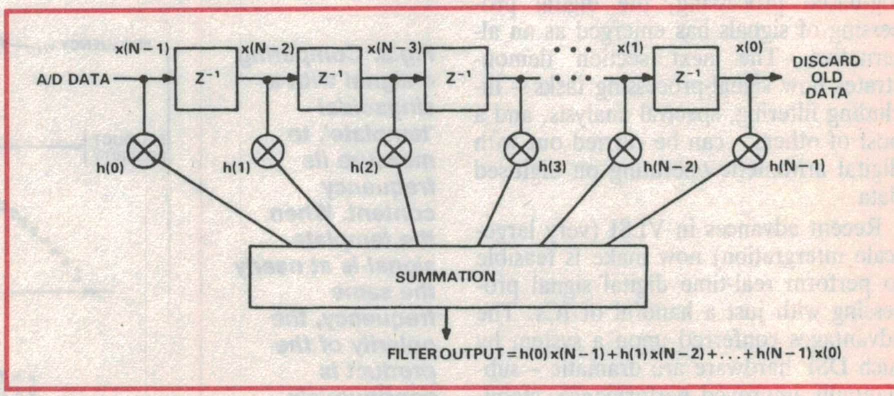


Fig.5: A finite impulse-response (FIR) filter visualised as a tapped delay line.

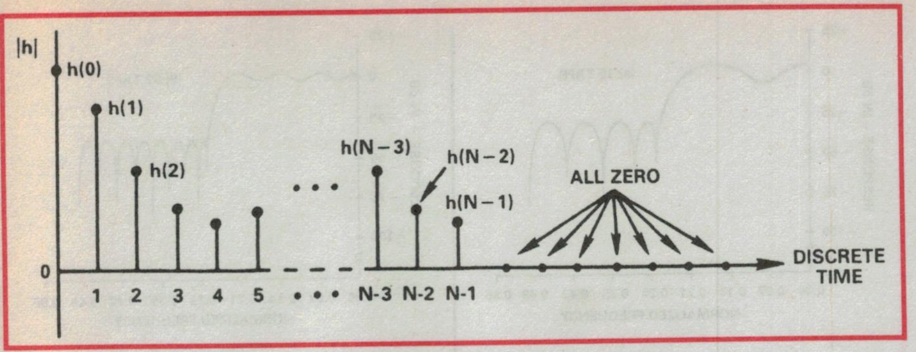


Fig.6: The impulse response h(i) of an FIR filter as in Fig.5.

suitably selected weighting factors.

In the case of a *Finite Impulse Response* or 'FIR' filter, these signal samples are all taken from *before* the filter itself. In contrast, for an *Infinite Impulse Response* or 'IIR' filter, additional samples are taken from the *output* of the filter - so-called 'feedback' terms.

FIR filters

An FIR filter can be viewed as a tapped delay line (see Fig.5); the parameter, N, corresponds to the number of taps of the FIR filter. The number of taps tells us the number of multiply/accumulate operations required to compute this convolution.

