

# An introduction to signal processing for speech\*

Daniel P.W. Ellis

LabROSA, Columbia University, New York

October 28, 2008

## Abstract

The formal tools of signal processing emerged in the mid 20th century when electronics gave us the ability to manipulate signals – time-varying measurements – to extract or rearrange various aspects of interest to us i.e. the *information* in the signal. The core of traditional signal processing is a way of looking at the signals in terms of sinusoidal components of differing frequencies (the Fourier domain), and a set of techniques for modifying signals that are most naturally described in that domain i.e. filtering. Although originally developed using analog electronics, since the 1970s signal processing has more and more been implemented on computers in the digital domain, leading to some modifications to the theory without changing its essential character. This chapter aims to give a transparent and intuitive introduction to the basic ideas of the Fourier domain and filtering, and connects them to some of the common representations used in speech science, including the spectrogram and cepstral coefficients. We assume the absolute minimum of prior technical background, which will naturally be below the level of many readers; however, there may be some value in taking such a ground-up approach even for those for whom much of the material is review.

---

\*To appear as a chapter in *The Handbook of Phonetic Science*, ed. Hardcastle and Laver, 2nd ed., 2009.

# 1 Resonance

Consider swinging on a child's swing. (It may be a while since you've been on one, but you can probably remember what it was like.) Even without touching the ground, just by shifting your weight at the appropriate points in the cycle, you can build up a considerable swinging motion. The amount of work you put in at each cycle is quite small, but it slowly builds up, until it is enough to lift you high off the ground at each extreme. Building up the swing requires making the right movements at just the right time – it can take a child a while to figure out how to do this. More vigorous movements can build up the swinging more quickly, but the cycle time – the time between two successive instants at the same point in the cycle, for instance the highest point on the back of the swing – is basically unvarying with the amplitude of the swinging, the amount of work you put in. Even the size and weight of the child doesn't have much effect, except in so far as their center of mass gets further from the seat as they get bigger.

The swing is an example of a pendulum, a simple physical system that can exhibit oscillations, or a pattern of motions that repeats with little variation with a fixed repetition time. The conditions that support this kind of oscillation are relatively simple and very common in the physical world, meaning that the simple mathematics describing the relationship between the input (child's weight shifts) and output (swing motion) apply largely unmodified in a wide range of situations. In particular, we are interested in the phenomenon of *resonance*, which refers to the single “best frequency” – the way that the largest swinging amplitude occurs when the swinger injects energy at a single frequency that depends on the physical properties of the oscillating system (which are usually fixed).

The swing is an interesting example because it reveals how small amounts of work input can lead to large amplitude output oscillations, provided the inputs occur at the right frequency, and, importantly, the right point in the cycle i.e. the correct “phase” relationship to the motion. But another pendulum example can more clearly illustrate the idea of frequency selectivity in resonance: Consider a weight – say some keys or a locket – on the end of a chain. By holding the top

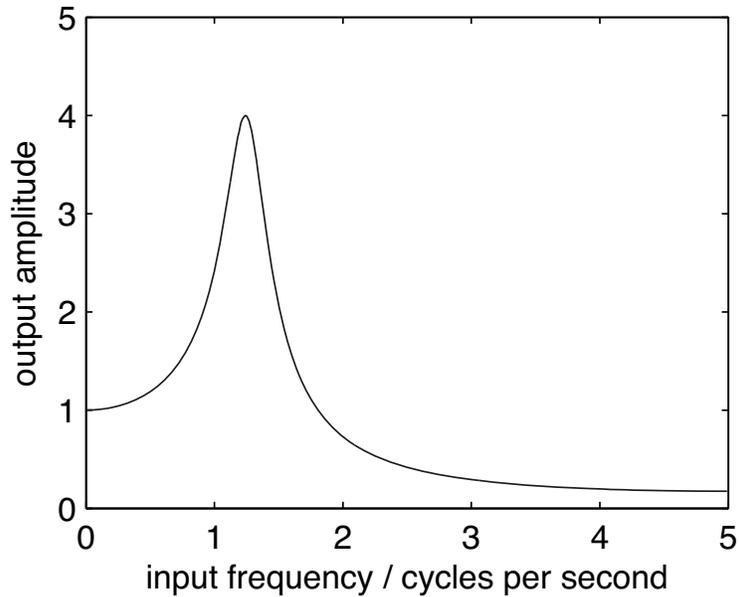


Figure 1: Sketch example of the amplitude of pendulum motion in response to a fixed amplitude input of varying frequency.

of the chain, and moving your hand from side to side, you can make the weight move from side to side with the same period. However, the *amplitude* of the weight's motion, for a fixed amplitude of hand motion input, will vary greatly with the cycle period, and for a small range around a particular frequency, the *natural frequency* of the setup, the output motion will be very large, just like the motion of the swinging child. If you deliberately start moving your hand slightly faster or slightly slower than this best frequency, you will still be able to make the weight oscillate with the same period as your hand, but the amplitude will fall rapidly as you move away from the natural frequency. We could make a plot of the amplitude of the side-to-side motion in response to a fixed amplitude input motion as a function of the frequency of that input motion, and it might look something like figure 1 below. At low frequencies, the ratio between input and output motion is approximately 1 i.e. when moved slowly, the bottom of the pendulum follows the top. Around the natural frequency, the ratio is very large – small motions of the pendulum top lead to wild swinging, at the same frequency, of the weight. At high frequencies, the ratio tends to zero: rapid motion at the top of the pendulum is 'lost', leaving the weight almost stationary.

## 2 Sinusoids

Mathematical equivalents of the pendulum and a few simple variants are remarkably common in the natural world, ranging from the task of trying to rock a trapped car out of a snowdrift to the shaking of the earth's crust after an earthquake, all the way to the quartz crystal at the heart of a digital watch or a radio antenna. There are a couple of aspects of this common phenomenon which we should note at this point relating to the shape of the resonant waveform.

Figure 2 returns to the swing example, plotting the position of the weight (the child) as a function of time. We see the regular, periodic motion appear as a repeating forward-backward pattern, but the particular shape of this pattern, the smooth alternating peaks of the *sinusoid*, is essential to and characteristic of this behavior. You probably came across sinusoids in trigonometry as the projections of a fixed length (hypotenuse) at a particular angle, but that doesn't seem to provide much insight to their occurrence here. Instead, the interesting property of the sinusoid is its slope – the rate of change at any given time – is another sinusoid, at the same frequency (of course, since the whole system repeats exactly at that frequency), but shifted in time by one-quarter of a cycle. The slope of a graph showing position as a function of time is *velocity*, the rate at which position is changing, and this is shown to the right of the figure. In the plot, the relative amplitudes of position and velocity are arbitrary, since they depend on the different units used to measure each quantity.

Looking in detail at the figure, there are four time points labeled, with a snapshot of the swing shown for each one. At time A, the beginning of the graph, the swing is hanging straight down, but it is about to move forward, perhaps because the swinger has just pushed off against the ground. Thus the position is at zero, but the velocity is positive and indeed at its maximum value. At point B, the swing has returned to its straight-down position, but now it is traveling backwards, so we see that the velocity is at its maximum backwards (negative value). Conversely, point C finds the swing at its maximum positive position, when the motion is changing from forward to backward, so the swing is still for a split second – i.e at zero velocity. D is the symmetric case at the very

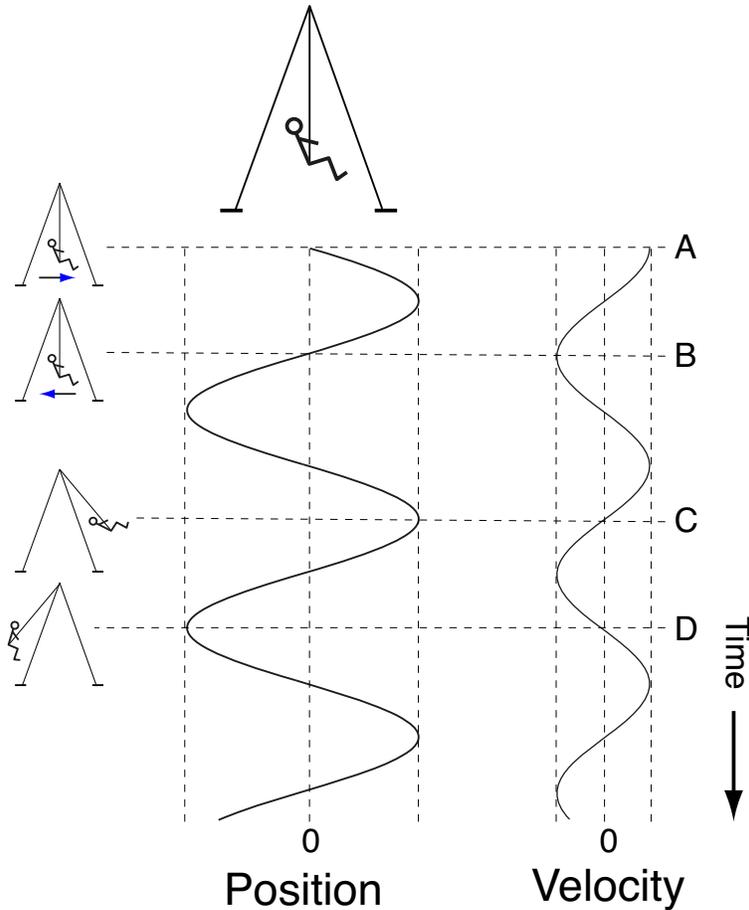


Figure 2: Motion of the simple pendulum swing. The slope (derivative) of the plot of position with time gives the velocity, which is shown on the right. Small sketches to the left show the instantaneous configuration of the swing at the times labeled A to D. Note that both position and velocity are sinusoids, a waveform with a ‘pure’ frequency, and that there is a one-quarter cycle phase shift between them.

back of the swing, again a moment of zero velocity.

