Introductions

The reader is already an expert in signal processing, although possibly unaware of it. We are all remarkably complex signal processing systems, adaptively processing intricate audio and video signals every moment of our lives. While awake we input intricate signals from our environment, extract highlevel representations of information carried by these signals, make decisions based on this information, record some of the information for later recall and processing, and produce new signals to change our environment in *real* time. Even while sleeping, although most of the input has been removed, we unconsciously continue the processing off-line; we reintroduce recently input signals in order to correlate them with previously stored signals, decide which signals should be stored for long periods of time, and generally perfect our signal processing performance. Due to this signal processing we are extremely good at understanding speech and immediately reacting based on what has been said. We scarcely think about our ability to recognize faces and greet (or avoid) their owners. We take our proficiency at reading handwriting for granted, except when admiring the pharmacist's even greater competency when presented with a physician's scrawl.

It is therefore extremely frustrating to discover how difficult it is to design artificial devices that can perform as well. After decades of research, devices that can understand unconstrained human speech are still extremely primitive, and even speech synthesis is still to be considered a nontrivial problem. Machine recognition of human faces is possible only in severely restricted environments, and even our limited capabilities are not yet commonplace due to being prohibitively expensive. While optical character recognition of high quality printed fonts has been perfected, acceptable machine reading of handwriting has yet to be attained.

These three examples—speech understanding, face recognition, and reading of handwriting—are typical of a long list of tasks which we find almost trivial, but which have turned out to be extremely difficult for machines. It is only due to our meager attempts at designing machines to perform

these functions that we have come to grasp their extreme complexity. Due to this inherent complexity, researchers and implementors attempting to mechanize these functions have turned to the strongest and most intelligent devices available. The most sophisticated and capable invention humankind has devised to date is the digital computer. For this reason it is natural that much of the state-of-the-art signal processing is performed digitally. In this, the first chapter, we introduce *signal processing*, and more specifically Digital Signal Processing, which from now on we shall call DSP.

In order to acquaint the reader with the concept of using digital technology in order to process signals, we will first trace the early history of signal processing. We then jump ahead to a survey of state-of-the-art applications, in order to convince the reader that the problem is still alive and interesting. Next we introduce the concept of signal processing by demonstrating analog signal processing on a simple example. Finally we present the basic ideas behind the use of computers in signal processing.

1.1 Prehistory of DSP

The first major accomplishments of humankind involved mastering the processing of *material* objects, and indeed humankind is often defined as the animal who fashions tools. Lower forms of animals do not, in general, change naturally occurring objects in order to adapt them to their needs. When humankind discovered that one could take stones and bones and by relatively simple processing convert them into arrows, knives, needles, fire-making implements, and the like, the species transcended all those that had come before it. More and more complex processing algorithms were then developed. For example, humans learned to till the soil, plant wheat seeds, water and fertilize them, harvest the wheat, separate the chaff from the grain, ground the grain into flour, mix the flour with water and yeast, and bake the dough to make bread. This represents a highly developed culture of material object processing.

The next stage in humankind's development involved the processing of *signals*. Signals, like materials, are real physical objects, but are intangible. Humankind learned to adapt a mouth (originally developed for eating) and an ear (designed for hearing predators approach) into a highly flexible acoustic communications system. The medium for this communications exchange was pressure waves in air, and to some extent visual clues conveyed via light. Primitive peoples also developed techniques for communications over distances, such as tom-tom drums and smoke signals. Then came the de-

velopment of the telegraph and telephone, which used the electrical signals, and radio, which used electromagnetic waves. The objects being manipulated remain physically existing quantities, although they became less and less tangible.

The final stage (so far) in humankind's development entailed learning to process *information*. Unlike material objects and signals, information is entirely abstract and cannot really be said to exist in the physical world. Information is like ideas, and while it can be quantified it is not held back by physical limitations. The seeds of information-processing were sown with the invention of writing and arithmetic, philosophy and algebra, art and logic, but were brought to full fruition with the invention of the digital computer. The computer can transcend nature by predicting physical phenomena before they occur, simulating worlds that cannot exist, and creating new information where none was before.

The marriage of the last two developments in mankind's history, i.e., utilizing digital computation for the purpose of processing of signals in the real world, is the objective of DSP. While perhaps not a major milestone in the history of humankind, DSP is a significant enough endeavor to warrant study by all interested in manipulating their world using information-processing techniques.