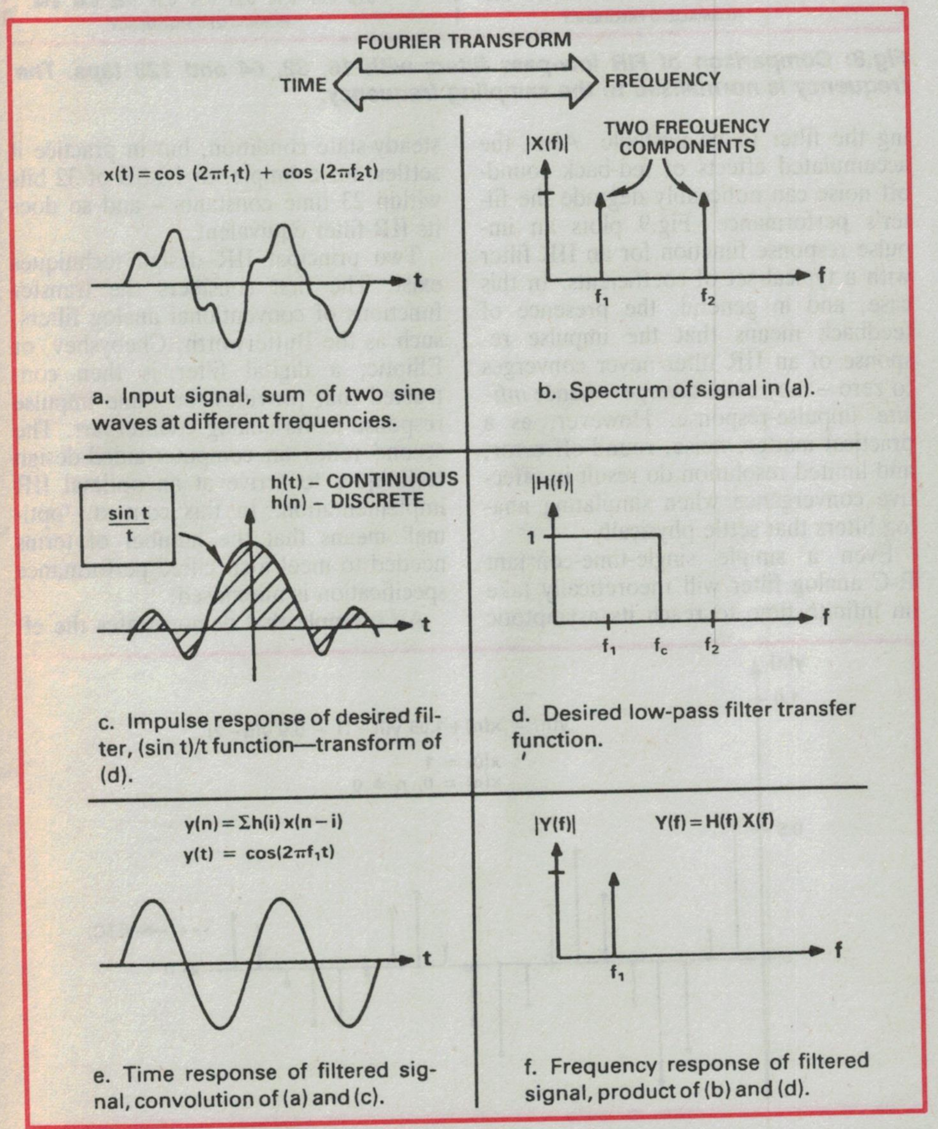


Fig.7: The basics of low-pass filter design using the Discrete Fourier Transform (DFT). The time domain is at left, frequency at right.

The coefficients, h(i), represent the impulse response of the FIR filter. As Fig.6 demonstrates, an input of 1 at time 0 ($x(0) = 1$), and zero at all other times, results in output values equal to h(i) for the periods $i = 0,\ldots,\#N - 1$. Note that the h(i) can be non-zero for only a finite number of time periods, hence the term 'finite' impulse response.

Since they use no feedback, FIR filters are unconditionally stable.

FIR filters can best be understood in the context of two fundamental relationships. First, a filter's time-domain impulse response, h(i), and its frequency response, H(f), are related via the Fourier transform. Second (a key principle of DSP), multiplication in one domain is equivalent to convolution in the conjugate domain.

With respect to FIR filters, this tells us that multiplying the input spectrum by the desired filter transfer function is equivalent to convolving the input time-function with the filter's impulse response in the time domain.

To further amplify the above point, consider Fig.7, where (a) illustrates an incoming signal that we wish to low-pass filter. It consists of the sum of two signals at frequencies, f_1 and f_2. Since there are just two frequencies present, its spectrum looks like (b). We'd like to design an FIR filter to filter out f_2, leaving just f_1, as shown vs. time in (e) and frequency in (f).