If we normalize time to be in units of complete cycles, we can equate it to the angles of trigonometry, in which case we speak of the *phase* of the sinusoid, and measure it either in degrees  $360^\circ$  for a complete cycle) or radians ( $2\pi$  rad for a complete cycle). At a given frequency, a time shift is equivalent to a phase shift, and in this case the quarter-cycle difference between position and velocity corresponds to a  $90^\circ$  or  $\pi/2$  rad *phase shift*.

Resonant systems of this kind involve the periodic transfer of energy between two forms. In this case, they are the kinetic energy of the motion of the weight, which is proportional to the

square of the velocity. At points A and B, all the system in the swing is in the kinetic energy of the swinger's motion – energy that would be violently transferred to someone unlucky enough to step into the path of the swing. The complementary energy form is potential energy, the energy gained by lifting the swinger up against gravity away from the ground due to the arc traced by the swing. At points C and D, when the swing is momentarily stationary (zero kinetic energy), the swinger is also highest from the ground, corresponding to maximal potential energy. The total of kinetic plus potential energy is constant throughout the cycle, but at all other points it is shared between the two forms. Not only does resonant behavior always involve such an exchange between two energy forms, but the particular  $90^\circ$  phase shift between the two domains we can see is also a common characteristic. It is the constant transfer of energy between these two forms that leads to the visible, dynamic behavior, even without any additional energy input i.e. if the swinger remained frozen on the seat.

This brings us to a second aspect of resonant motion – *exponential decay*. If the swinger builds up a large-amplitude motion, then stops shifting their weight to inject energy into the system, the sinusoidal motion continues, but its amplitude gradually decays away until the swing is essentially still. What has happened is that the energy is being lost e.g. to air resistance and heating up the moving joints of the swing mechanism, so the maxima of kinetic and potential energy are steadily decreasing. Figure 3 shows an exponentially-decaying sinusoid, as might occur with undriven swinging. Notice that the amplitude decays by a constant factor for each unit of time i.e. if it decays to half its initial amplitude by time  $T$ , it has decayed to one quarter at time  $2T$ , one eighth at time  $3T$ , one sixteenth at time  $4T$  etc. This is the hallmark of exponential decay.

Exponential decay of an oscillating, resonant system is also related to its “tuning” i.e. how sharply it responds to its natural frequency. This is visible as the sharpness of the peak in a plot like figure 1, and actually depends on the rate of energy loss. A highly tuned resonance has a very sharp resonant peak as a function of frequency, and its resonant oscillations die away very slowly. A more strongly “damped” resonance dissipates more energy each cycle, meaning that oscillations die away more rapidly, and the preference for the natural frequency over other oscillation frequencies

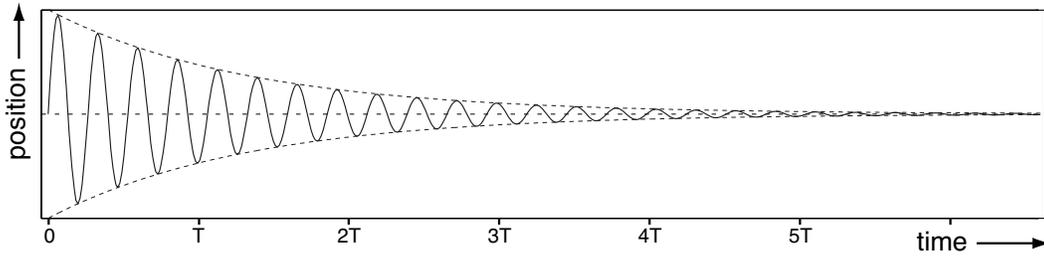


Figure 3: Exponential decay. The sinusoid amplitude steadily decreases, losing a fixed *proportion* of its amplitude in a fixed time, i.e. it halves in amplitude every  $T$  seconds.

is less dramatic. As an example, the shock absorbers (also known as dampers) in a car’s suspension system are needed to minimize “bouncing” oscillations that would otherwise occur with the spring-mass system at each wheel. When the shock absorbers age or fail they lose their ability to dissipate energy, and the car will develop a tell-tale (and very disconcerting) tendency to oscillate up and down every time it goes over a bump.

A simple resonance always involves sinusoidal motion. Returning to the pendulum being shaken at its top, even if you move your hand in a much less smooth way, for instance shifting almost instantaneously from one extreme to the other for a square-wave input, the motion of the weight will still be essentially a sinusoid with the same period. And if you shake your hand randomly, most often the weight will begin to move at its resonant frequency with a sinusoidal motion – as we will see below, we interpret this as the resonant system “filtering out” a frequency component at the natural frequency from the random motion.

Lest we lose sight of the motivation of this entire discussion, figure 4 shows a brief excerpt of a voice waveform, extracted from a vowel sound. We see a few cycles of the fundamental voice period, but notice how within each period, the waveform looks a lot like an exponentially-decaying sinusoid. This is because the vocal tract, shaping the sound, is behaving as a simple resonant system being shocked into motion once every pitch cycle.

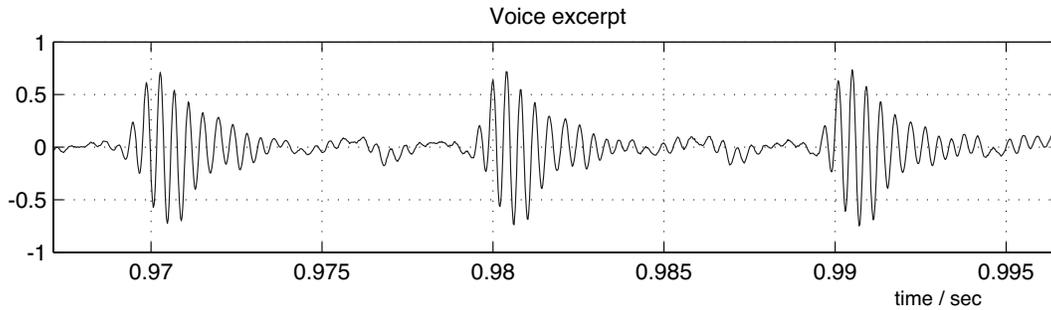


Figure 4: Example of vocal waveform during a vowel. We see three cycles of the voice pitch; within each cycle, the waveform looks a lot like an exponentially decaying sinusoid.

### 3 Linearity

Prior to presenting the central idea of Fourier analysis, there is one more supporting concept to explore: Linearity. Very roughly, linearity is the idea that scaling the input to a system will result in scaling the output by the same amount – which was implicit in the choice of using the *ratio* of input to output amplitudes in the graph of figure 1 i.e. the ratio of input to output did not depend on the absolute level of input (at least within reasonable bounds). Linearity is an idealization, but happily it is widely obeyed in nature, particularly if circumstances are restricted to small deviations around some stable equilibrium.

In signal processing, we use ‘system’ to mean any process that takes a signal (e.g. a sound waveform) as input and generates another signal as output. A linear system is one that has the linearity property, and this constitutes a large class of real- world systems including acoustic environments or channels with rigid boundaries, as well as other domains including radio waves and mechanical systems consisting of rigid connections, ideal springs. and dampers. Of course, most scenarios of interest also involve some nonlinear components, e.g. the vocal folds that convert steady air pressure from the lungs into periodic pressure waves in the (largely linear) vocal tract.

