Cambridge University Press
978-1-107-02097-9 - Digital Waveform Generation
Pete Symons
Excerpt
More information

1 Introduction to waveform generation

Systematic generation of periodic signals with electronically controlled frequency, phase, amplitude and waveform shape (or *waveshape*) is ubiquitous in nearly every electronic system. The sinusoidal local oscillator in a super-heterodyne radio receiver is a simple example of a signal source whose controllable frequency tunes the receiver. Another example is a step input waveform (e.g. a square wave) that allows us to measure the step response of a closed-loop control system (e.g. rise time, fall time, overshoot and settling time) under controlled excitation conditions. A more complex 'staircase' input waveform allows us to measure step response at particular points over the system's dynamic range and is useful for investigating non-linear behaviour.

The progressive migration towards 'software defined'[1] systems across all application domains is driving the development of high performance bespoke digital signal generation technology that is embeddable within a host system. This embedding can take the form of a software code or a 'programmable logic' (e.g. FPGA) implementation depending on speed, with both implementations satisfying the software definable criterion. Today, applications as diverse as instrumentation, communications, radar, electronic warfare, sonar and medical imaging systems require embedded, digitally controlled signal sources, often with challenging performance and control requirements. Furthermore, many of these applications now require signal sources that generate *non-sinusoidal* waveforms that are specified according to a precisely defined waveshape or spectrum function that is peculiar to the application. Moreover, in addition to conventional frequency, phase and amplitude control, these signal sources can have vastly increased utility by providing parametric and thereby dynamic control of waveshape or corresponding spectrum. As we will see, there are several digital waveform generation techniques that provide this functionality.

After reviewing introductory theoretical material and some established analogue approaches, this book focuses on purely digital techniques for generating waveforms with programmable frequency, phase, amplitude and most importantly *waveshape*. The specification and dynamic control of waveshape (and hence the corresponding spectrum) are relatively new topics in the published literature where hitherto the emphasis has largely been on sine wave generation. We call such user-programmable waveforms *arbitrary waveforms*. The utility of a waveform with an

[1] We take 'software defined' to describe a system which is configurable through *any* locally stored digital data (e.g. software program code, FPGA firmware or memory-based lookup tables).

arbitrary waveshape or spectrum is apparent when we consider that the instantaneous amplitude of that waveform can be engineered to control or emulate any parameter in a digital or analogue system. Some typical examples include:

- set the physical position or velocity profile of a servo mechanism;
- set the operating point of any parametric control system (e.g. the current through a laser diode or the temperature of an oven);
- emulate a parametric sensor output or other 'real-world' signal for test or diagnostic debug purposes;
- modulate the instantaneous amplitude, frequency or phase of a carrier signal;
- provide an input forcing function to measure the time or frequency response of a linear or non-linear system.

Nearly all of the arbitrary and sinusoidal waveform generation techniques presented in this book are based on the established technique of *fixed* sample rate phase accumulation frequency synthesis. Here a phase increment parameter controls output frequency, and a lookup table 'phase–amplitude mapping function' generates the amplitude waveform. This powerful technique provides intrinsically linear and independent control of frequency, phase and amplitude. It provides near instantaneous, phase-continuous frequency transitions and almost unbounded frequency control resolution. With advanced, innovative phase–amplitude mapping we can achieve dynamic control of waveshape and hence corresponding spectrum in real time. An alternative approach to controlling waveform frequency is outlined that is based upon a *variable* sample rate approach, but only as an adjunct to fixed sample rate phase accumulation.

Generation of sine waveforms using phase accumulation and phase–amplitude mapping is well reported in the literature [1] where it is variously known as *direct digital synthesis* (DDS), *direct digital frequency synthesis* (DDFS) or the *numerically controlled oscillator* (NCO). In this book we adopt the DDS acronym. After reviewing sinusoidal DDS as a special case of a general paradigm, we proceed to develop a generalised DDS form that synthesises arbitrary waveforms (including sine waves) according to a time or frequency domain specification. In this book, we denote this as *DDS arbitrary waveform generation* or DDS AWG. For completeness, we include a cursory review of recursive sinusoidal oscillators as they offer a computationally efficient method for digitally generating sine waveforms, albeit with some performance limitations that we compare across several algorithms. The underlying theory and conceptual development of recursive oscillators also serves as an introduction to sinusoidal DDS.