- 1.1.1 Does *Digital Signal Processing* mean 'the processing of digital signals' or 'the digital processing of signals'?
- 1.1.2 What possible relationships might there be between DSP and the following *computer science* fields?
 - 1. Numerical Analysis
 - 2. Compiler Design
 - 3. Operating Systems
 - 4. Database Programming
 - 5. Artificial Intelligence
- 1.1.3 Listen to an extremely weak station on an AM radio. Can you understand what is being said? Would you be able to understand were the language spoken to be one in which you are not completely proficient? What happens if there are interference and whistles? Other radio stations? Does the same happen with an FM radio station? Repeat the above experiment with a shortwave radio. Find stations using SSB modulation. What happens if you do not tune the signal in properly? Sit in a cocktail party where many groups of people are talking. Focus on one conversation after another. How well can you separate out voices? What have you learned from this exercise?

1.2 Some Applications of Signal Processing

So what exactly *is* signal processing and why do we want to do it? Signal processing is the discipline of detecting, manipulating, and extracting information from physical quantities that vary in time (*signals*).

The only way to really understand what we mean by this definition is to consider examples of signal processing applications.

Voice communications, processing, and store-and-forward. The main means of communications between humans is speech. One human broadcasts information as an acoustic signal that can be detected by other humans. When the persons desiring to converse are not colocated, we must provide a mechanism to transfer the signal from place to place. When they are not available simultaneously, we need to record this acoustic signal for later playback. Digital forwarding and recording of speech have certain advantages, as we shall discuss later. In order to use digital transfer and storage we require a method for making a digital representation of the acoustic signal, as well as algorithms for Automatic Gain Control (AGC), Voice Activity Detection (VAD), and perhaps compressing the digital representation in order to preserve disk space or communications bandwidth. Additional processing entails enhancing the quality of speech in noise, and acceleration/deceleration of the playback speed without distortion. More complex processing algorithms are required for separation of one voice from others (cocktail-party effect), machine-synthesized speech (text to speech), speech recognition and speaker identification.

Music synthesis, recording, and playback. Much of what we said about speech holds for 'wider bandwidth' acoustic signals, such as music. Here the emphasis is on high-quality transfer (e.g., broadcast), compression (e.g., MPEG files), storage (e.g., compact disks), and noise reduction, (for example, restoration of old recordings). However, there are also processes specific to music such as accurate recreation of the original sound in different acoustic environments (equalization), digital simulation of musical instruments (synthesizers, keyboard organs, MIDI), and special effects (mixing, echo, reverberation).

Data communications on voice-grade channels. Another extension of voice processing is the adding of data bearing signals to channels originally designed for voice use. *Touch-tone* dialing (technically known as DTMF) has become almost universal for dialing and for menu selection. Facsimile machines that transmit documents over public telephone circuitry have also become commonplace, and high speed modems enable computers to interconnect over this medium. It is also useful to convert the audio itself to digital form, for example, to enable several conversations to share one telephone line, or for the purposes of secure communications (encryption).

Automobile Industry. A conventional muffler reduces noise by passing exhaust gases through a series of baffles that reduce their velocity. Unfortunately this same process requires the engine to waste energy forcing these gases through the muffler, energy that would otherwise have been used to increase horsepower and fuel efficiency. The electronic muffler uses *active noise cancellation* instead; the noise is sensed by a microphone, and identical noise is added 180° out of phase. This same technique can be utilized to add out-of-phase vibration to the mounts of the engine on the chassis. Acoustic DSP can also be used to diagnose and control engine faults.

Industrial Applications. The measurement of vibrational modes and the discovery of their underlying causes and mechanical structural problems they may indicate is a well-known industrial application of signal processing. Chemical process control relies heavily on instrumentation that employs advanced signal processing. Robots on assembly lines receive signals from sensors and adaptively act upon them by moving their mechanical appendages. Other applications include the diagnosis of electric motor faults from current signatures, the rapid and precise measurement of fluid flow, the control of welding and smelting apparatus, and pump wear monitoring.

Biomedical engineering. The human brain is a massively parallel computer containing about 10¹⁰ processing units called *neurons*. These neurons fire electric impulses that are not externally observable, but by placing electrodes at various positions on the scalp, voltages that represent sums of many neurons are detectable. These recordings are known as electroencephalograms (EEG) and after proper processing they can be used for diagnosis of sleep disorders, epilepsy, and brain disease. The electric activity of the heart can also be monitored, using the electrocardiogram (ECG). Processing this signal aids the physician in diagnosing heart problems. Monitoring during labor involves continual display of fetal heart rate as well as uterine muscular activity. These signals require removal of hum introduced from the electric power source and extensive real-time preprocessing.

Radar and sonar processing. The purpose of radar and sonar is to locate bodies in space and optionally to determine their speeds. Well-known radar applications include air traffic control, aircraft radar, smart-missiles, weather satellite radar, and police speed traps. The distance determination relies on the sensitive detection and accurate timing of return signals; electromagnetic signals for radar and acoustic signals in water for sonar. This processing relies on *matched filtering* and high resolution spectral analysis. Doppler radar speed measurement requires precise frequency measurement. 6

Radar signals usually have very high bandwidths, and consequently require very fast processing rates. Sonar bandwidths are much lower than those of radar, but the processing power required is high due to the interference being stronger, and the return signals being weaker and more distorted. Multipath reception complicates the location effort and often arrays of sensors are employed and *beamforming* used. Electronic intelligence (ELINT) and electronic warfare (EW) exploit interception of radar signals in order to detect/identify and to deceive/defeat the radar system, respectively.

Seismology. Seismic signal analysis is used by the oil and gas industries in the exploration of subsurface hydrocarbon reserves; by government agencies for nuclear detonation detection; and by long-term planning authorities for investigation of subsurface geological formations and their significance to architecture and urban development. Signals passively collected during naturally occurring seismic events such as earthquakes and volcanic eruptions may aid in their detection, epicenter location, and prediction. During active exploration such seismic disturbances must be initiated, for example, by setting off high-energy charges (although environmental considerations may mandate the use of lower energy sources such as acoustic speakers). The seismic waves are scattered by interfaces between different geological strata, and collected at the earth's surface by an array of seismometers. Thus multiple seismic signals must be digitized and processed to lead to source location or mapping of the geological strata.

- 1.2.1 What other areas utilize signal processing? List several applications not on the above list. Research at a library or search the Internet.
- 1.2.2 What areas may potentially benefit from signal processing, but are not yet using it? Write up a detailed description and submit to the patent office.
- 1.2.3 Consider a mobile robot able only to avoid obstacles, and to locate an electric outlet when its batteries are low. What technologies would be needed to implement such a robot? Where is DSP needed? Now give the robot the ability to receive verbal commands, to retrieve objects, and to keep its owner informed. What DSP is needed now?
- 1.2.4 Dual Tone Multi Frequency (DTMF) tones consists of two frequencies. The same is true for *dial tone* and *ring-back tone*. What simple tunes can you play recognizably using DTMF? Who answers the phone when you do? Why are two different frequencies used—wouldn't it be easier to use only one? (Hint: Take the expense of incorrectly routed calls into account.)

- 1.2.5 Using a computer with multimedia capabilities, record some naturally spoken speech and observe its waveform graphically. Does this picture contain all the information we can obtain from listening to the speech? You can easily find long silences and tell the difference between whispering and screaming. Try to tell where the words begin and end. Can you differentiate between male and female voices? Can you guess what is being said? Assuming you answered in the affirmative to the first question, where exactly is the information?
- 1.2.6 You are given the job of saving the several megabytes of information from an old computer, about to be discarded. The computer has no serial output ports or modem, but *does* have an analog output that can produce 256 different voltage levels. What is the simplest encoding method for outputting information? How can you decode and store the information (you can use any readily available computer or peripheral)? How fast can you go? Do you think it can be decoded this fast? What are the real limitations? What happens if background noise is recorded along with the signal? This is the basic idea behind the download path of the so-called *PCM modem* that achieves 56 Kb/s over telephone lines.
- 1.2.7 Same problem but this time the computer has an internal speaker and can generate tones of different frequencies (all of the same amplitude). You may decide to convert the data to be saved, byte by byte, into one of 256 tones, and to record the tones onto an audio cassette. Design a transmitter (modulator) for this case (try writing a program). What do you need to decode this information (demodulator)? How fast can you go? Perhaps you decide to convert the data to be saved, bit by bit, into one of only two tones. What do the modulator and demodulator look like now? This is the FSK modem, capable of 300 b/s on phone lines.