An ideal low-pass filter is suggested in (d); note that multiplying it by the input spectrum in (b) will give the spectral domain representation of a low-pass filtered output, allowing f_1 to pass and completely attenuating f_2. Now, the Fourier transform of (d)'s ideal filter is the *sinc* function (sin x/x) in (c). Consequently, if the input (a) is convolved with a discretised sinc function, (c), we can directly compute the filtered ouput signal, as a function of time (e).

More generally, an FIR filter boils down to simply convolving the digitised input signal with the filter's time-domain coefficients, h (i) – an action equivalent to multiplying the frequency representation of the input signal by the filter's transfer function.

Unfortunately, from the perspective of practical implementation, Fig.7(c)'s sinc function is infinite in duration. To obtain a filter that can be implemented, we must somehow truncate the number of coefficients used to represent (c); this can be carried out by discarding the tails – or, more effectively – by multiplying the function by some window. This truncation/windowing, however, makes it impossible to realize (d)'s ideal low-pass filter transfer function, and ripple and rolloff are necessarily introduced (see Fig.9).

By taking an adequate number of taps and properly choosing the coefficients, an FIR filter can provide excellent discrimination, as the response spectra shown in Fig.8 illustrate, for various numbers of taps. In general, the greater the number of taps used in an FIR filter, the better the filter's performance, at the expense of reduced throughput.

In designing FIR filters, tradeoffs must be made among several attributes (i.e., ripple in the passband, ripple in the stopband, width of the transition band, phase distortion, and throughput). These tradeoffs are reflected in the number of coefficients used – and their particular values. This selection can be made directly in the time domain (for example, to implement a pure time delay or an N-point running average), but more commonly is made employing powerful and easy-to-use computer-aided-design (CAD) techniques to determine optimal parameter values for the desired filter performance.

The highest-performance filters of Fig.8 could not be matched by an analog-based implementation. Moreover, these digital filters are straightforward to design and implement in hardware.

There are other advantages of digital FIR filters that further increase their desirability. Once designed, they are stable; performance is insensitive to the effects of temperature or aging. In addition, a key consideration is that the filter's performance can be changed simply, just by modifying the number of coefficients used and their values. For instance, a simple software modification would shift a filter's performance from (a)'s to (d)'s – with no change in hardware, except that slightly more memory is used.

IIR filters

Infinite Impulse-Response filters are the other commonly used digital filter, differing from FIR filters in one fundamental respect: feedback. Because of feedback, the filter's impulse response can continue long after the initial impulse – indeed, for an infinite duration.

The use of feedback allows an IIR filter to economize in the number of multiplications required to provide a given filter performance. But this efficiency is not without its costs. As in other recursive systems, input perturbations can 'ring' indefinitely – in some cases causing the filter to be unstable. Also, the accumulated effects of fed-back round-off noise can noticeably degrade the filter's performance. Fig.9 plots an impulse response function for an IIR filter with a typical set of coefficients. In this case, and in general, the presence of feedback means that the impulse response of an IIR filter never converges to zero – may even diverge – hence *infinite* impulse-response. However, as a practical matter, noise, round-off error, and limited resolution do result in effective convergence when simulating analog filters that settle physically.

Even a simple single-time-constant R-C analog filter will theoretically take an infinite time to reach its asymptotic steady-state condition, but in practice it settles, for example, to 1 LSB of 32 bits within 23 time constants – and so does its IIR-filter equivalent.

Two principal IIR design techniques exist. The first considers the transfer functions of conventional analog filters, such as the Butterworth, Chebyshev, or Elliptic; a digital filter is then contructed that provides the same impulse response as its analog counterpart. The second relies on computer-aided-design techniques to arrive at an optimal IIR implementation. In this context, 'optimal' means that the number of terms needed to meet a specified performance specification is minimised.