We endeavour to combine a general reference text and a designer's guide that will help appropriately skilled engineers to design bespoke digital waveform generator implementations in technologies appropriate to their application. The reader is assumed to have an undergraduate-level understanding of basic signal processing theory. Waveform synthesis algorithms are presented as a signal-flow of arithmetic and computational block descriptions deliberately abstracted from specific implementation technologies (e.g. FPGA, ASIC or DSP code). Accordingly, it is assumed that design professionals 'skilled in the art' can implement and optimise these descriptions in a hardware or software technology appropriate to their application. Established digital hardware processing techniques that enhance throughput such as sample and block level pipelining, parallel processing and time-division multiplexing are discussed in the specific context of fast, high performance DDS AWG implementations. However, design and optimisation guidance for FPGA, ASIC or DSP software implementation is beyond the scope of this book.

To supplement the written material and to assist the reader in exploring ‘what happens if’ design scenarios, we use Mathcad models to simulate the behaviour of qualitative performance metrics with variation of key design and control parameters. Mathcad models generate many of the graphical figures that are used throughout the book to assist communication of key concepts and behaviours. We use Mathcad because of its popularity, relatively low cost and the ease with which mathematical formulations can be quickly scripted, simulated and visualised in a mathematically stylistic way. These models are available for free download from a website that supports this book: www.petesymons.com/dwg.

1.1 Preliminaries

We begin this section by outlining the chapter content of this book. We proceed to discuss the definition of a waveform and use the sine waveform to explore and develop the key elementary mathematical properties that underpin later discussion on generation techniques. We clarify other descriptive terms associated with electronic waveforms, such as *signal* and *spectrum* both in general and in the context of this book. Finally, we briefly summarise the historical development of digital waveform generation to help set the presented material in context.

1.1.1 Outline chapter content

Chapter 1 begins by discussing the definition, important properties and key parameters of a waveform before proceeding to briefly review the historical development of digital waveform generation. We present a taxonomical grouping of electronic waveform generation techniques and discuss each of the three main subclasses concluding with an outline of the state of the art in digital arbitrary waveform generation. To provide a comparative backdrop to digital waveform generation, we review several established analogue waveform generation methods and summarise their strengths and weaknesses. Finally, we discuss some of the key application areas for bespoke standalone and embedded digital arbitrary waveform generators.

Chapter 2 begins by presenting an outline of several important mathematical concepts that underpin later material. We introduce the fundamental concept of tabulating a sampled waveform in a lookup table – a concept which underpins the *wavetable* (a term we borrow from computer music parlance) discussed further in Chapter 4. We discuss the most important control parameters and their ideal properties, before defining qualitative performance metrics that are pertinent to all of the techniques presented in the book. We use Mathcad models to quantify these metrics under various design and control parameter conditions throughout this book.

In Chapter 3 we introduce the first digital waveform generation technique – the recursive sine wave oscillator – and show how this leads to a generic recursive oscillator description. We proceed to investigate several recursive oscillator structures each having a unique attribute according to a particular performance metric. We discuss implementation considerations common to all recursive oscillators and finally compare their relative properties.

Chapter 4 introduces DDS sine wave generation, beginning with a discussion of phase accumulating frequency synthesis. We investigate several sinusoidal phase–amplitude mapping techniques using computer simulation of the qualitative performance metrics introduced in Chapter 2 to compare their relative performance. The wavetable is introduced together with the related concepts of phase truncation, fractional addressing and interpolated wavetable lookup. Finally, some alternative sinusoidal phase–amplitude mapping techniques are discussed.