Linearity has an important and subtle consequence: *superposition*. The property of superposition means that if you know the outputs of a particular system in response to two different inputs, then the output of the system in response to the *sum* of the two inputs is simply the sum of the two outputs. Figure 5 illustrates this. The left columns show inputs, and the right columns show

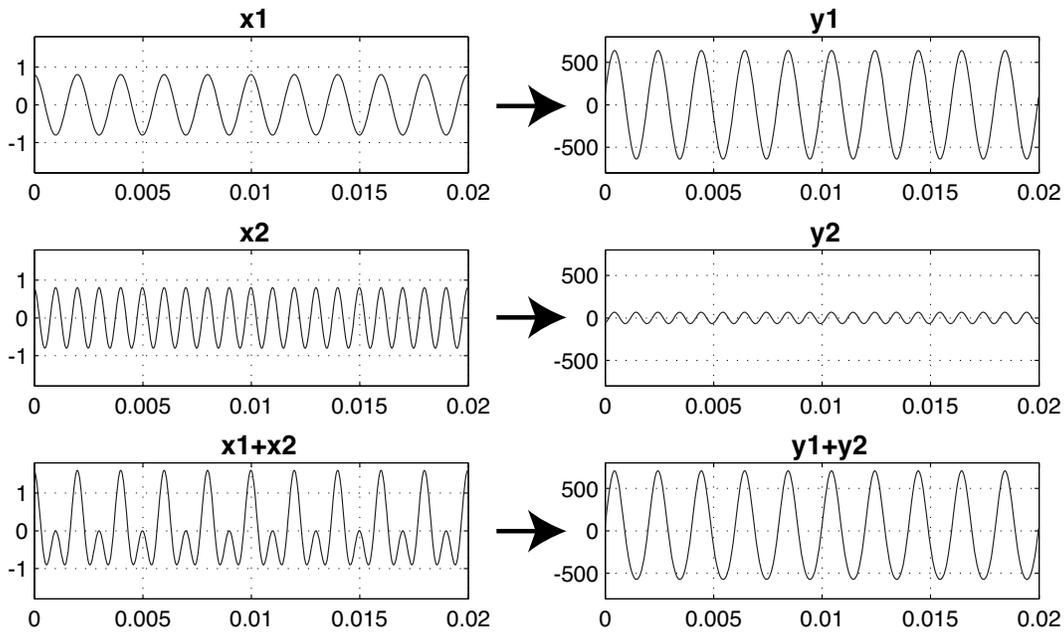


Figure 5: Illustration of superposition. Left column shows three inputs to a linear system, where the third is the sum of the preceding two. Right column shows the corresponding outputs; because the first row has a much larger amplitude than the second, the third looks largely the same as the first.

outputs, for a simple resonant system as described in section 1. The first row shows the system response when the input is a sinusoid right around the natural frequency. Notice that the scale on the output graph is much larger, reflecting the increased amplitude of the output sinusoid. The second row is for a sinusoid at double the frequency, outside the resonant peak. The output is still a sinusoid at the same frequency, but its amplitude is much smaller than in the resonant case. Finally, adding the two inputs results in a waveform with the same basic period as the first row, but no longer sinusoidal in shape. For this linear system that obeys superposition, the output is simply the sum of the outputs from the previous two conditions. But because the output in the first row was so much larger than that in the second row, the sum is dominated by the first row, so the contribution of the higher frequency component in the input is practically negligible.

Now we can learn one more very important property of sinusoids: they are the *eigenfunctions* of linear systems. What this means is that if a linear system is fed a sinusoid (with or without an exponential envelope), the output will also be a sinusoid, with the same frequency and the same

rate of exponential decay, merely scaled in amplitude and possibly shifted in phase. The scale factor (or *gain*) and shift angles will depend on the sinusoid frequency, but are otherwise constant properties of the system. Any other waveform will in general be modified in a more complex way that cannot be explained by a single, constant factor or phase shift. This property is evident in figure 5, where the output of the resonance to each of the two sinusoids is simply a scaled and delayed version of the input, but the more complex waveform is extensively modified (to look more like a sinusoid). This sinusoid-in, sinusoid-out property, combined with superposition, is the key to the value of Fourier transform, presented in the next section, which describes an arbitrary input as a sum of sinusoids.

One last definition: If a system's property changes – for instance if a acoustic tube changes shape or length – the specific scaling and phase shift values for each sinusoid frequency will likely change too. Much of our analysis assumes *time-invariant* systems, so that we can assume the way in which a signal is modified does not depend on precisely *when* the signal occurs. However, even systems which are not time-invariant (such as the human vocal tract) can be treated as *locally* time invariant i.e. the modification applied to a given sinusoid will change only relatively slowly and smoothly, so the linearity assumptions can be applied successfully over sufficiently short time scales.

## 4 Fourier analysis

The core of signal processing is Fourier analysis, and the core of Fourier analysis is a simple but somewhat surprising fact: Any periodically-repeating waveform can be expressed as a sum of sinusoids, each scaled and shifted in time by appropriate constants. Moreover, the only sinusoids required are those whose frequency is an integer multiple of the fundamental frequency of the periodic sequence. These sinusoids are called the *harmonics* of the fundamental frequency.

To get an arbitrarily good approximation to an arbitrary periodic waveform, it may be necessary to include a very large number of sinusoids i.e. continue up to sinusoids whose frequency is very

high. However, it turns out that the scaling of any single sinusoid that gives the best approximation doesn't depend on how many sinusoids are used. Thus, the best approximation using only a few sinusoids can be derived from a higher-order, more accurate approximation simply by dropping some of the harmonics.

It makes sense that the only sinusoids involved are the harmonics (integer multiples of the fundamental frequency), since only these sinusoids will complete an exact, integer number of cycles in the fundamental period; any other sinusoids would not repeat exactly in each period of the signal, and thus could not sum to a waveform that was exactly the same every period.

Figure 6 illustrates the Fourier Series concept. The original periodic waveform is a square wave i.e. with abrupt transitions from +1 to -1 and back in each cycle. It is particularly surprising that the sum of a series of smooth sinusoid functions can even approximate such a discontinuous function, but the figure illustrates how the first five Fourier components (which in this case consist only of *odd* multiples of the fundamental frequency), when appropriately scaled and aligned, begin to reinforce and cancel to match the piecewise-constant waveform. We also note that in the special case of the square wave, the harmonic amplitudes are inversely proportional to the harmonic number.

It turns out that finding the Fourier series coefficients – the optimal scale constants and phase shifts for each harmonic – is very straightforward: All you have to do is multiply the waveform, point-for-point, with a candidate harmonic, and sum up (i.e. integrate) over a complete cycle; this is known as taking the *inner product* between the waveform and the harmonic, and gives the required scale constant for that harmonic. This works because the harmonics are *orthogonal*, meaning that the inner product between different harmonics is exactly zero, so if we assume that the original waveform is a sum of scaled harmonics, only the term involving the candidate harmonic appears in the result of the inner product. Finding the phase requires taking the inner product twice, once with a cosine-phase harmonic and once with the sine-phase harmonic, giving two scaled harmonics that can sum together to give a sinusoid of the corresponding frequency at any amplitude and any phase.

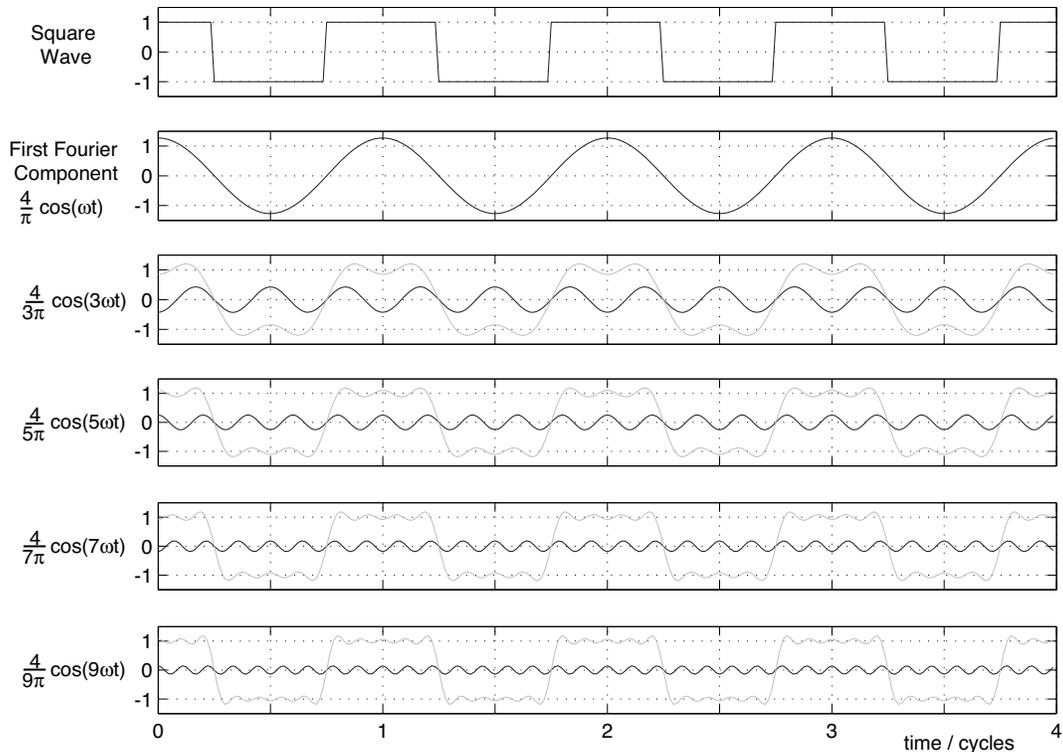


Figure 6: Illustration of how any periodic function can be approximated by a sum of harmonics i.e. sinusoids at integer multiples of the fundamental frequency of the original waveform. Top pane shows the target waveform, a square wave. Next panes show the first five harmonics; in each pane, the dark curve is the sinusoid, and the light curve is the cumulative sum of all harmonics so far, showing how the approximation comes increasingly close to the target signal.