An example that demonstrates the ef-

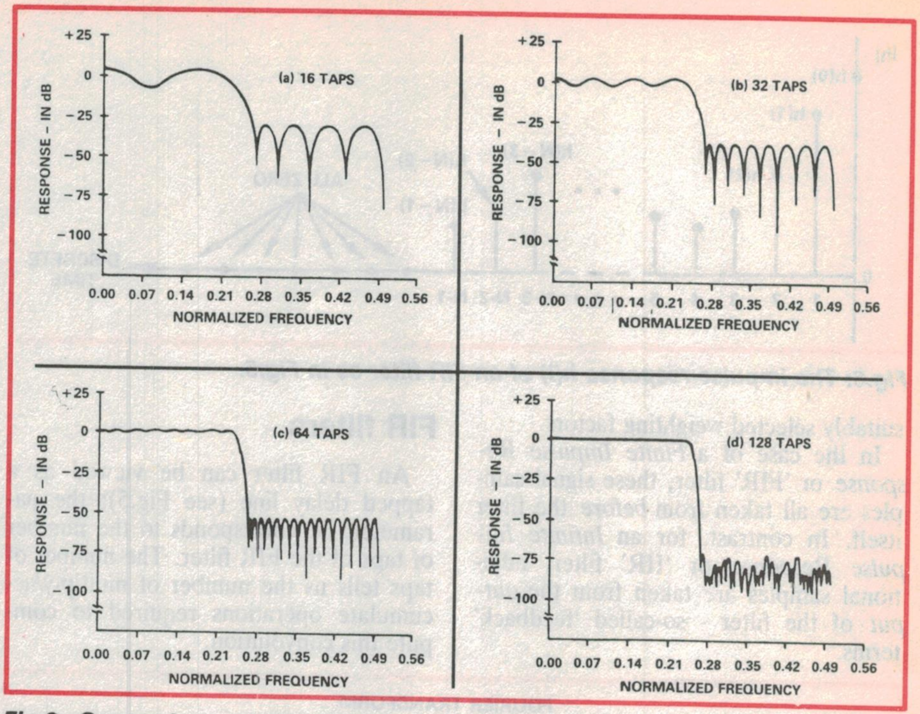


Fig.8: Comparison of FIR low-pass filters with 16, 32, 64 and 128 taps. The frequency is normalised to the sampling frequency.

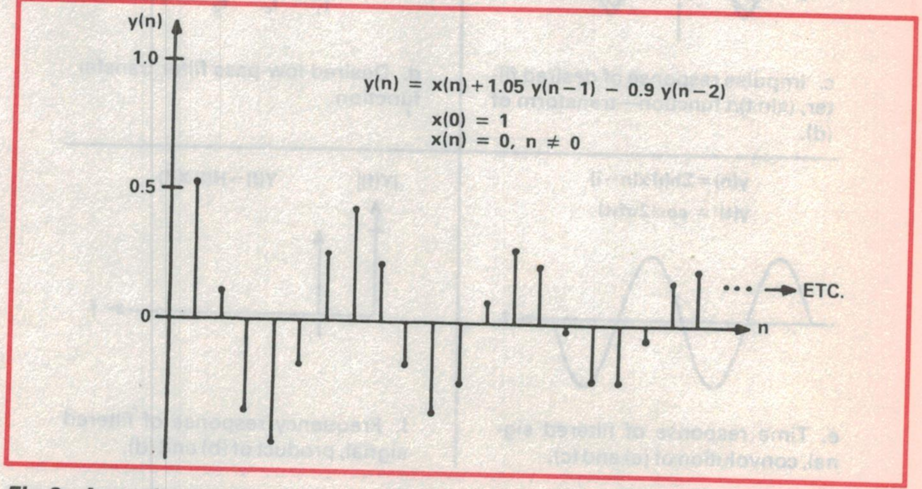


Fig.9: A portion of the impulse response of an infinite impulse-response (IIR) filter.

ficiency of an IIR implementation is a comparison of an FIR and IIR implementation of a 70dB stopband attenuation filter. To achieve this performance, an FIR filter would require nearly three times as many multiplications-per-second as an IIR implementation. These performance advantages, however, require tradeoffs to be made in other key respects, as summarised in Table 1.

Other algorithms

DSP is not limited to FFTs and digital filters. In fact, one of the prime advantages of DSP is that, once the data is digitised, fast digital hardware can perform a broad range of tasks.