Chapter 5 generalises the material introduced in Chapter 4 and investigates DDS arbitrary waveform generation or DDS AWG. We consider methods for filling the wavetable according to different waveform specification techniques, and a fundamental aliasing error mechanism that arises when we ‘sample’ the wavetable by indexing it with a phase accumulator. Phase truncation errors are exacerbated with multi-harmonic waveforms typical of DDS AWG. One method for reducing the magnitude of these errors is the use of phase interpolation according to a fractional phase representation. We present an introductory interpolation tutorial before investigating several interpolated phase–amplitude mapping algorithms with computer simulation of their qualitative performance metrics that were introduced in Chapter 2. We discuss phase accumulating frequency synthesis with ‘analogue waveshaping’ and DDS digital clock generation. This chapter concludes with a discussion of some specific design considerations peculiar to computer music and audio applications.

Chapter 6 explores several methods for dynamic ‘parameterised’ control of waveshape. We proceed to develop dynamically controlled harmonic and non-harmonic waveform generation techniques; the latter enabling the generation of band-pass spectra. One of the techniques investigated is equivalent to the inverse discrete Fourier transform (IDFT), and allows real-time control of the synthesised signal’s harmonic spectrum.

Chapter 7 develops the concept of *phase domain processing* that is introduced in Chapter 4 to efficiently compute the inverse discrete Fourier transform (IDFT) as a DDS phase–amplitude mapping algorithm. This allows the generation of periodic waveforms with independently controllable harmonic amplitude and phase. Techniques for generating waveforms based upon contiguous and non-contiguous (i.e. *arbitrary*) harmonic series are introduced ahead of further development in Chapter 8.

Chapter 8 investigates design considerations for real-time hardware implementation of the techniques presented in earlier chapters including arithmetic pipelining and parallel processing. The chapter presents several design examples, including a novel vector memory suitable for the implementation of phase and wavetable interpolation.

The chapter concludes with some design examples of the contiguous and non-contiguous IDFT waveform generators based upon phase domain processing that are introduced in Chapter 7.

Chapter 9 investigates the design considerations surrounding digital to analogue conversion, focusing on performance metrics and specifications that are peculiar to digital waveform generation applications. We conclude by investigating the post-DAC low-pass reconstruction filter and output signal conditioning.

1.1.2 Digital signal processing

The science of *all* electronic signal processing is primarily concerned with representation, analysis and generation of only two types of signal: analogue or *continuous-time* signals and digital or *discrete-time* signals. From a mathematical perspective, a continuous-time signal is one whose independent time variable is continuous (i.e. 'well-behaved' and free from discontinuity). Conversely, a discrete-time signal is one whose independent time variable is quantised (i.e. only defined in discrete, regularly spaced steps called the *sampling interval* or *sample period*). Accordingly, the signal's instantaneous amplitude value is only defined at discrete instants of time. The discrete-time signal therefore comprises a sequence of impulse functions whose amplitude is equivalent to that of the underlying continuous-time signal at that time instant. Elsewhere, from an analytical perspective, a discrete-time signal is zero-valued. In this book we define a *digital* signal as a discrete-time signal that is also quantised in amplitude according to a particular number representation (i.e. number of bits). Similarly, *digital control* of a parameter implies that the parameter can be changed only at discrete-time intervals. Digital generation of a signal therefore implies the discrete-time execution of an algorithm at fixed sample intervals that processes amplitude quantised internal data values and control parameters.

Historically, purely analogue signal generation techniques that operated in continuous time were prevalent; but with the progressive advances in digital hardware, digital techniques have become well established and now define the state of the art. The emergence and development of digital signal processing (DSP)[2] which we define as *the representation and processing of discrete-time, amplitude-quantised signals*, has revolutionised the field of electronic signal generation as well as many others. Today, the availability of high speed DSP and related technologies (e.g. semiconductor memory) combined with advanced design support software, enables the implementation of bespoke, highly complex signal generation systems that are software defined. Important examples of these 'enabling technologies' include:

[2] There is a subtle, but important distinction between digital signal *processing* and digital signal *processor*, both of which use the DSP acronym. The former relates to the analytical discipline of processing quantised information in discrete-time, whereas the latter describes the hardware in which this processing takes place. In this book, we use 'DSP' to represent both definitions and it is assumed the diligent reader will determine context depending on the prevailing discussion.