Finding the Fourier series representation is called Fourier analysis; the converse, Fourier synthesis, consists of converting a set of Fourier coefficients into a waveform by explicitly calculating and summing up all the harmonics. A waveform that is created by Fourier synthesis will yield the exact same parameters on a subsequent Fourier analysis, and the two representations – the waveform as a function of time, or the Fourier coefficients as a function of frequency, may be regarded as equally valid descriptions of the function, i.e. together they form a *transform pair*, one in the time domain, and the other in the frequency, or Fourier, domain.

If Fourier analysis could be applied only to strictly periodic signals it would be of limited interest, since a purely periodic signal, which repeats exactly out to infinite time in both directions, is a mathematical abstraction that does not exist in the real world. Consider, however, stretching

the period of repetition to be longer and longer. Fourier analysis states that within this very long period we can have any arbitrary and unique waveform, and we will still be able to represent it as accurately as we wish. All that happens is that the ‘harmonics’ of our very long period become more and more closely spaced in frequency (since they are integer multiples of a fundamental frequency which is one divided by the fundamental period, which is becoming very large). Put another way, to capture detail up to a fixed upper frequency limit, we will need to specify more and more harmonics.

Now by letting the fundamental period go to infinity, we end up with a signal that is no longer periodic, since there is only space for a single repetition in the entire real time axis; at the same time, the spacing between our harmonics goes to zero, meaning that the Fourier series now becomes a continuous function of frequency, not a series of discrete values. However, nothing essentially changes – and, in particular, we can still find the value of the Fourier transform function simply by calculating the inner product integral. Now we have the most general form of the Fourier transform, pairing a continuous, non-repeating (aperiodic) waveform in time, with a continuous function of frequency. In this form, the symmetry between time and frequency (and the lack of privilege for either domain, from the point of view of mathematics) begins to become apparent.

One kind of aperiodic waveform that might interest us is a finite-length waveform i.e. a stretch of signal that exists over some limited time range, but is zero everywhere else. Since it never repeats, its Fourier transform is continuous. However, it turns out that the constraint of finite extent in time imposes smoothness in the frequency domain, meaning that we can be sure not to miss any important detail if we only evaluate the Fourier transform at a limited number of regularly-spaced frequency points.

As an example, figure 7 shows the brief speech excerpt from figure 4 along with the magnitude of its Fourier transform, up to 4 kHz. (The magnitude is the length of the hypotenuse of the right-angle triangle formed by the sine and cosine coefficients for a particular frequency, and corresponds to the amplitude of the implied sinusoid). A Fourier transform magnitude plot like this is commonly known as a magnitude *spectrum*, or just spectrum. It is shown in two forms: the middle

pane uses a linear magnitude axis, and the bottom pane plots the magnitude in decibels (dB), a logarithmic scale that reveals more detail in the low-amplitude parts of the spectrum. Note that the 80 dB vertical range in the bottom plot corresponds to a ratio in linear magnitude of 10,000:1 between most and least intense values. The time waveform has been scaled by a tapered window (shown dotted) to avoid abrupt transitions to zero at the edges, which would otherwise introduce high-frequency artifacts. The time-domain signal is zero everywhere not shown in the image. We notice dense, regularly-spaced peaks in the spectrum; these arise because of the pitch-periodicity evident in the waveform (i.e. the repetitions at roughly 10 ms). If the signal were exactly periodic and infinitely repeating, these spectral peaks would become infinitely narrow, existing only at the harmonics of the fundamental frequency; the Fourier transform would become the Fourier series. Superimposed on this fine structure we see a broad peak in the spectrum centered around 2400 Hz; this is the vocal tract resonance being driven by the pitch pulses, and corresponds to the rapid, decaying oscillations we noticed around each pitch pulse in the time domain. A quick count confirms that these oscillations make around 12 cycles in 5 ms, which indeed corresponds to a frequency of 2400 Hz. The spectrum has no significant energy above 4 kHz, although the dB-domain plot shows that the energy has not fallen all the way to zero.

## 5 Filters

Taken alone, the existence of the Fourier transform might be no more than a mathematical curiosity, but in combination with our previous examination of the properties of sinusoids, linearity, and superposition, it becomes extremely powerful. Recall that in section 3 we said that (a) feeding a linear, time-invariant system with a single sinusoid will result in a scaled and phase shifted sinusoid of the same frequency at the output, and (b) the output of a linear system whose input is the sum of several signals will be the sum (superposition) of the system's output to each of those signals in isolation. The Fourier transform allows us to describe any signal as the sum of a (possibly very large) set of sinusoids, and thus the output of a particular system given that signal as input

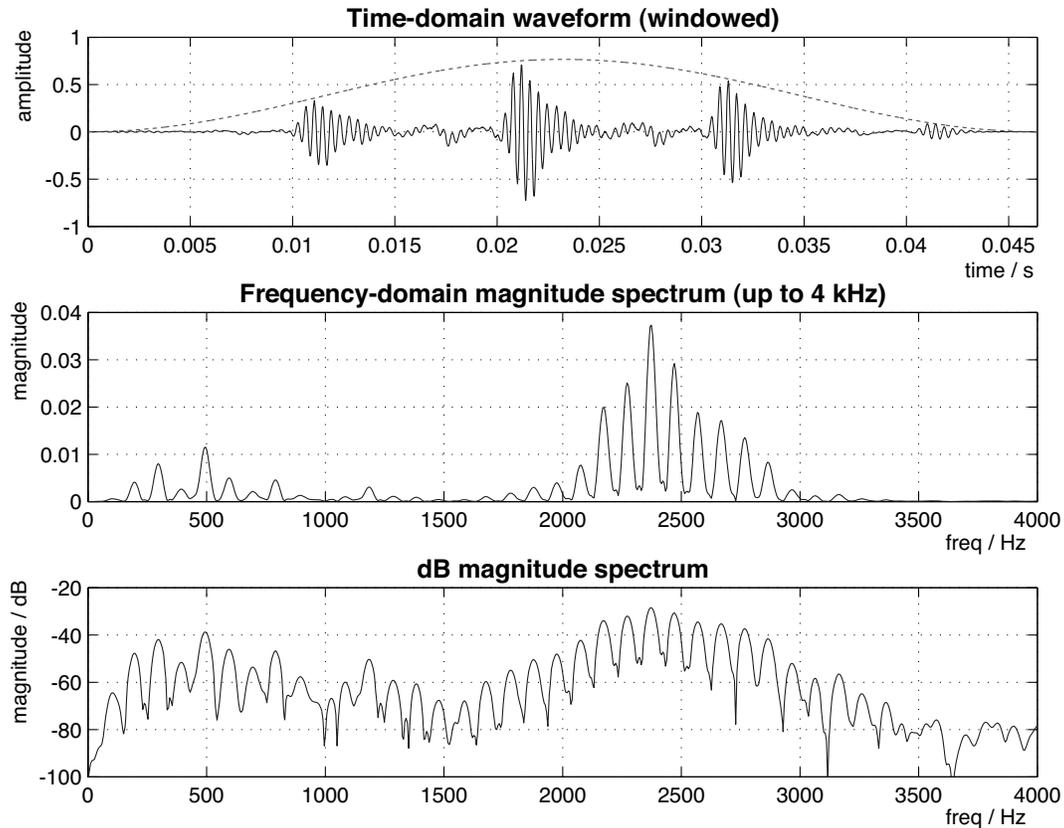


Figure 7: Fourier transform pair example. Top pane: Time domain waveform of a brief vowel excerpt, windowed by a raised cosine tapered window (dotted). Middle pane: Fourier transform (spectral) magnitude, up to 4 kHz. Bottom pane: spectral magnitude using a dB (logarithmic) vertical scale.

will be the *same* set of sinusoids, but with their amplitudes and phases shifted by the frequency-dependent values that characterize that system. Thus, by measuring – once – the way in which a system modifies sinusoids of all relevant frequencies, it is a simple matter to predict the Fourier transform (and hence the time-domain waveform) of the system’s output in response to any input signal described by its Fourier transform.

In signal processing, a *filter* is essentially any system with an input and an output, but the term implies that the properties of the system are being viewed as emphasizing certain aspects of the signal while reducing or removing certain others. In a linear time invariant filter, it is the Fourier components – sinusoids of differing frequencies – that are selected, meaning that they are either amplified (made larger) or attenuated (made smaller). There are infinitely many possible filters,

even within this relatively narrow, idealized set, but they are typically categorized according to the broad properties of how their scaling effects vary with frequency: a low-pass filter boosts low frequencies close to zero; high pass does the converse, attenuating lower frequencies; band-pass selects frequencies within a limited range, and band-stop or notch filters remove specific frequency ranges. Note that the simple resonance with which we introduced this chapter in figure 1 is a kind of band-pass filter. One way to make a band-stop filter is to ‘cancel out’ energy at certain frequencies (e.g. by adding it to a negated version of itself, corresponding to a  $180^\circ$  phase shift) leaving the low and high frequencies. Resonances in systems are often called “poles”, referring to a specific feature of the mathematical description of the system; the locally-attenuating aspects, as found in band-stop filters, are known as “zeros” because they can remove certain sinusoidal or decaying-sinusoidal inputs to give zero output.

Much of the foundation of signal processing involves techniques to design and construct filters to achieve specific goals and characteristics. There are a number of ‘optimal’ design procedures that design filters, for implementation in electronics or software, that do the best possible job in terms of leaving some frequencies unmodified while removing others, subject to various constraints such as cost or complexity. While linear filtering is a relatively limited subset of all possible signal modifications – only slightly more complex than the “treble” and “bass” controls on a hi-fi amplifier – it turns out to be very useful in a wide range of applications, particularly when trying to separate a particular piece of information, such as a particular voice, from the middle of a large amount of background noise.

As we just mentioned, filters can be implemented in a variety of forms: it is possible to build *acoustic* systems with controlled resonant properties, such as an organ pipe which is a very sharply-tuned band-pass filter coupled to a nonlinear air-jet oscillator. (You may have noticed that the sound of a pure sinusoid is reminiscent of an organ or a flute). However, the birth of signal processing occurred when it became possible to represent signals (be they audio, radio, or others such as television) as electrical voltages and process them using electronic circuits. Much of the theoretical foundation was based on analog electronics, but from the 1960s onwards more and more signal

processing has been performed on digital computers using signals represented as sequences of values stored in memory.

This required a modified theoretical foundation, known as discrete-time or digital signal processing (DSP), because whereas an analog voltage can in theory vary at any frequency from the very slow to the extremely fast, a digital representation involves measuring and storing the voltage only at a finite set of discrete instants (usually regularly spaced). DSP systems usually have a fixed sampling rate, which is the number of samples recorded every second, and any variations in the signal which involve significant detail below the timescale of this sampling will not be accurately captured. In fact, it turns out that to store components up to some particular frequency, it is necessary to sample at at least double that frequency, thus the highest correctly-represented frequency is half the sampling rate, known as the Nyquist frequency. For example, in CD digital audio, the sampling rate was chosen as 44.1 kHz to ensure that highest frequencies perceptible by humans – around 20 kHz – could be adequately represented, with the extra 10% providing some breathing space to make it easier to construct the digital-to-analog converters required to render the digital representation back into an actual, physical sound for listeners. The problems that arise when sampling a frequency higher than the Nyquist rate are illustrated in figure 8.

## **6 The Spectrogram**

Filters and signal processing turn up in many places in phonetics and speech science, from cleaning up field recordings through to performing data compression on archives, but perhaps its greatest impact is in providing analysis tools that can measure and quantify different acoustic phenomena, and perhaps the most familiar of those is the spectrogram. We can now precisely describe how a spectrogram image is constructed, using the ideas presented so far, but first we must mention one more reason why sinusoids and the frequency domain are so important and relevant for sound: the nature of hearing.

Once air pressure fluctuations have been collected by the outer ear, converted into microscopic

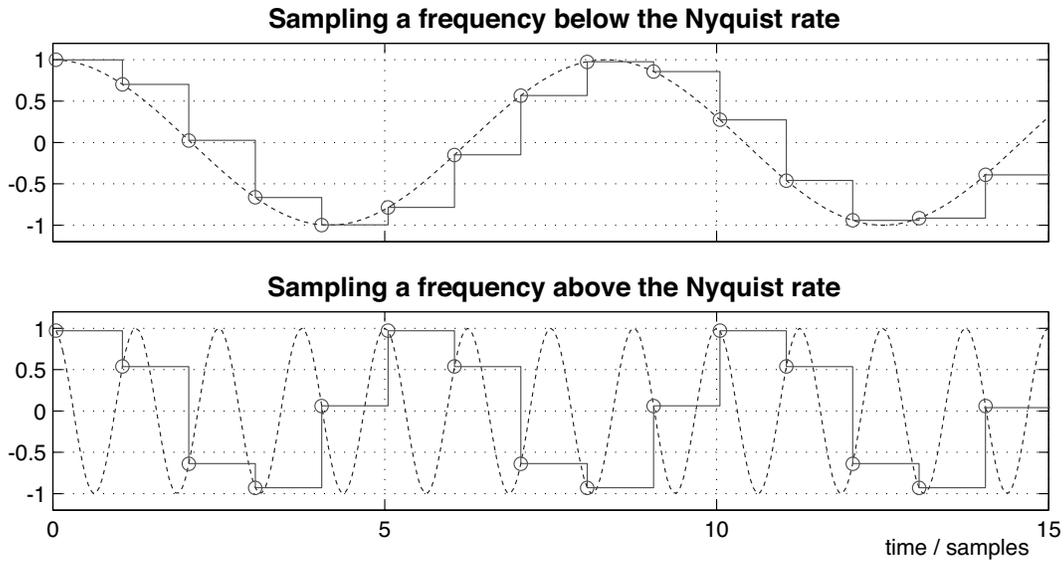


Figure 8: Illustration of discrete-time (sampled) representations of sinusoids below (top) and above (bottom) the Nyquist rate. Analog domain signals are shown as dotted, and sampled values are shown with circles, connected by stairstep lines. Note that the discrete, stairstep signal captures the general shape of the below-Nyquist frequency sinusoid, but for the higher frequency, the sampled representation appears to reflect a completely different periodicity.

force variations by the eardrum, and transferred to the inner ear by the bones of the middle ear, they encounter the single most critical component of auditory perception, the cochlea, which is responsible for converting a one-dimensional time-varying pressure into a sequence of firings on the tens of thousands of nerve fibers flowing to the brain down the auditory nerve. The cochlea is exquisitely sensitive and very complex, but in essence it is just a series of resonators, not so very different from the one described by figure 1. Each resonator responds to energy in a narrow frequency range, causing a certain subset of nerve fibers to fire when there is energy at those frequencies. Thus, the auditory system performs something like a Fourier transform, breaking down the time-domain pressure fluctuations into separate sinusoidal components through a bank of resonant filters.

The representation on the auditory nerve, however, is not a pure frequency domain representation: Although different sets of nerves indicate energy in different frequencies, they also vary in time – whereas a pure Fourier representation would only have a frequency axis. In fact, the

transformation performed by the cochlea is closer to a short-time Fourier transform (STFT), which breaks up a longer signal into a succession of smaller fragments, centered around different, specific times by gating the original signal with a sliding window, then calculating the Fourier transform of each of these time-localized pieces to reveal the varying energy in each frequency bands as the input signal changes. Note that this is exactly equivalent to constructing a set of bandpass filters, one for each frequency being considered, then calculating the total energy coming out of each band over a succession of short windows.

Although there is rather more to how the ear works than this, the magnitude of the short-time Fourier transform is exactly what we see when we look at a spectrogram, and one of the reasons it is such a valuable tool for visualizing sound content is because of its correspondence, in very broad terms, to the internal representation of sound employed by our brains. A spectrogram is actually a fine grid of cells, indexed by frequency on the vertical axis, and by time on the horizontal axis, where the color (or darkness) of each cell indicates the amount of energy in that frequency band at around that time. The spectrogram typically uses a logarithmic mapping from signal intensity to pixel darkness (i.e. a deciBel scale), corresponding to the nearly-logarithmic mapping from signal intensity to perceived loudness observed in hearing.

The main parameter of a spectrogram is its analysis window length, which in turn determines its spectral resolution: there is an uncertainty relation between time and frequency (inevitable since they both arise from the same one-dimensional waveform, rather than being in any sense independent quantities). When we are interested in seeing voicing pitch revealed as a set of parallel sinusoidal harmonics, we must use a longer time window of 20 ms or more to give a fine (narrow-band) frequency resolution. To see fine timing detail of pitch pulses and stop bursts, we use a much shorter time window of a few ms at which point pitch harmonics blur together and disappear, but the broader spectral variation due to the formant resonances in the vocal tract remain quite visible.

Figure 9 illustrates the steps involved in converting a time waveform into a sequence of short-time Fourier transforms and assembling these into a spectrogram image. Although the first versions of this representation used analog bandpass filters, we now always calculate the spectrogram on

a computer (meaning, among other things, there is always a hard limit on the frequency range at the Nyquist rate). Since each column of pixels involves taking a Fourier transform, the computation involved in creating a spectrogram can be significant, however it is made significantly more feasible through a special, optimized algorithm for calculating the Fourier transform of discrete-time signals called the Fast Fourier Transform (FFT). The FFT manages to exploit redundancy between the values being employed when taking the inner products against sinusoids of different frequencies to reduce the computation required by a factor that actually improves as the transforms become larger (longer time windows).

## 7 Linear Prediction

Imagine we have a system and we want to know how to build a copy – an artificial system with the same properties. (This problem is sometimes called *system identification*). We could measure its gain and phase shift at a set of different frequencies (and in fact we can do this very quickly by feeding it as input the superposition of all frequencies at once, which turns out to be an impulse, the briefest possible click). We could then build a bank of bandpass filters, adjust the gains of each one to match the measured gains at each frequency, then add the outputs together again, and we'd have a system that performed much like our original. But it would be an approximation, and it would involve a very large amount of computation. By contrast, a direct implementation of the discrete-time, single-pole resonance like figure 1 requires just two multiplies per sample. If we knew that the system we were trying to duplicate consisted of only a few simple resonances, we could in theory create a more accurate and much more efficient duplicate by identifying the parameters of those resonances (their best frequencies and tuning), then implement an equivalent resonant system.

As it happens, there is an efficient and robust procedure for doing just this. In a discrete-time implementation, a resonant filter involves a few delays applied to the output signal, then feeding back these delayed outputs (with particular scaling constants) to the input. (The actual number of

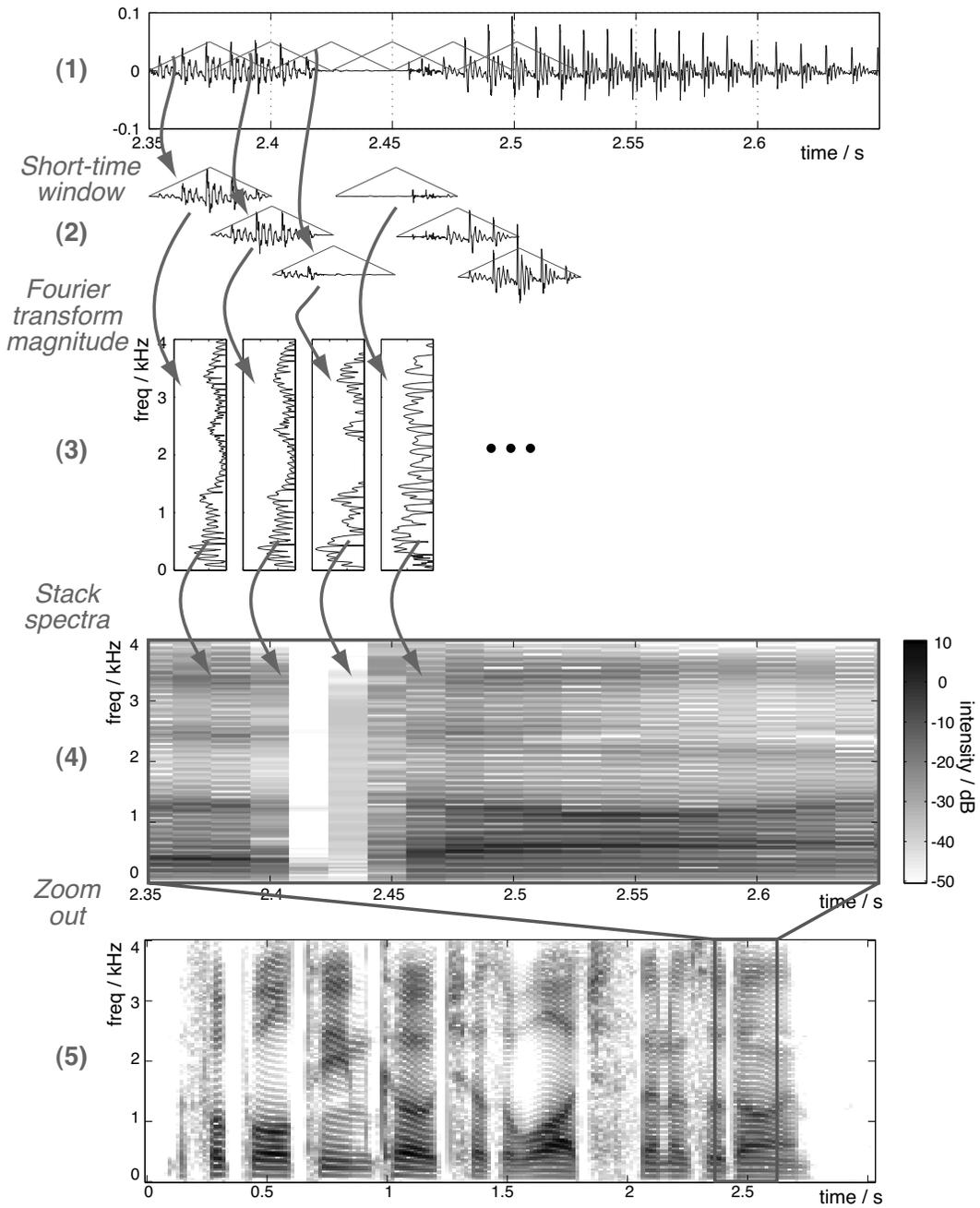


Figure 9: Calculation of the spectrogram. Input signal (1) is converted into a sequence of short excerpts by applying a sliding tapered window (2). Each short excerpt is converted to the frequency domain via the Fourier transform (3), then these individual spectra become columns in the spectrogram image (4), with each pixel's color reflecting the log-magnitude at the corresponding frequency value in the Fourier transform. After zooming out, the individual columns of pixels cannot be distinguished, and we have the appearance of a continuous image – the spectrogram (5).

delays determines the *order* of the filter i.e. how many distinct resonances it will have.) In effect, in the absence of inputs, the output at a particular time is a linear combination of a few recent output values, and the process of fitting a resonant filter to a particular signal consists of choosing the scaling constants that do the best job of matching (or predicting) each output sample from its immediate predecessors. For this reason, the technique is known as *Linear Prediction* (LP). It is such a useful and powerful technique that it turns up in various other places under names like all-pole modeling and autoregressive modeling.

The actual mathematics is a little involved, but the net result is that given only a segment of the output of a system, linear predictive analysis finds a simple resonant filter that does the best job of accounting for the spectrum of the signal being analyzed, along with the input signal (called the ‘residual’) which, when fed to the resonant system, would recreate the original signal. The approach minimizes the energy of the residual, which is achieved by making it as close as possible to a purely random (white noise) sequence – but any signal structure that cannot be explained by the resonant filter, either because the model has a lower order than the true filter, or due to the input to the original system, can be left in the residual. This makes the approach particularly robust – it does the best job it can within the limitations of the model, but it is perfectly able to approximate more complex signals and systems.

The biggest limitation of linear prediction is that it can only model systems that consist purely of resonances (poles), whereas very many systems of interest will also include zeros. No approach of comparable simplicity and power exists for modeling systems with zeros – partly because once zeros are introduced it becomes much harder to define the unique, best system to approximate any signal, since similar results can in many cases be achieved with either poles or zeros. However, there is one system of great interest that is well approximated by an all-pole model, namely the vocal tract. Apart from nasals, in which the parallel nasal path gives rise to zeros in the overall spectrum, most speech sounds are well approximated as a spectrally-flat input signal – often called the excitation *source*, but equivalent to the LP residual – being shaped by a set of resonances which are generally identified with the *filter* effected by the variable cavities of the vocal tract. In practice,

this LP source-filter model leads to usable simulations of voice sounds; it was the key to the first wave of mass-market speech synthesis applications (pioneered by the “Speak and Spell” toy in 1978), and it is at the heart of every speech compression algorithm, including GSM and other cellphone encoding schemes. In these applications, speech is broken up into short (10-30 ms) segments, which are then encoded as a single, fixed, LP filter, plus an excitation signal, which, in the popular Code-Excited Linear Prediction (CELP) scheme, is encoded as an index into a large dictionary known to both encoder and receiver. It is the relatively slowly-changing character of the physical shape of the vocal tract, and hence its acoustical properties, that allows it to be described by relatively infrequent model updates, leading to very significant data compression gains.

Figure 10 illustrates the kinds of approximations that result from LP modeling. The top trace shows the spectrum of a 30 ms segment of speech, similar to figure 7. Below are the spectra (gains as a function of frequency) for series of LP approximations for models of order 2, 4, 6, 8, 10, and 12. Each resonance actually requires two poles to be modeled, so a 12th order model (with 12 poles) can reproduce up to six resonant peaks. These relatively small LP filters cannot reproduce the spectral detail of the harmonic peaks, which are provided by the excitation in a complete system. In human voice, harmonic structure comes from the nonlinear oscillations of the vocal folds, not from the resonances of the vocal tract.

## 8 Speech Features

As we have mentioned, the goals of signal processing can be quite diverse, but they all revolve around the idea of manipulating the information content in signals to facilitate some application. One important and illuminating application is automatic speech recognition (ASR), where signal processing is involved at the very beginning to convert the raw speech signals into features that attempt to extract the information from the speech signal most relevant to recognition, while excluding (being invariant to) irrelevant information, and at the same time making the representation as small as possible, to reduce the computational burden. Speech recognition will be described in

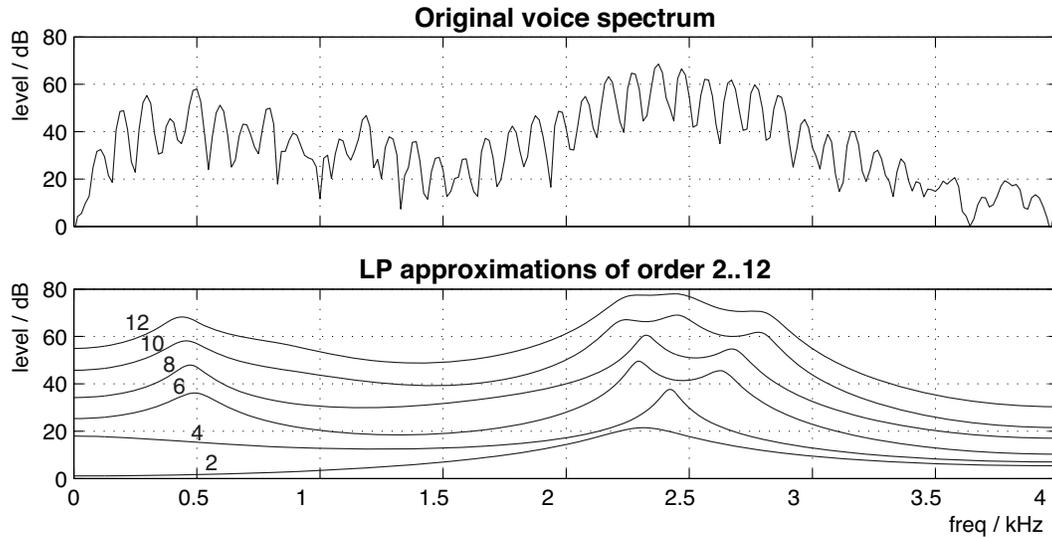


Figure 10: Examples of linear predictive models of different orders (lower pane) that are approximating the broad spectral resonances evident in a short speech fragment whose spectrum is shown in the top pane. Adjacent LP spectra are offset by 10 dB to aid visibility.

more detail in the next chapter, but here we will briefly look at the most popular speech features from a signal processing perspective.

## 8.1 Spectral features

Although it is only used directly in fairly rare circumstances, the vast majority of speech recognition features are essentially based on the spectrogram. In particular, speech is first segmented into overlapping short fragments of 20-40 ms, which are given smooth edges with a tapered window, then transformed to the frequency domain to find the magnitude of the energy in each frequency band, while discarding the phase. One reason the unmodified spectrogram is unpopular is that this is still a very large representation e.g. 256 values per frame, which just means more work and more parameters in the later pattern recognition stages. In fact, the essential challenge of speech recognition is successfully recognizing different instances of the same sound as belonging to the same class – for example, a particular vowel pronounced by different people and at different pitches.

Too much spectral detail tends only to make this generalization harder (although it might help in distinguishing two voices from one another).

## 8.2 LP features

One way to avoid capturing spectral detail that goes beyond that required simply to recognize the phone being pronounced is to fit a low-order, constrained model such as Linear Prediction. As illustrated in figure 10, a low-order LP model will capture the broad spectral shape of the sound but smooth away all the pitch harmonics – an advantage for languages like English where the actual pitch contributes very little phonetic information. By the argument that LP modeling is approximately identifying the resonances arising from the shape and status of the vocal tract (which control exactly how different speech sounds are generated), we even have a feature that directly and compactly describes the nature of the vocal tract configuration. The success of classification tasks often depends on details of how the feature values vary and how well this matches the classifier being used; the mathematical simplicity of LP models opens a wide range of alternative descriptions that carry equivalent information but which have secondary properties making them more suitable for various tasks such as classification, interpolation, and compression. LP models are somewhat vulnerable to background noise, however, since the poles will attempt to model any energy in the original signal, whether it comes from noise or interference.

## 8.3 MFCCs

The most common features used in speech recognition are the Mel-frequency cepstral coefficients or MFCCs. Let us explain these two parts separately. The Mel-frequency scale is a nonlinear mapping of the audible frequency range that was proposed in the first half of the twentieth century to account for listener's judgments about the relative distance between tones – a fixed separation on the Mel axis is supposed to result in pairs of tones that are judged as equally different in pitch. The scale is approximately linear below 1000 Hz and approximately logarithmic above 1000

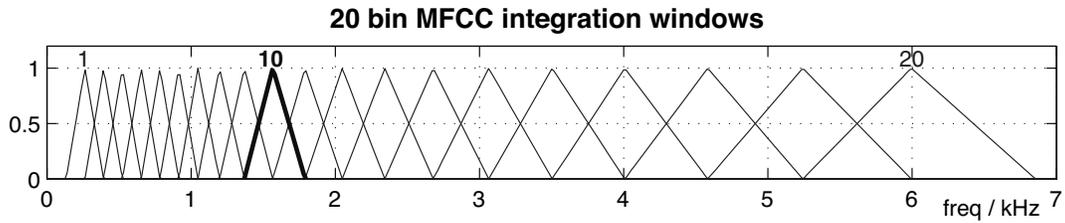


Figure 11: Weighting curves to convert a linear frequency spectrum into Mel frequency. The weighting used to construct bin 10 (the middle bin) is shown in bold as an example.

Hz, reflecting the widely-supported result that human auditory perception has a bandwidth that increases with frequency – this is even observed in the cochlea, where the resonant structures have broader and broader tuning, and wider spacing, in the higher frequencies. The consequence of this is that a conventional spectrogram, which allocates as many pixels to the spectrum between 0 and 500 Hz as it does to the 3500 to 4000 Hz range, seems to be paying too much attention to the higher frequencies at the expense of low-frequency details. There are a number of different auditory frequency scales other than Mel (including Bark), but they all share the property of expanding detail in the low frequencies and compressing it in the high frequencies. In Mel scaling, this can be done by calculating a relatively high-resolution spectrum, then combining together subsets of the frequency values using a weighted average, where the averaging occurs over a wider range of frequencies as the center frequency rises. Figure 11 illustrates the typical weighting curves used in warping a spectrum to the Mel axis: Each Mel bin combines energy with a triangular weighting scheme, spanning from the center frequencies of its two adjacent bins, with the bin spacing increasing with frequency. For instance, Mel bin number 10, the middle band in this scheme (shown bold in the figure), is composed of a weighted combination of frequencies in the range 1.4 to 1.8 kHz, with the greatest contribution coming from around 1.6 kHz.

The net result of these combinations can be seen in the top two panes of figure 12, which compare a linear frequency spectrogram with the equivalent visualization after the Mel frequency warping is applied. It is clear to see how the bottom quarter of the linear-frequency spectrogram has been expanded to fill more than the bottom half in the Mel-scaled version, and the energy above 4 kHz has been squeezed into a small band at the very top of the image, broadly reflecting

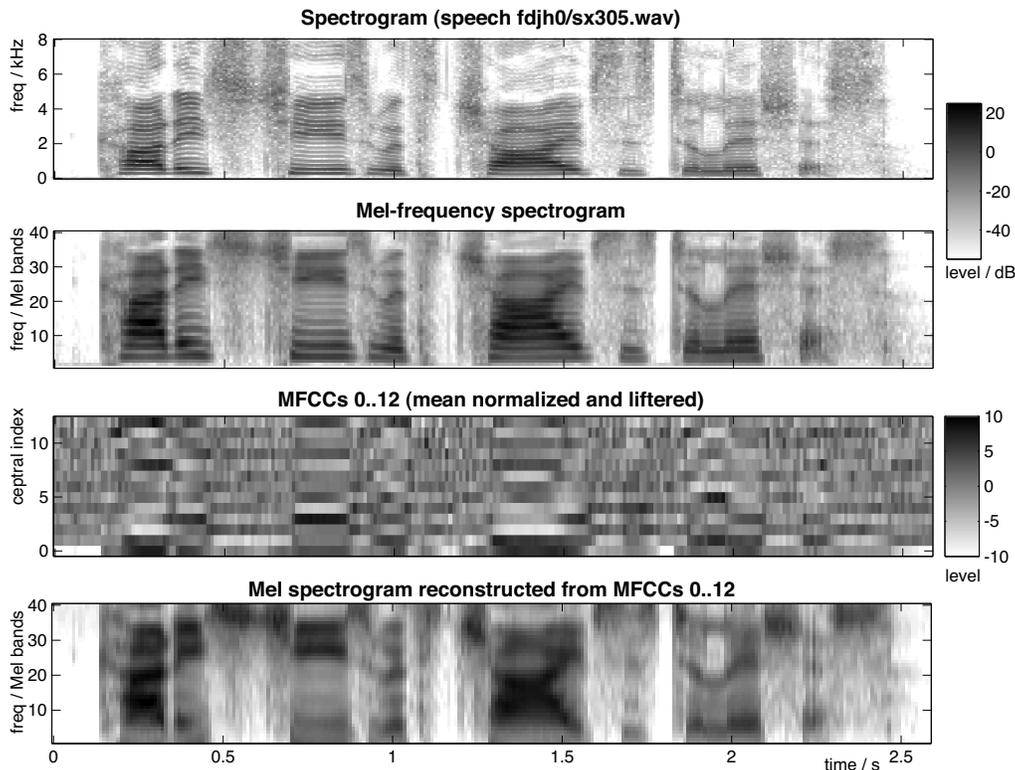


Figure 12: Mel-frequency spectra and cepstra. The top pane is a standard, linear-frequency spectrogram; in the second pane, the frequency axis has been warped to the Mel scale by combining each column of the top pane according to a 40-bin version of the weights in fig. 11. The third pane shows the first 13 values of the DCT of each column of the Mel spectrogram, which gives the MFCC features most commonly used in speech recognition. The final pane shows the effect of inverting those 13 values back into a Mel spectrogram, showing how discarding the higher-order cepstral coefficients has effectively smoothed the spectrum across frequency to remove any pitch-related information or other fine-structure detail.

its relative perceptual importance.

The second part of MFCCs is the Cepstral Coefficients. Cepstra – the name is a play on “spectra”, with the idea that the parts have been flipped around – were proposed in the late 1960s as a representation in which the complex effects of filtering on a waveform were converted into simple addition, which could then make them easier to reverse and remove. It amounts to taking a second Fourier transform on the logarithm of the magnitude of the original spectrum (Fourier transform of the time waveform). Because of the symmetry between time and frequency in the basic Fourier mathematics, without the intervening log-magnitude step, taking the Fourier transform of a Fourier

transform almost gets you back to the original signal. But taking the magnitude removes any phase (relative timing) information between different frequencies, and applying a logarithm drastically alters the balance between intense and weak components, leading to a very different signal. Cepstra can be calculated on conventional spectra, but Mel-cepstra, where the transform is applied to Mel-warped spectrum, introduces even more changes. As the third panel of figure 12 illustrates, MFCCs are not very useful for visualization, but they are extremely effective as a basis for phonetic classification in speech recognizers. This can be explained by the way that they compactly describe the broad shape of the short-time spectrum using just a few values – and that these values tend to all be relatively independent, meaning there is little redundant information in the feature vectors. These two properties – compactness and low redundancy – have large practical benefits when building pattern recognition systems, even though the information conveyed is all already present in the original linear-frequency spectrogram.

The final pane of figure 12 shows a reconstruction of the Mel spectrogram based only on the MFCCs. What we see is the smoothing effect that comes from keeping only the first 13 cepstral coefficients. This simply doesn't have enough space to fully describe all 40 Mel values, but it preserves the broad ridges and dips, and discards (smooths out) finer structure, such as the individual harmonics which are still visible at low frequencies in the original Mel spectrum. It is to be expected that this implicit smoothing is helpful to speech recognizers, since we expect the broad spectral information (i.e. the resonant peaks or formants) to be the relevant information for making phonetic classifications, and that the pitch information would only be a distraction that is better off being discarded.

## **8.4 Perceptual Linear Prediction (PLP)**

Further insight into speech features can be obtained by comparing MFCCs with an alternate, popular representation called Perceptual Linear Prediction (PLP, [Hermansky, 1990]). PLP features often perform comparably to MFCCs, although which feature is superior tends to vary from task

to task. PLP features use the Bark auditory scale, and trapezoidal (flat-topped) rather than triangular windows, to create the initial auditory spectrum. Then, rather than smoothing the auditory spectrum by keeping only the low-order cepstral coefficients, linear prediction is used to find a smooth spectrum consisting of only a few resonant peaks (typically 4 to 6) that matches the Bark-spectrum. Although the resulting linear predictor doesn't correspond to any time waveform that has been calculated, it is still possible to perform this fit using a neat piece of mathematics that finds the LP solution starting from only the magnitude-spectrum (which we have) rather than the waveform itself. Finally, this smoothed PLP spectrum is again converted to the compact, decor-related cepstral coefficients via another neat mathematical trick that finds cepstra directly from an LP model – although the same result would arise from calculating the values of the LP model's gain at regularly-spaced frequencies, converting to log, and taking the final Fourier transform.

## **8.5 Other speech feature processing**

There are two more steps commonly applied in speech recognition and that can have a significant benefit on recognition accuracy. The first is calculating “delta coefficients” i.e. an estimate of the local slope, along the direction of the time axis, for each frequency or cepstral coefficient. This means that sounds which are better characterized by changes in the speech signal than by their instantaneous characteristics – like liquids – can be better recognized in the space of a single frame if their rates of change in different frequency regions are consistent (meaning the delta features will show less spread than the direct spectrum, and will thus be easier for a pattern classifier to identify). The delta slopes are typically calculated by finding the best-fitting straight line over 50-100 ms of signal, to smooth out large variations resulting from noise and other local instability in the voice.

The second commonly-applied enhancement is some kind of normalization, most often Cepstral Mean Normalization (CMN), in which the average value of each cepstral dimension over an entire segment or utterance is subtracted from that dimension at every time step, making the time-

averaged value equal to zero. This additive offset in the cepstral domain is equivalent to an additive offset in the log-spectral domain as well (the inverse Fourier transform of the mean cepstral vector), which is itself equivalent to a constant *multiplicative* factor at each frequency in the linear frequency domain (i.e. a non-time-varying profile of gain as a function of frequency, exactly the effect of a stationary filter. Thus, CMN can largely remove the effect of a large class of fixed, linear filters that might have been applied to the speech, as might arise if the speech has been recorded by different microphones, or passed through a different channel.

## 9 Conclusions

This chapter has reviewed some aspects of signal processing, starting from a minimum of assumed background, with the aim of giving some additional insight into the properties and meaning of the signal processing operations and results most often encountered in phonetics. Without any equations, we hope to have supplied some useful, intuitive insights and explanations concerning the operations of speech signal processing. Those seeking greater detail can consult one of the excellent texts on this topic: Lathi [2002] gives a very clear, but mathematically complete, general introduction to signals and systems; Oppenheim et al. [1999] is the most authoritative reference on Digital Signal Processing in general. Finally, Gold and Morgan [2000] (which the current author is currently revising) provides an accessible, wide-ranging and entertaining overview of speech signal processing and recognition, among other topics.

## References

- Ben Gold and Nelson Morgan. *Speech and Audio Signal Processing*. John Wiley and Sons, New York, 2000. ISBN 0471351547.
- H. Hermansky. Perceptual linear predictive (plp) analysis for speech. *The Journal of the Acoustical Society of America*, 87:1738–1752, 1990.

B.P. Lathi. *Signal Processing and Linear Systems*. Oxford University Press, 2002. ISBN 0195219171.

A.V. Oppenheim, R.W. Schaffer, and J.R. Buck. *Discrete-time signal processing (2nd ed.)*. Prentice Hall, 1999. ISBN 0137549202.