Commonly used DSP routines include modulation/demodulation (heterodyning), waveform generation, correlation, estimation, control, power spectrum calculations, and multi-dimensional transforms.

While a discussion of these areas would take us far afield of this article's focus, their breadth points to an important advantage of DSP – system flexibility. By converting signals early and incorporating fast multiply/accumulate hardware to perform digital filtering and/or spectral analysis, a system can readily offer numerous enhancements.

DSP applications

DSP began as a specialised technology used in military applications. With US government funding, a high-speed integrated-circuit array multiplier was developed and first offered commercially in 1976. This multiplier formed the heart of high-performance radar, sonar, and missile-control systems.

Over time, however, the use of DSP has spread from specialised military niches into a broader set of industrial and commercial markets, as Table 2 confirms. Two important uses – modems and studio recording – are discussed below, primarily to illustrate how DSP's advantages benefit the application.

A tremendous amount of information is transmitted today over analog communication links, such as telephone lines. With the growing role of computer-based systems, this information is increasingly digital in nature (for example, digital data and digitised voice transmission). The challenge of transmitting digital data over analog links at high speeds, and reconstructing the received data with high noise immunity, thereby reducing communication costs, is met by a modulator-demodulator (modem).

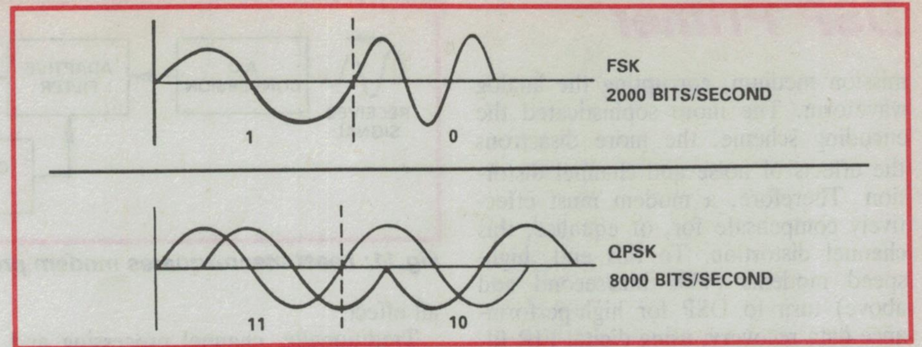


Fig.10: Frequency-shift keying (FSK) and quadrature phase-shift keying (QPSK).

In transmitting digital data over analog communication lines, a digital bit pattern is represented by modulating the phase, frequency, and/or amplitude of an analog signal. Fig.10 shows a simple scheme, which involves changing the frequency of the signal to denote a '0' or '1'; this frequency-shift keying (FSK) method can send 2000 bits/second over a telephone line.

A more sophisticated encoding method, quadrature phase-shift keying (QPSK), modifies the phase of the signal and is capable of transmitting data at four times the rate of simpler methods.

When a modem is sending information, it encodes the digital data into the corresponding analog waveform; in receiving mode, it decodes the waveform and determines the bit pattern that was transmitted. This latter mode is the more difficult to implement.

If the transmission medium were noiseless, a modem's tasks would be limited to simple encoding and decoding – a relatively straightforward exercise. However, a phone line is a noisy trans-

	IIR	FIR
Performance/Throughput	Higher	
Ease of Design		Easier
Filter Stability	Sensitive	Unconditional
Round-off Noise	Sensitive	Insensitive

TABLE 1: Comparing FIR and IIR filter characteristics.

PRINCIPAL DSP MARKETS

Instrumentation:	Spectrum analyzers, vibration analyzers, mass spectroscopy, chromatography
Audio:	Studio recording, music synthesis, speech recognition
Communications:	Modems, transmultiplexers, vocoders, satellite transmission, repeaters, voice storage and forwarding systems
Computers & Computer Peripherals:	Arithmetic acceleration, servo controls for disk head positioning, array processors, engineering workstations
Imaging:	Medical, satellite, seismic, bandwidth compression, digital television, machine vision
Graphics:	CAD/CAM, computer animation and special effects, solids modelling, video games, flight simulators
Defense Electronics:	Radar, sonar, missile/torpedo control, secure communications
Control:	Robotics, servo links, skid-eliminator adaptive control, engine control.

TABLE 2: Applications developed to date for DSP.

mission medium, corrupting the analog waveform. The more sophisticated the encoding scheme, the more disastrous the effects of noise and channel distortion. Therefore, a modem must effectively compensate for, or equalise, this channel distortion. To this end, high-speed modems (4800 bits/second and above) turn to DSP for high-performance data recovery, using digital FIR filters.

An additional complexity of telephone-line transmission is that its distortion properties change over time. Therefore, a modem's digital filter must be able to adapt to changes in the environment. This need to respond to a changing environment underscores another advantage of DSP – a digital filter's characteristics can be modified simply by changing its coefficients. Coefficient updating in a modem is determined by the observed drift of a property of the distortion in the system.

Aided by DSP, then, a modem can make it possible for high-speed data transmission to be implemented effectively. As Fig.11 illustrates, the DSP is the heart of a high-speed modem. The processing required generally can be handled by one digital multiplier, surrounded by the appropriate support devices (a program sequencer, an address generator). Alternatively, depending on the requirements of the modem, a single-chip processor may adequately handle all DSP requirements.

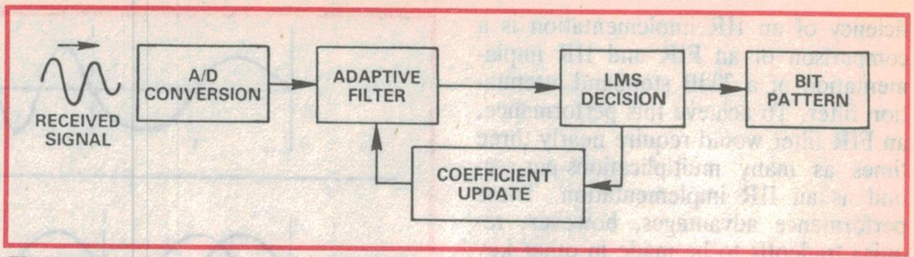


Fig.11: Least-mean-squares modem processing architecture.

Studio recording

One of the most interesting applications of DSP is emerging in the audio processing performed in recording studios. This processing starts after the initial recording of voices and instruments in the studio; after a large number of steps, it ends with the recorded version that reaches the home stereo. Increasingly, DSP is being used to handle all intermediate steps.

The flow of activities in studio recording is complex and varied. Generally, multiple channels are used, with each track dedicated to one or more sources (instruments/voices). All channels need not be recorded at the same time. Each channel is subjected to extensive processing, including gain control, filtering, non-linear compression or expansion, reverberation adjustments, spectral equalisation, and special-effects enhancements. The contributing channels are then mixed together to obtain a final arrangement with the desired overall effect.

Traditionally, channel processing and mixing were implemented entirely in the analog realm – with numerous disadvantages. Each channel's information – stored as an analog signal on magnetic tape – degrades as the cutting, splicing, and re-recording process progresses, undermining the benefits of the processing. The limited performance range available with analog processing sets a ceiling on the signal enhancement that can be attained. Also, analog circuitry can only handle one channel at a time; multi-channel mixers are expensive and difficult to control. Finally, if analog processing hardware is used, overall mixing flexibility can be achieved only through hardware modifications. In practice, this means that the mixing process loses its ability to creatively explore special effects.

Increasingly, audio processing is relying on digital techniques to improve audio quality. The first step in this transition was digital recording, which became prevalent about seven years ago. Audio signals are first converted to digital form before being stored on magnetic tape. Digital recording eliminates several sources of degradation that hamper analog recordings, including the effects of non-linearities and additive noise in the magnetic materials used for recording, and wow and flutter in the tape playback mechanism.

In studio mixing applications, however, digital recording does not eliminate all complications. In the mixing and enhancement process, information is passed from one tape to another – requiring D/A and A/D conversion processes, a source of noise. These conversions are no longer necessary if all processing and mixing are handled with DSP techniques.

In DSP-based studio recording systems (see Fig.12), signals are converted to digital as early as possible. In fact, some implementations place a remotely controlled amplifier/converter at the recording microphone. After conversion, the audio processing is handled digitally, with high performance and flexibility. Gain factors are handled with digital multiplication. Filtering and equalisation can be handled with an IIR filter that replicates the performance of standard analog filters. Alternatively, digital FIR filters can provide high-performance linear-phase filters or complex comb filters. Dynamic-range control is easily included in the system by using a multiplier for non-linear compression-/expansion computations.

The traditional mixing process is also easily implemented in a DSP-based system. Digital channels to be mixed are simply added together. Relative time delay lags can be easily introduced into the channel flows, allowing phase coherence to be explored without adding ex-

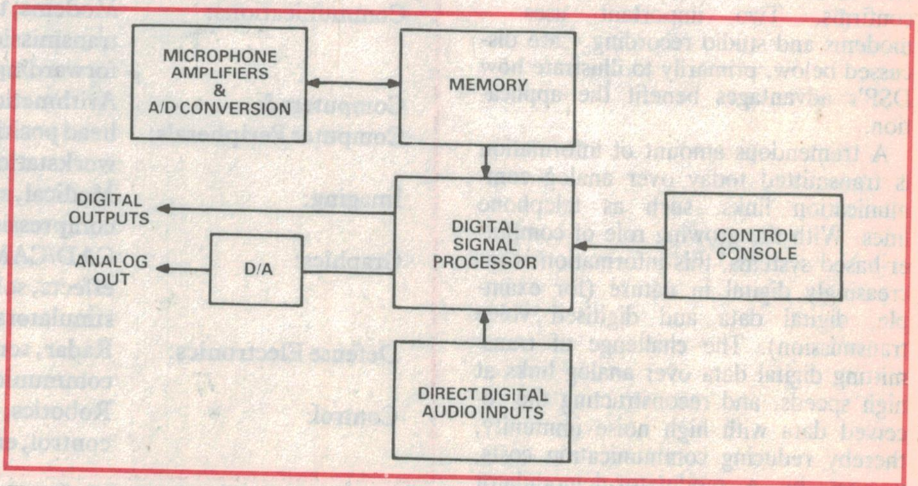


Fig.12: Block diagram of DSP applied to studio recording.

pensive delay lines to the system. An additional advantage is that the channel interconnections – which have to be hardwired in an analog processor – can be easily reconfigured in a DSP system.

In addition to improving on traditional operations, a DSP studio recording system opens up numerous new options. Unusual special effects are readily included in the system. Reverberation effects can be modelled, simulated, and integrated into the final recording. An FFT routine's spectral analysis of the signal forms the basis for frequency-domain filters that provide optimal equalisation. Overall system flexibility allows the entire mixing system to be dynamically configured – processing steps can be re-ordered, mix groups and subgroups re-specified, and effects such as fading, equalisation, and compression/expansion included at any juncture.

Studio recording, then, follows the pattern of other applications using DSP. DSP techniques offer increased precision for processing steps traditionally performed with analog circuits. Of equal importance, DSP's flexibility paves the way for many new and creative processing steps. As in other areas, the DSP is shifting the role of converters; accurate ADCs and DACs are used in the system, but as close to the real-world interface as possible. The signal processing is conducted in the digital realm.

(Adapted from 'Analog-Digital Conversion Handbook', by permission of Analog Devices, Inc.).

Glossary of DSP terms

Accumulator An arithmetic element that adds together, or accumulates, a sequence of inputs. A DSP multiplier with an accumulator on-ship is called a multiplier/accumulator (MAC).

Algorithm A DSP algorithm, such as the fast Fourier transform, or a finite impulse-response filter, is a structured set of instructions, and/or operations, tailored to accomplish a signal-processing task. Each algorithm has a well-defined structure; however, variations in algorithm parameters, such as the number of input points or taps, allow the same basic algorithm to perform different functions.

ALU An arithmetic and logic unit, which performs additions, subtractions, or logical operations (e.g., AND, OR, XOR) on operand pairs.

Attenuation The damping-out, or suppression, of signal content. Filters will attenuate the frequency content of a signal that lies in the filter's stopband.

Barrel Shifter A device that accepts a digital number as its input and – as a function of the controls – shifts the number up or down, or rotates the word as though it were placed on a barrel. A barrel shifter is used in a system for many tasks, including scaling and normalisation.

Biquad A particularly simple recursive, or infinite impulse-response (IIR), digital filter form, often used as a building block for constructing more complicated recursive filters. A biquadratic, or biquad, section uses the three most recent input points and the two most recent output, or feedback, values to compute each output point.

Block Floating Point A compromise between fixed-point and floating-point arithmetic. Data grouped in "blocks" is assumed to be normalised with a common exponent (but, not being attached to the data words, the exponent need not be explicitly processed with the data). In essence, the process is carried out in fixed point, with its inherent speed advantage.

Convolution In discrete computations, a mathematical operation, defined as the summation, or integral, of a product of two functions over a range of differences in the independent variable. In the time domain, one function is the impulse response, as a set of coefficients, h(i), over N time intervals; the other is the input, f(n – i), as a function of the differences between the time at the instant at which the function is being evaluated, n, and the input at earlier instants, determined by the variable delay, i, from 0 to N. In DSP, the convolution of an input signal, x, with the coefficients, h, results in the filtering of the input signal.

Correlation A mathematical operation that indicates the degree to which two signals overlap. A high positive correlation reflects two signals that closely track each other. A negative correlation indicates that the two signals are closely related, but out of phase by roughly 180°. If the correlation is close to zero, the two signals are unrelated.

Digital Signal Processing DSP is a technology for high-performance signal processing that combines algorithms and fast number-crunching digital hardware.

Discrete Fourier Transform The discrete Fourier transform (DFT) is a DSP algorithm used to determine the Fourier coefficient corresponding to a particular frequency.

FFT An n-point fast Fourier transform (FFT) is computationally equivalent to performing n DFT's but, by taking advantage of computational symmetries and redundancies, can reduce the computational burden by several orders of magnitude.

FIR Filter A finite impulse-response (FIR) filter is a commonly used type of digital filter. Digitised samples of the signal serve as inputs; each filtered output is computed from a weighted average of a finite number of previous inputs.

Fixed-Point Arithmetic Each number is represented in a fixed arithmetic field of n bits, allowing integers in the range 0 to $2^n - 1$, to be represented.

Floating-Point Arithmetic Each number consists of a mantissa and an exponent, allowing wide dynamic range to be accommodated in the numbering system.

IIR Filter An infinite impulse-response (IIR) filter is a commonly used type of digital filter. This recursive structure accepts as inputs digitised samples of the signal; each output point is computed on the basis of a weighted average of past output – or feedback – terms as well as past input values. An IIR filter is more efficient than its FIR counterpart, but poses more challenging design issues.

MAC Multiplier/accumulator; see Accumulator.

Microcode A set of instruction control signals in a program memory that govern the cycle-by-cycle operation of the various devices in a building-block architecture.

Passband The frequency range over which a filter passes, to within some tolerance, the incoming signal content.

Pipeline An architectural structure that allows two or more operations to be carried out simultaneously, like the stages of an assembly line. While each basic operation requires several cycles to complete, a later stage of one operation is simultaneous with an earlier stage of another operation. This structure allows the effective throughput rate for each operation to be substantially increased.

Rolloff A measure of filter performance defined as the rate-of-change of the filter's amplitude response with respect to frequency over a transition band.

Stopband The frequency range over which a filter attenuates, to within some tolerance, the incoming signal content.