- software-programmable digital signal processors and microprocessors;
- 'DSP optimised' field-programmable gate arrays (FPGA);
- fast, high density semiconductor memory;
- analogue to digital convertors (ADC);
- digital to analogue convertors (DAC).

The ADC and DAC, whose development in performance and level of integration has paralleled that of digital hardware, provide fundamental signal processing operations. In the digital signal *generation* field which concerns us here, the DAC provides a crucial operation in moving from the discrete- to continuous-time domain. We should remember that although signals may be generated using purely digital techniques, they are often used or processed further in the analogue domain. In conjunction with a process known as *reconstruction filtering*, the DAC allows us to move from the discrete-time, amplitude-quantised (i.e. digital) domain to the continuous-time (i.e. analogue) domain. This is clearly a fundamental (and occasionally overlooked) operation in digital signal generation systems that are required to produce an analogue output with well-defined performance characteristics. We may digitally synthesise a signal with noise and distortion comparable with the amplitude quantisation level, only to find catastrophic degradation when converted to an analogue signal by poor design or specification of the DAC and its associated analogue signal processing.

1.1.3 Periodic and aperiodic waveforms

Electronic signals are often described as *waveforms*, describing the wave-like variation of voltage with time that is observed when the signal is measured with an oscilloscope. The measured voltage waveform may correspond to a 'true' voltage signal or some other parameter (e.g. current or power) by a proxy measurement (e.g. a current probe which produces an output voltage signal proportional to the current flowing through the device). By definition, a waveform is the *time domain* view of an electronic signal, where we are explicitly concerned with how the signal's instantaneous amplitude varies over time. We describe this variation as *waveshape*.

A typical waveform produced by an electronic signal generator may be defined as *a time-varying, periodic voltage whose waveshape function can be described (or approximated to some level of accuracy) by a mathematical function whose parameters control frequency, phase, amplitude and waveshape*. This function can take on many forms – the sine function, an amplitude and phase weighted Fourier series or a piecewise-linear function that specifies the waveform shape as a collection of linear segments over one cycle of the waveform (e.g. a triangle or sawtooth wave).

In terms of its time domain behaviour, a waveform can be described as either periodic or aperiodic. A periodic waveform repeats over time at a fixed interval called the *period* and the number of waveform cycles observed in one second is called the *frequency*. A waveform that is periodic over some time interval has an *instantaneous frequency* defined on that time interval as the reciprocal of the period. Aperiodic waveforms do not exhibit this property and by definition show no cyclic or repetitive behaviour over *any* time interval.

The mathematical condition for periodicity can be stated quite simply. If a signal whose instantaneous amplitude is denoted by $y(t)$ for all t, where t is the independent time variable, then $y(t)$ is periodic if

$$y(t) = y(t+\tau) \text{ for all } t, \tag{1.1}$$

where τ is defined as the period of the signal.

To help illustrate these concepts, Figure 1.1 provides some examples of periodic and aperiodic waveforms as plots of instantaneous amplitude $y(t)$ against time t.

Figures 1.1a to e illustrate some classic periodic 'function' waveforms – the sine wave, triangle wave, sawtooth wave, half-wave rectified sine wave and full-wave rectified sine wave, respectively. Figure 1.1f illustrates a real-world example of a human ECG waveform indicating normal cardiac function. This is essentially a periodic waveform and a good example of a complex waveform that can be synthesised at a given frequency (i.e. simulated heart rate) using the DDS AWG techniques to be described. Using DDS AWG it is possible to conceive of a cardiac waveshape that changes with frequency, perhaps to emulate a particular anomalous medical condition for training or equipment verification purposes.