1.3 Analog Signal Processing

Signal processing is the discipline of detecting, manipulating and extracting information from physical quantities that vary in time. Now that we have seen *why* we want to do it, we can begin to discuss *how* to do it. DSP processes signals *digitally*, that is, by programming, rather than by building *analog* electronic circuits. However, before we jump into digital processing, a brief discussion of analog processing is in order.

It is clear that signals can be processed using analog circuits such as amplifiers and filters. These devices take analog signals as inputs and return analog signals as outputs. Electronic engineers know how to design these circuits to obtain specific processing characteristics (obtaining certain

voltage levels, amplifying certain frequency ranges while eliminating others, etc.). Quite complex systems can be designed, for example, receivers that are sensitive only to very specific waveforms. In-depth explanation of the techniques that have been developed in this field is beyond the scope of our book, and for our purposes it is sufficient to analyze a simple example.

Assume that wish to input a sine wave of arbitrary frequency, offset and amplitude (within bounds of course) and output a train of narrow pulses of equal frequency. One can observe the input and desired output as 'X' and 'Y' in Figure 1.2. Why would one want to perform this operation? There may be a number of reasons. For example, one may want to measure the frequency of the sine wave using a digital counter that increments upon receiving a narrow pulse. Or one may need the pulse as a synchronization signal for some process that should be locked in time with the sine wave. It could be that we need to generate a pulse for triggering an oscilloscope or some other instrument.



Figure 1.1: Diagram of an analog sine to pulse converter.

One way of producing the desired effect is depicted in Figure 1.1, with the input, intermediate signals, and output drawn in Figure 1.2. The first step is to pass the signal (waveform X) through a *capacitor* that acts as a DC blocker. This ensures that the signal values are centered around zero voltage (waveform A). Next we put the signal through a hard limiter. This is an amplifier driven to its maximum amplification, so that its output will be $+V_{max}$ for any positive input, and $-V_{max}$ for any negative input (waveform B). Next we split the signal so that it traverses two paths, one slightly delayed with respect to the other (this delay determines the width of the pulse to be obtained). The delayed signal is now subtracted from the nondelayed version, producing an output that is almost always zero (waveform C). The subtraction is performed, once again, using an amplifier, this time a differential amplifier that has noninverting and inverting inputs. The amplifier's output is nonzero and positive at the leading edge of the square wave (since



Figure 1.2: Analog signals from the sine to pulse converter.

there we have $+V_{max} - -V_{max}$) and nonzero and negative one half-cycle later. This latter artifact is eliminated by the use of a *half-wave rectifier*, a component that only passes positive voltages, suppressing negative ones. The final result (waveform Y) is a narrow positive pulse locked to the leading edge of the original sine, as required.

- 1.3.1 The above example assumes the existence of a *delay* element, which may be quite difficult to implement. For high-frequency signals, a long piece of cable may be used, relying on the finite speed of propagation of the signal through the cable to introduce the time delay. For low frequencies, even extremely long lengths of cable introduce delays that are insignificant fractions of the period. Assume you have an analog differentiator, a device whose output is the derivative of its input. How would you use it in our sine to pulse converter? What would the output pulse look like?
- 1.3.2 The device we described above is basically a zero crossing detector, a device that determines when the signal goes through zero voltage. We can avoid the need for a rectifier if we employ a *peak picker*, which outputs pulses at the maxima of the input signal. How can a peak picker be implemented given a differential amplifier and a reference voltage source?
- 1.3.3 How can an *integrator* (a device whose output is the integral of its input) be used to solve our problem?

1.3.4 Assume you have a digital representation of the input; that is, a sequence of voltage measurements uniformly spaced in time. Write software routines for the zero crossing detector and the peak picker, assuming that the sine wave is sampled very densely (many equally-spaced samples per cycle). Will your routines work if the sampling rate is much lower, for example, eight samples per cycle? Four samples per cycle?

1.4 Digital Signal Processing

In the previous section we saw an example of how signals can be processed using analog circuits. How can we similarly process analog signals digitally? A very general scheme is depicted in Figure 1.3.



Figure 1.3: Generic DSP scenario.

The purpose of the filters will only become clear later on (see Section 2.10). The blocks marked A/D and D/A represent devices that convert Analog signals into Digital ones, and vice versa. These devices allow us to translate signals in the physical world into sequences of numbers that computers can accept as input and process, and to convert sequences of numbers output by computers back into physical signals.

The heart of the system is the digital signal processor, which we shall usually just call the DSP. (This double use of the acronym DSP should not be confusing, with the differentiation between the *processor* and *processing* being easily understood from context.) You may think of the DSP as basically a computer that performs the needed computation. It may be a general-purpose computer, such as a desktop workstation, readily available and easily programmed. Or it may be special purpose digital hardware designed specifically for the task at hand. Intermediate between these extremes is a general-purpose programmable digital signal processor. These DSP *chips* are similar to microprocessors, with arithmetic capabilities, memory access, input and output ports, etc. However, as we shall discuss in Chapter 17, they are augmented with special commands and architecture extensions in order to make them particularly efficient for computations of the type most prevalent in DSP applications. While programming DSPs in high-level languages is becoming popular, their special architectures can be best exploited by low-level (assembly) programming.

At this point you are probably asking yourself whether DSP is truly superior to analog signal processing. Why should we replace a handful of electronic components with two filters, an A/D and a D/A, and an expensive and hard-to-program DSP? The main reasons to favor digital techniques over analog ones, are:

- greater functionality,
- accuracy and reproducibility,
- modularity and flexibility,
- increased price/performance, and
- reduced time-to-market.

The greater functionality derives from the possibility of implementing processes that would be extremely difficult and/or expensive to build in analog circuitry. In particular, arbitrary time delays, noncausal response, linear-phase (see Chapter 6), and adaptivity (see Chapter 10) are simple to implement in DSP, while practically impossible in analog.

Accuracy and reproducibility are characteristics of digital numbers in contrast to analog voltages. Precision is a function of the number of bits used in computation, and digital numbers can be protected against inaccuracy by error-correcting codes. A copy of a copy of a copy of a digital recording is identical to the original, with no added noise and no 'drift' caused by temperature or aging.

The modularity and flexibility are byproducts of programmability; DSP code can readily be reused and modified. DSP code, like all software, can be made generic and placed into libraries with little sacrifice. Last minute changes are a hardware engineer's worst nightmare, while field debugging is commonplace in the software arena.

In recent years significant advances have been achieved in digital technology, including the development of smaller, faster, more power efficient and less expensive digital processors. For these reasons digital technology is finding its way into almost every facet of our lives. Once the process *can* be performed digitally, it usually takes only a very short time until it is profitable to do so.

There are, admittedly, a few drawbacks associated with the use of DSP. The most notable of these are:

- limited speed of general-purpose DSPs,
- finite word-length problems, compounding of round-off errors, and 'stability', as well as
- the need for specialized algorithms and programming.

As a result of the first problem many applications, for example, those dealing with real-time processing of high bandwidth signals, cannot yet be handled digitally. The second shortcoming is more a hindrance than a true impediment, compelling us to analyze our numeric algorithms more cautiously. The third drawback is actually a favorable opportunity for students of DSP. It ensures a steady demand for competent DSP personnel for many years to come.

- 1.4.1 'Some **DSP** practitioners rarely deal with **DSP** theory at all, rather are experts at programming **DSP**s for control and general algorithmic applications, rather than as a **DSP**.' The acronym **DSP** appears four times in this sentence. Explain which of the various meanings (processing, block diagram function, programmable processor) best matches each.
- 1.4.2 Figure 1.3 depicts a situation with exactly one input signal and one output signal. Describe an application with no inputs and one analog output. Two analog inputs and one output. One input and two outputs. Can there be useful applications with no outputs?
- 1.4.3 An *amplifier* increases the magnitude of a signal, while an *attenuator* decreases the magnitude. An *inverter* inverts the polarity of a signal, while a *clipper* limits the magnitude of a signal. A *DC blocker* shifts the average to zero. What mathematical functions are performed by these components? Code a routine for each.
- 1.4.4 What differences do you expect to find between DSPs and conventional CPUs?
- 1.4.5 Are there functions that can be performed in analog electronics, but cannot be performed in DSP?
- 1.4.6 Compare digital Compact Disc (CD) technology with the older Long Playing (LP) records. Explain why CD technology has totally replaced LPs by considering sound quality, playing duration, noise, stereo separation, the effect of aging media, and the ability to make copies.