Figures 1.1g through 1.1l illustrate several aperiodic waveforms. Clearly, some of these waveforms exhibit some underlying periodic behaviour, but do not satisfy the periodicity criterion expressed in Eq. (1.1). Figure 1.1g shows an aperiodic waveform composed of a sinusoid with occasional 180° phase inversions. If these phase inversions are randomly distributed (as may apply if they correspond to binary data encoding) then this type of waveform is aperiodic for all time. Figure 1.1h is a sinusoid with some additive white noise. When integrated cycle by cycle over a sufficiently long period the *mean* signal approaches a pure sinusoid and is therefore periodic, but the 'cycle to cycle' waveform is strictly aperiodic due to the localised noise variance. Figure 1.1i shows a periodic sinusoid, but with exponentially decaying amplitude typical of the impulse response of an under-damped second-order system. These last three examples illustrate that although a signal may be strictly aperiodic (in an analytical sense) it can have underlying periodic behaviour – in these examples a *sinusoid* signal that is modified by binary phase modulation, additive noise or an exponential amplitude envelope.

Figure 1.1j shows a pure white noise signal which is aperiodic and has a random amplitude distribution. Figure 1.1k shows an exponentially asymptotic curve with a small additive signal comprising a sum of sinusoids with non-harmonic frequencies. Figure 1.1l shows a waveform segment comprising the sum of four sinusoids with non-harmonic frequencies. Such a waveform is *strictly* periodic with a period equal to the least common multiple of the constituent sinusoid periods. However, as the composite period can be extremely large when more than two sinusoids with rational frequencies are added, we can say that such waveforms are 'quasi-aperiodic'. We note at this point, that some DDS AWG techniques (as discussed in Chapter 6, for example) are based on

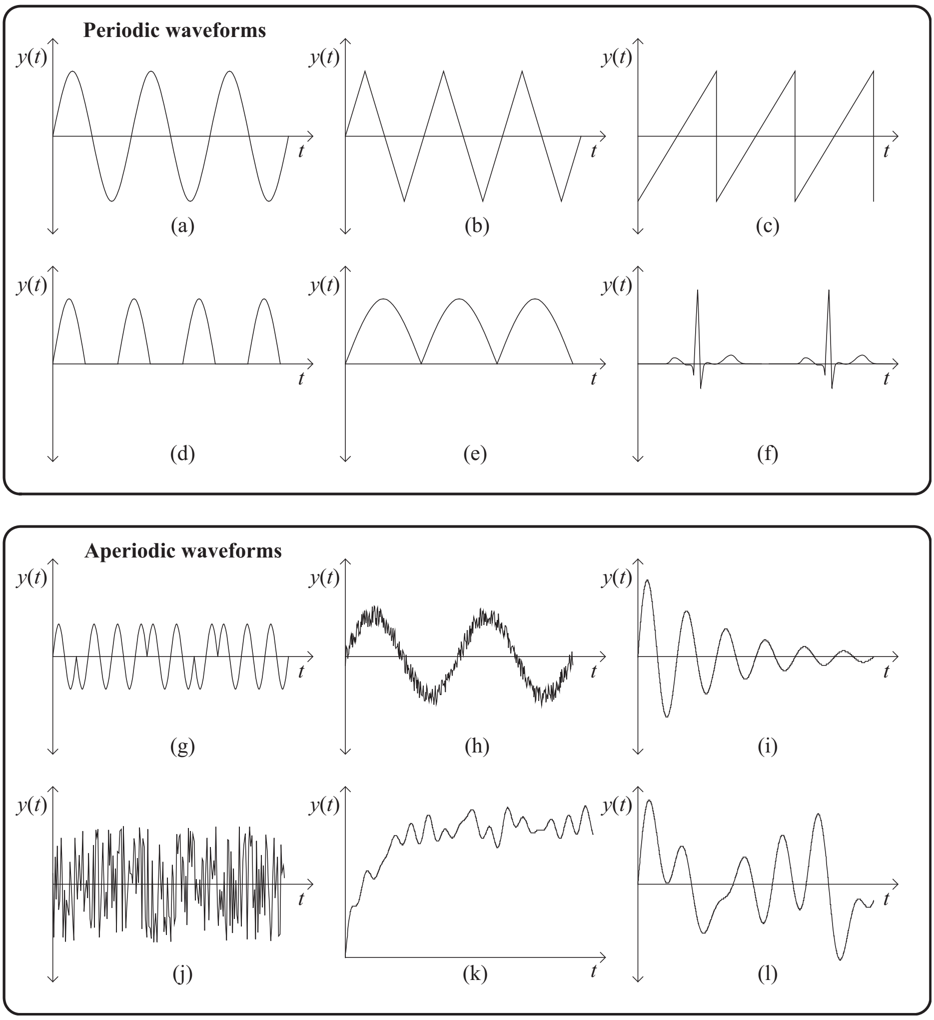

Figure 1.1 Some simple examples of periodic and aperiodic waveforms.

the weighted summation of sine waves with *non-harmonic* frequencies and can therefore easily generate such quasi-aperiodic waveforms.

From a waveform generator perspective, periodic waveforms exist for as long as the generator is switched on and running. Truly aperiodic waveforms are by definition non-repetitive and should strictly be considered as 'single-shot' signals initiated by some trigger event and therefore existing for a predefined time. Aperiodic waveforms, such as Figure 1.1i, become periodic if they are repeated over some interval. Quasi-periodic waveforms are 'locally periodic' waveforms that are arranged to exhibit infrequent transient anomalies that emulate some real-world signal behaviour (e.g. a short-duration transient 'spike' or 'drop out'). An example of a quasi-periodic waveform is a sine wave where 1 in every 10 000 cycles exhibits the pre-defined transient anomaly. Generation of such quasi-periodic waveforms is straightforward using DDS AWG where we dynamically change the phase–amplitude mapping function according to a waveshape control parameter.

1.1.4 Introducing the sine wave – properties and parameters

A fundamentally important periodic waveform is the sine wave. Many physical systems that exhibit resonant or oscillatory behaviour do so with a nominally sinusoidal motion. A classic example is the mass-spring oscillator, which oscillates with an exponentially decaying sinusoid following an impulsive disturbance. In electronic instrumentation, swept frequency sinusoids are used to measure the frequency and phase response of linear systems. Similarly, high purity fixed frequency sinusoids are used to characterise the distortion behaviour of quasi-linear electronic systems (e.g. amplifiers or analogue to digital convertors). In these applications, spectral purity, frequency and amplitude stability are essential attributes.

A sinusoidal oscillation whose instantaneous amplitude at time t we denote by $y(t)$ is defined by the cosine function, thus:

$$y(t) = A\cos(2\pi ft + \theta), \tag{1.2}$$

where A denotes the oscillation amplitude, f denotes the *cyclic frequency* in hertz[3] (or the number of waveform cycles per second) and θ denotes the *initial phase* or *phase offset* in radians relative to $t = 0$. Applying the periodicity criterion expressed in Eq. (1.1) to $y(t)$, we obtain the expression $A\cos(2\pi ft + \theta) = A\cos(2\pi f(t + \tau) + \theta)$. Since the angular period of a cosine (or sine) function is 2π radians, this equality holds for all t if $2\pi f\tau = 2\pi$ or $f = 1/\tau$, thereby proving the reciprocal relationship between sinusoid period and frequency.

The phase offset determines the time locations of the sinusoid maxima and minima relative to $t = 0$ and can be equated to a time shift applied to $y(t)$. To see this, let $y(t)$ be delayed in time by one quarter of a period or $\tau/4$. We then have $y(t - \tau/4) = A\cos(2\pi f(t - \tau/4) + \theta) = A\cos(2\pi ft - \pi/2 + \theta)$ since $f = 1/\tau$. This is equivalent to the original sinusoid $y(t)$ but with an additional phase offset of $-\pi/2$ radians. We also observe that since the sine and cosine functions are periodic with angular period 2π radians, a phase offset of $\theta \pm 2k\pi$, where k is an integer, is indistinguishable from a phase offset of θ. We therefore constrain the phase offset to the interval $\theta \in [0, 2\pi)$ radians.

Figure 1.2 illustrates the cosine waveform and its parameters. Phase offset is exemplified by showing a second dashed sinusoid waveform having a phase offset of $\theta = -\pi/2$ radians relative to the cosine waveform.

The amplitude parameter A is a scaling factor which determines how large the cosine waveform will be. Since the cosine function oscillates between $+1$ and -1, $y(t)$

[3] The standard unit of frequency is the hertz (equivalent to $\sec^{-1}$) and was formally established in 1930 by the International Electrotechnical Commission (IEC) in honour of Heinrich Hertz for his pioneering work on radio waves.

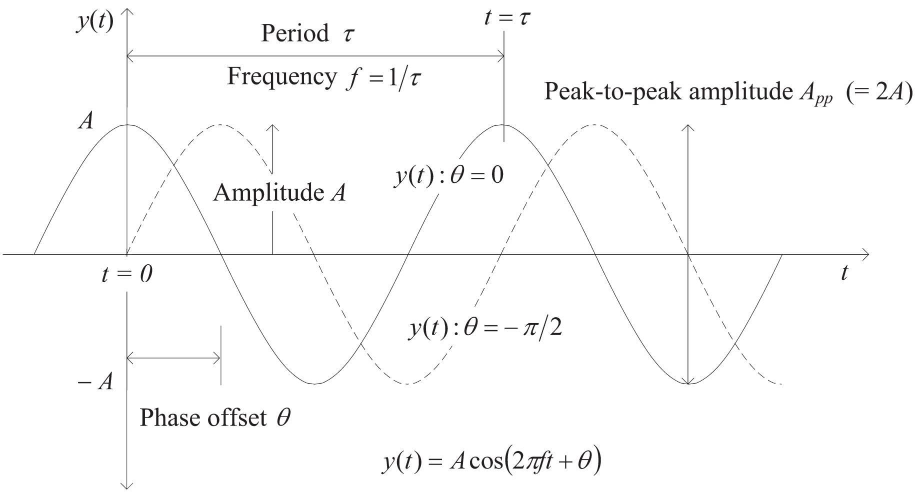

Figure 1.2 A sinusoidal waveform and its parameters.

oscillates between $+A$ and $-A$. There are three amplitude measurements which concern us here – *peak amplitude A*, *peak-to-peak amplitude*, $A_{pp} = 2A$ and the *RMS amplitude*, which is $A/\sqrt{2}$ for the cosine or sine waveform and used when calculating the electrical power of a sinusoidal voltage waveform dissipated in a resistive load. The *mean value* of the cosine or sine waveform taken over an integer number of cycles is precisely zero (assuming zero DC offset).

The waveform *crest factor* is defined as the ratio of peak amplitude to RMS amplitude and for the cosine or sine wave is $\sqrt{2}$. A waveform's crest factor is important when attempting to measure its RMS amplitude, for example, with an electronic AC voltmeter. Many RMS measuring AC voltmeters have an accuracy derating factor which is a function of the signal crest factor being measured. Knowledge of the worst signal case waveform crest factor is important when specifying the headroom relative to RMS before the onset of clipping in both analogue and digital processing chains. In the context of digital waveform generation, crest factor is important since the *signal to noise ratio* (SNR) performance metric is defined relative to the RMS signal amplitude. Waveforms with a high crest factor have a reduced RMS amplitude to avoid peak clipping with a given amplitude dynamic range (i.e. number of DAC bits). Therefore, for a given noise amplitude, high crest factor waveforms have lower SNR.

These amplitude measurements are not just peculiar to sinusoids, they are applicable to *all* periodic waveforms and it is instructive to outline their interrelationship for other simple periodic waveforms. Figure 1.3 tabulates the mean and RMS amplitudes and crest factors of several simple waveforms as a function of normalised peak amplitude.

1.1.5 Instantaneous phase and frequency

The argument of the cosine function in Eq. (1.2) is called the *instantaneous phase* $\phi(t)$, and for a constant frequency sinusoid is given by: