Chapter for:
Hardcastle, W. J., Laver, J. (eds.),
***A Handbook of Phonetic Science*.**
Oxford: Blackwell.

Speech signal processing

Johan Liljencrants,
Speech Communication and Music Acoustics,
Royal Institute of Technology, Stockholm

0. Introduction

The speech signal is normally picked up as an analog electrical representation of the acoustic sound pressure as sensed by a microphone which can be analyzed, amplified, transmitted, or recorded using whatever kind of device is appropriate. Historically all processing like filtering, coding, analysis, and synthesis was done with analog devices, theoretically conceived and operating with continuous quantities like time, voltage, frequency, etc. In consequence of the technical evolution most processing today is more conveniently done with digital computing, in appliances for various purposes by special signal processors, in the laboratory often with personal or larger computers. This has allowed for an ever growing system complexity that could never realistically be implemented with analog systems. Also modern work often adopts signal processing tools and methods in higher level modelling such that the concept of a signal is wider than perhaps suggested by its naive original meaning. It is for instance commonplace to regard system parameters like formant frequencies as just another set of signals.

Digital signal processing (DSP) has expanded tremendously as a field of its own during the last few decades with important contributions from several diverse research disciplines, particularly those of speech and communications, statistics, and seismology not to mention mathematics. There is an abundant handbook literature on DSP, more often than not going into great mathematical detail, but the use of it is sometimes complicated by variant, but largely synonymous terminology and conventions that reflect the field background of the authors. This chapter gives a cursory presentation of some frequently used DSP concepts and applications that should be familiar to all speech workers. Formulas and examples of flow diagrams are shown for a few standard basic procedures, the main purpose being to assist the reader to identify them as such in the literature on theoretical developments and applications. The approach is directed into the domain of discrete signals rather than into the historic roots within classical continuous theory.

1. The discrete Fourier transform

A signal can be completely quantified in either of the two domains of time and of frequency. The classical invention of Fourier was to consider a signal to be constructed from a number of sinusoidally shaped components. For a periodic signal, one that repeats a pattern with a period *T0*, he showed that it can be represented by a fundamental of frequency 1/*T0* and a number of *harmonics*, all multiples of this frequency. The amplitudes and phases of these components constitute the amplitude and phase *spectrum* of the signal. This spectrum is discrete, it has spectral lines at the frequency intervals $\Delta f = 1/T0$, but nothing between them. The spectrum is a prescription of how much to take of each frequency component in order to synthesize the time signal.

Let us assume our signal has no extreme temporal fine structure, such that the spectrum has a limited number of harmonics. One of these frequency components has the index number k, it represents the frequency $k \cdot \Delta f = k/T0$ and may be denoted with a *complex* number X_k. The use of such a number is no more than a practical convention and an expedient to keep formulas simple. It can be expanded alternate equivalent ways, for instance in terms of real and imaginary parts A_k and B_k, or magnitude $|X_k|$ and phase ϕ_k which for purposes like graphic plotting may be more appealing.

$X_k = A_k + j\,B_k = |X_k|\, e^{\,j\phi k}$

The formula to construct a sequence of N time samples x_n for one period is called the Inverse Discrete Fourier Transform, IDFT:

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{\,j(2\pi / N)\,kn} \quad ; \qquad n = 0 \ldots N\text{-}1 \qquad (1)$$

This compact mathematical notation should not hide the fact that there is a considerable quantity of computation involved; the formula is to be computed N times, once for every time sample x_n. And every such sample is built up as the sum of contributions from N different X_k. The second factor in each summed term is the sinusoidally shaped elementary function which can likewise be expanded as

$$e^{\,j(2\pi/N)\,kn} = \sin((2\pi/N)\,kn) + j\cos((2\pi/N)\,kn) \qquad (2)$$

For a transform to be interesting it is of course required that it is invertible, we must be able to analyze a signal x to compute this prescription X_k. This amounts to solve X_k from a given set of x_n in the system (1) of N equations, and this mere procedure explains why we use N time samples x_n. The result is

$$X_k = \sum_{n=0}^{N-1} x_n e^{\,-\,j(2\pi / N)kn}; \qquad k = 0 \ldots N\text{-}1 \qquad (3)$$

in this notation known as the Discrete Fourier Transform, DFT. That the direct and inverse transforms are so similar is because the elementary shapes are *orthogonal*, that is, if you sum the product of two harmonics over the range N, then the result is zero unless the harmonics are of the same frequency.

The DFT exhibits a number of symmetries that come inherently from the sine and cosine shapes. A most important one is that if the time signal x_n is real (no imaginary part, as with all physical signals), then the spectrum is symmetrical such that $X_k = X^*_{-k}$, that is, the real part of the spectrum as well as its magnitude have even symmetry, $\mathrm{Re}\{X_k\} = \mathrm{Re}\{X_{-k}\}$, and the imaginary part and the phase have odd symmetry, $\mathrm{Im}\{X_k\} = -\mathrm{Im}\{X_{-k}\}$. In the same manner as the sequence X_k of frequency samples represent the spectrum of a periodic signal now the sequence x_n of time samples have a periodic spectrum, and we need to use only one of these spectrum periods; all the other are called *aliases*. In computing it is general practise to use positive subscripts only, so the convention is to use X_0 to X_{N-1} and with the symmetry point at $X_{N/2}$. The symmetry point

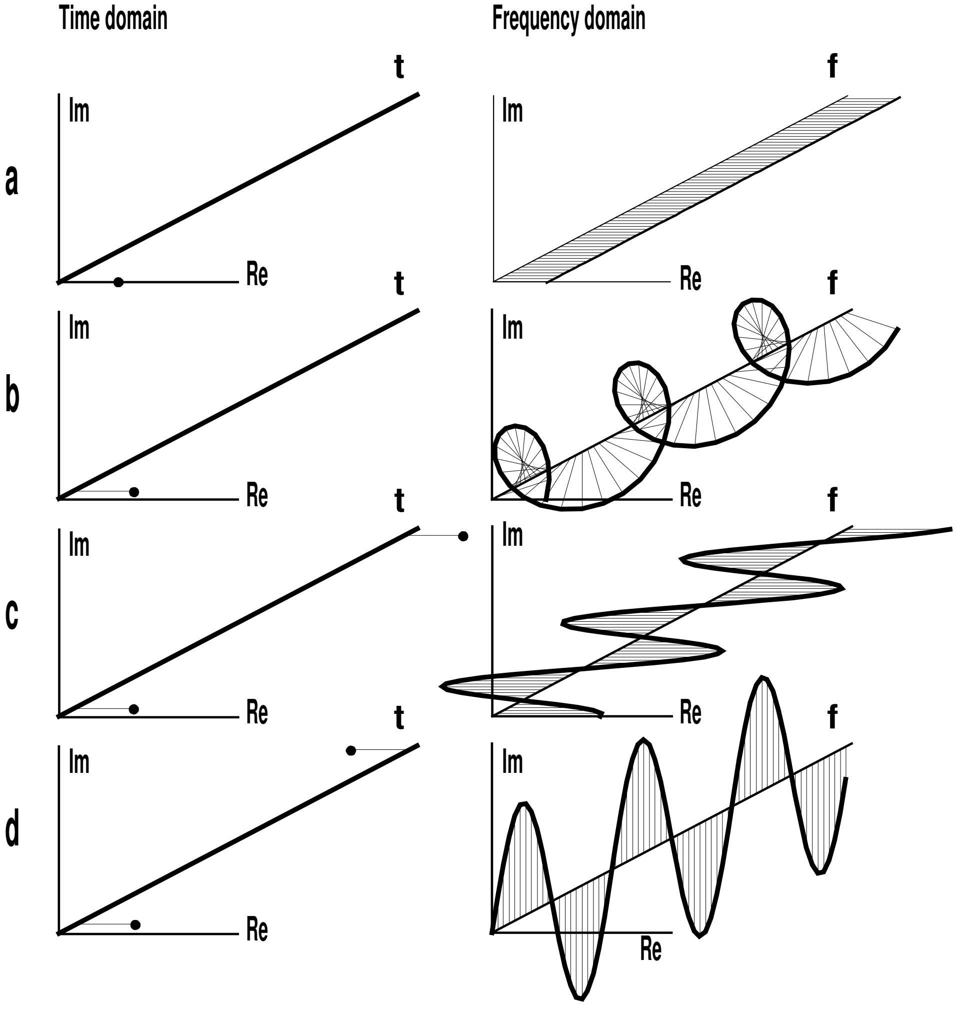

Fig 1. Prototype examples of three-dimensional time and frequency representations linked by the discrete Fourier transform. The signal real and imaginary compinents are shown horizontally and vertically, while the time and frequency axes extend away. a: a unit impulse with a white spectrum, b: delayed impulse gives phase increasing with frequency, c: even pulse pair gives real spectrum, d: odd pulse pair gives imaginary spectrum. The two domains can be interchanged when allowing for a scale factor.

should properly have been at zero frequency, but this is no problem in the discrete world since $X_{-k} = X_{N-k}$ in the next alias, identically.

Fig 1 shows a number of prototype examples of DFT time-frequency representation pairs. In a and b the time sequence is a unit impulse, x_n is zero for all n except at one place in each, at $n = 0$ and 3 respectively, and $N = 64$. The sequences are plotted in a 3D coordinate system of real and imaginary components vs. time or frequency indices respectively. This way you can also see the result in the form of magnitude and phase, displayed as lengths and inclinations of the thin lines extending from the frequency axis. When the pulse advances in time we observe that the magnitude of the DFT is constantly the same unit value, but the phase increases more rapidly with frequency, an example of the *delay theorem*. When the impulse advances one step, then the total frequency sequence will contain exactly one more revolution in phase. The ends of the sequence always meet. A programmer must be aware there is no X_{64} in fig 1. That sample would be number 0 in the next alias, identical to X_0.

Further illustration of some symmetries is given in fig 1c. Adding an equal pulse in number 64-3=61 makes the time series even (and still real), and the transform is even and real. The transform is the sum of two spirals like in b, equal in magnitude and pitch, but opposite in direction of phase rotation. Similarly making an odd time series in d gives an odd and imaginary transform.

In the figures 1c and d we can also for a moment switch in order to let the right hand graph represent time and the left hand represent frequency. We then see how the transform of a cosinusoid comes out as a pair of equal spectral lines at plus and minus the appropriate frequency. Changing the phase of the signal to make it a sinusoid, causes it to show up as a half revolution phase shift and an odd and real transform.

If we increase the frequency of a sinusoidal signal such that it contains one more cycle within the N samples, then the spectral line appears at the next sample in the frequency interval. Now, what happens is we transform a sinusoidal signal that does *not* have an integer number of periods within the time interval? The spectrum should perhaps then ideally be a line at some frequency that is not represented by any sample.

Remembering that the discrete spectrum we got represents the finite time signal repeated periodically, then when we join successive N sample time segments having a non-integer number of sine periods we will get a discontinuity at every joint. This will be seen in our spectrum as strong components extending over the entire frequency range.

The remedy is to modify the input prior to the transformation with a suitable *window* that has a gradual fall off toward the ends to reduce the discontinuity. Some well known windows and their transforms are shown in fig 2. The *Hanning* window is an inverted cosine period, raised to give zero values at its ends. The *Hamming* window has the same basic shape, but is raised on a small pedestal. This is optimized such that the transform sidelobes are approximately the same level everywhere, about 43 dB below the main lobe. For even more stringent sidelobe requirements the Blackman or Kaiser windows can be used. Within the dynamic range shown they are similar to the truncated Gaussian shape at bottom of the figure, and which has a special interest since a Gaussian remains a Gaussian when transformed.

The window will impose an effective duration T_e for which the spectrum analysis is valid. This time is shorter than the total duration of the window, by a factor of 1, 2, and 3 in the cases of fig 2. In the frequency domain what would ideally have been a single spectral line will be spread out as the shape of the window transform. The shown examples were scaled such that the width of the main lobes in the frequency domain are about equivalent. They illustrate the important rule of thumb that the effective resolution bandwidth in the spectrum, the effective bandwidth of the window main lobe at its -3dB points (approximately), is

$$B_e = 1/T_e \qquad (4)$$

The reason for windowing is thus to suppress artificial components that would otherwise arise from the abrupt truncation of the time interval to NT. The cost is that the frequency resolution is impaired. Or conversely, to keep a prescribed frequency

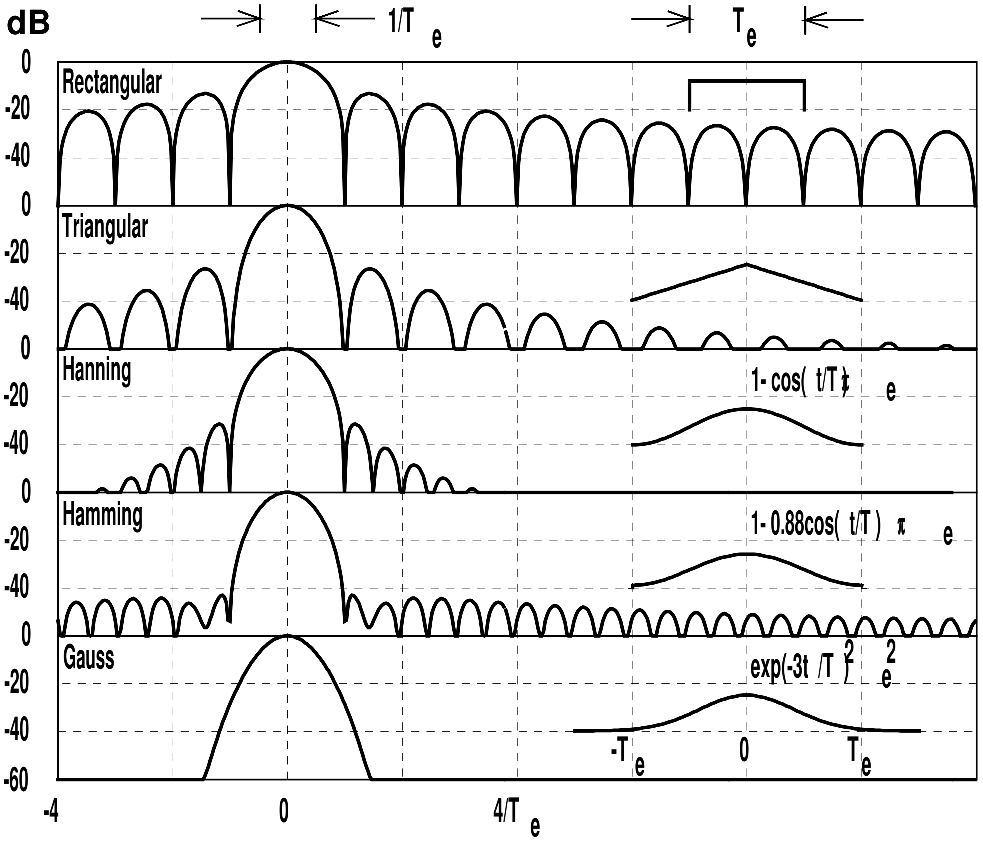

Fig 2. Common window functions (inserts) and their transforms.

resolution we must select a greater *NT* to accommodate a total window duration that is some moderate factor longer than its effective duration.

The fast Fourier transform algorithm

The amount of computation to perform the transform by the definition (3) is N^2 complex multiplications and additions, an amount that may be inordinate for larger *N*. The Fast Fourier Transform, FFT, is an elegant algorithm that optimizes an order in which the partial computations are performed, the final result is exactly the same as from the definition. FFT stands for the feature of computational speed in making the DFT, it is not a transform in itself as is DFT. The algorithm was published by Cooley and Tukey (1965) and since remains a prime standard tool in DSP making countless old and new developments practically usable. The total computational effort for FFT is proportional to $(N/2)*{}^2\log(N)$, where *N* is a power of 2, instead of N^2 as would be for the definition formula. For instance, with N=1024 the reduction of computation is about 200-fold. Another merit of the FFT is that the accuracy of the result is improved because there are fewer rounding errors accumulated.

Other transforms

Having entered the domains of complex numbers and with Fourier derivations still valid it is no wide step to generalize the frequency variable from an imaginary $j\omega$ to a general complex quantity $s=\sigma+j\omega$, also including a real part which means the elementary function is a sinusoid which is exponentially increasing or decreasing as $e^{\sigma t}$. The continuous transform using this generalized frequency goes under the name of Laplace, and its discrete correspondent is in practice easily embraceable by the DFT.

The sinusoidal elementary building block is not the only one from which we can build up a world of signals. In fact any shape can be used, orthogonal shapes are preferred, but the Fourier method is standard for reasons of conceptual and mathematical simplicity as well as arithmetic efficiency. Over the times mathematicians have devised many other transforms, all with the same basic concept of building the time and/or the frequency representation from variants of an elementary shape, but differing in what exactly is this shape.

The last decade has seen a vivid activity in applying the concept of time-frequency distributions to speech analysis as well as other disciplines. This alternate development for the study of time-varying spectra originated as probability distributions in quantum physics, but its mathematical formalism can be used for power estimation. The best known of several such distributions, see for instance the review by Cohen (1989), is perhaps the Wigner-Ville distribution, WVD. This makes use of the *analytic signal* known from signal theory and which can be concisely described as the complex signal that arises when you filter away negative frequencies from a real physical signal, but retain the positive, an operation easily implemented by use of the DFT. Doing this involves a ´Hilbert transformer' which is a phase shifter in the time domain. This device is a kind of filter rather than a transform between domains, so its name can cause confusion.

2. Analog to digital conversion

We must be able to represent continuous analog signals as sequences of discrete numbers in order to treat them with digital devices. This process, the *analog to digital conversion*, A/D or ADC, can be broken down into two stages, the *sampling* and the *quantization*.

The sampling means that we measure a continuous signal at some specified intervals in time, and neglect what values the signal may have between those samples. Normally (but not necessarily) the time interval T between samples is constant such that we can define a *sampling rate* $f_s=1/T$. In discrete theory the term rate is preferred to frequency which rather belongs in continuous theory, but in general usage they are mostly treated as synonyms. As just outlined, when we take N time samples, in all covering the time span $T_0=NT$, then we also get N frequency samples, covering the frequency span $N/T_0=1/T$.

One point of essence is then that on our frequency axis with indices k the sampling rate is located at $k=N$. Another is that the symmetry requirement permits us to have only $N/2$ unique frequency samples, the other $N/2$ must be mirror values if the spectrum is to represent a real physical signal. This leads to one cornerstone of sampling theory: we can correctly handle only such signals that have no frequency components above half the sampling rate. Should there be any such higher frequency components their aliases will superimpose on those in our permitted range below $f_s/2$ and cause irreparable damage known as *aliasing distortion*. Therefore it is mandatory that an analog signal is band limited to $f_s/2$ by a pre-filter before it is sampled. The minimum sampling rate f_s is called the *Nyquist rate*, in some literature the term is unfortunately used for maximum allowable signal frequency $f_s/2$.

The other cornerstone of the sampling theorem is that we can reconstruct the original continuous signal exactly without loss of information, namely if we send the sampled signal through a lowpass (ideal) filter having its cut-off frequency at $f_s/2$. Then we will retrieve the base spectrum alone and remove all the higher frequency aliases. The reconstruction lowpass filter generates the missing signal shape between the samples and implements a special way of interpolation. - Suppose the original signal indeed had some wiggles between the sampling points that are not present in the reconstruction. In that case it must also have had some too high frequency components for the sampling rate actually used.

The second stage of A/D conversion is to represent the continuous range of sample values on a numerically quantized scale with a certain number of steps, we round off the continuous value to the nearest step. Each sample is assigned a numerical code for its value, and the accuracy depends on how many digits we care to use for this code which is normally binary. If we for instance use an 8 bit code, about the minimum to be practically usable, then the code can represent 2^8=256 different steps on the scale. Through this incomplete description we introduce an error with a peak value of half a step which manifests as a pseudo random *quantization noise* superimposed on our signal. It we compare the maximum representable signal amplitude (256/2) to this noise amplitude we find a coarse approximation to the signal to noise ratio as 48 dB. A rule of thumb says we can expect about 6 dB of signal to noise ratio for each bit in the quantization; 10 bit quantizing would give 60 dB, and 16 bits 96 dB.

A quantizing scale with equal steps as with conventional A/D converters makes a noise background of constant level, irrespective of signal level. In speech coding for use in telephony 8 bit quantizers are normal, but where instead the steps on the scale are of unequal size, small steps for small signal amplitudes, and gradually larger steps with increasing amplitudes. There exist two slightly different international standards recognized as A-law and µ-law. These schemes render a signal to noise ratio that is only about 38 dB, but instead of being constant the noise background essentially follows at this distance below the actual signal level.

The development of consumer equipment for digital sound like CD and DAT includes a dramatic improvement of performance versus cost in ADC technology. A prime contributor is high speed circuitry which makes it possible to sample the signal at a much higher rate (several MHz) than needed from bandwidth considerations with the signals actually present. This means that lowpass filters for bandlimiting and reconstruction can be omitted at the analog side. Another is the use of so called delta-sigma quantizers that include a filtered feedback from the discrete to the continuous side with an effect to shape the spectrum of the quantizing noise. The total noise power is unaffected, but is concentrated at high frequencies such that its low frequency components are suppressed. Not until now the bandlimiting lowpass filter, implemented as a digital filter with a performance close to ideal is introduced. The filter not only band limits the desired signal, but also removes a major part of the quantizer noise. This scheme is today perfected with such extreme noise suppression that precision conversion is attained with only a 1-bit sign detector toward the analog side. Finally most of the output samples are discarded (if ever computed), we only keep those (for instance every 128th) necessary to make the resultant sampling rate at least twice the band limit.

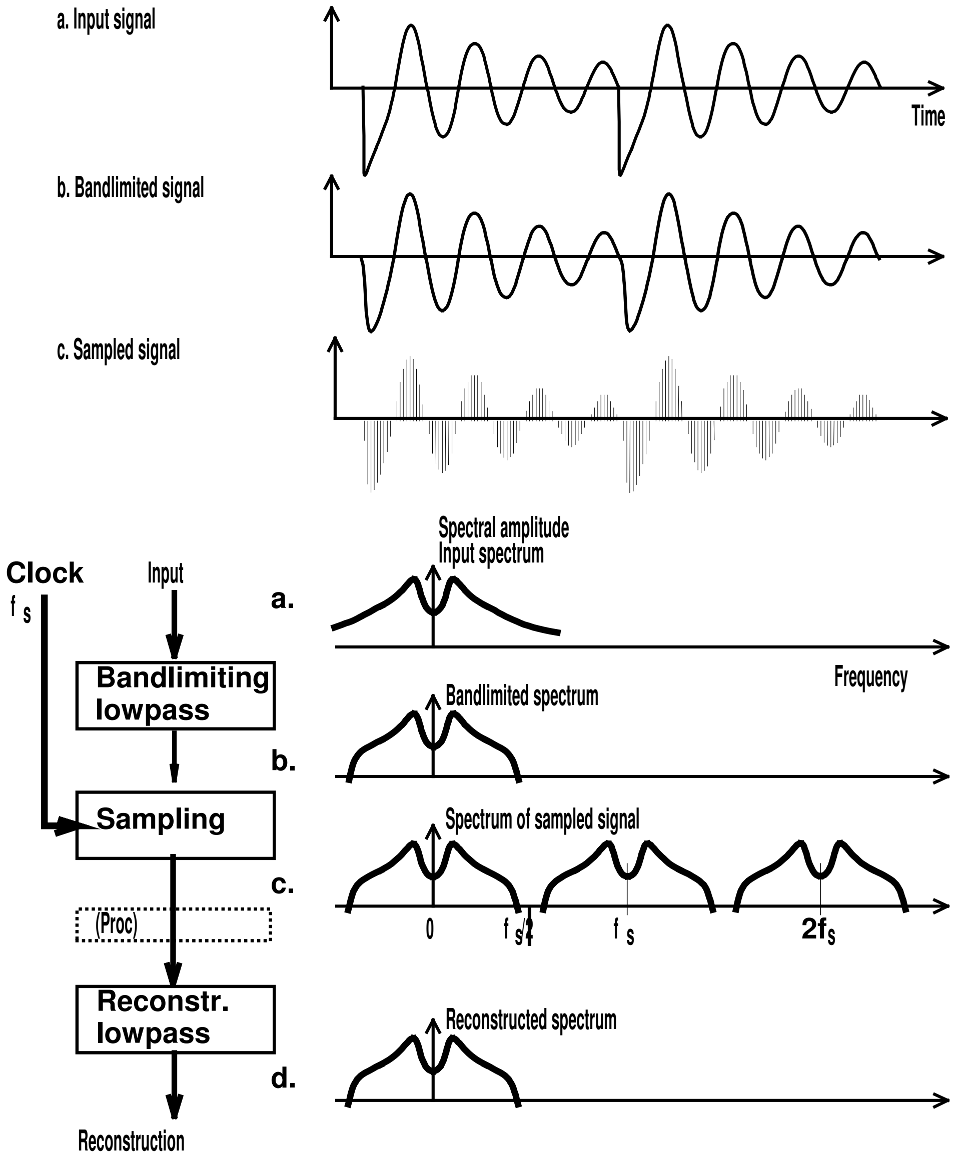

Fig 3. The stages in the sampling and reconstruction of an analog signal, top: time waveforms, bottom left: block diagram, bottom right: spectra.

In sound engineering 48 or 44.1 kHz sampling are standard. For speech work 20 kHz is sufficient for most practical purposes, but lower rates like 16, 12, or 8 kHz are commonly used, the latter for speech communications. With the lower rates part of fricative speech sounds can of course not be registered. It should be recognized that a high sampling rate is not automatically beneficial. An excessive frequency range will not only increase the volume of computation but may at instances also be incompatible with model validity.

3. Filtering

Classically a filter is considered in the frequency domain as a device by which we modify the spectrum $X(f)$ of a signal. The filter is then characterized by a *transfer function* $B(f)$ by which we multiply the input to obtain the output

$$Y(f)=X(f)\bullet B(f). \qquad (5)$$

A lowpass filter, for instance, has a magnitude of $B(f)$ close to unity below its cutoff frequency, and a small value above the cutoff. $B(f)$ is a complex function and can be represented with its real and imaginary parts, but mostly one prefers to show it in terms of magnitude and phase.

A conventional analog filter is *causal*, it gives no response at its output at times before any input has reached it. It is noteworthy that this restriction is in a practical sense often somewhat relieved in digital processing. If the total system is anyway set up for a certain delay between input and output we may access a number of 'future' input signal samples in store, relating to times later than that for which we are currently computing the filter output.

It is illuminating to study a filter starting from the unit impulse as a prototype input signal. In the discrete time domain let the impulse have its unit value at sample number zero, and be zero everywhere else. The spectrum of this impulse $X(f)=1$ for all frequencies. The corresponding output (the *frequency response*) of the filter is then just $B(f)$ and we can regard this as a special kind of signal that represents the filter itself and nothing else. If we now transform this signal into the time domain we get the filter time response, or *impulse response* $b(t)$.

This gives an opening to examine one particular class of digital filters. We start with a known or prescribed impulse response manifest in a table of regularly spaced samples b_k. We then use these values to set up a number of multipliers in a device like in fig 4a. Here an input pulse will travel along a chain of delay units, each with a delay equal to the sampling interval T. The signal is tapped between the delay units, multiplied with the appropriate b_k, and forwarded via a summing unit to the output. If the input is a single impulse, then when it travels down the delay chain it is present at only one tap at a time and the output will take the corresponding value and thus reproduce the impulse response in sequence. And when the input is a sequence x_n of a signal the output will be the sum of all the weighted and delayed samples. The general time domain notation for the filter is

$$y_n = \sum_{k=0}^{K} b_k x_{n-k} \tag{6}$$

This type of filter is called a *transversal* filter, or commonly a *FIR* filter, for *finite impulse response*, which emphasizes that in a practical implementation the delay chain must have a finite length. The output being a weighted average of $K+1$ samples moving along with time n lies behind the term *moving average* (*MA*) filter in statistics terminology.

It also illustrates the kind of processing called *convolution* between b and x to find y. An important aspect is that to do the same operation in the frequency domain is just the simple multiplication of (5). And by the DFT/IDFT symmetry, multiplication in the time domain is equivalent to convolution in the frequency domain. The windowing of a time signal is a prominent example - the spectrum after transformation is the convolution between the signal and window transforms.

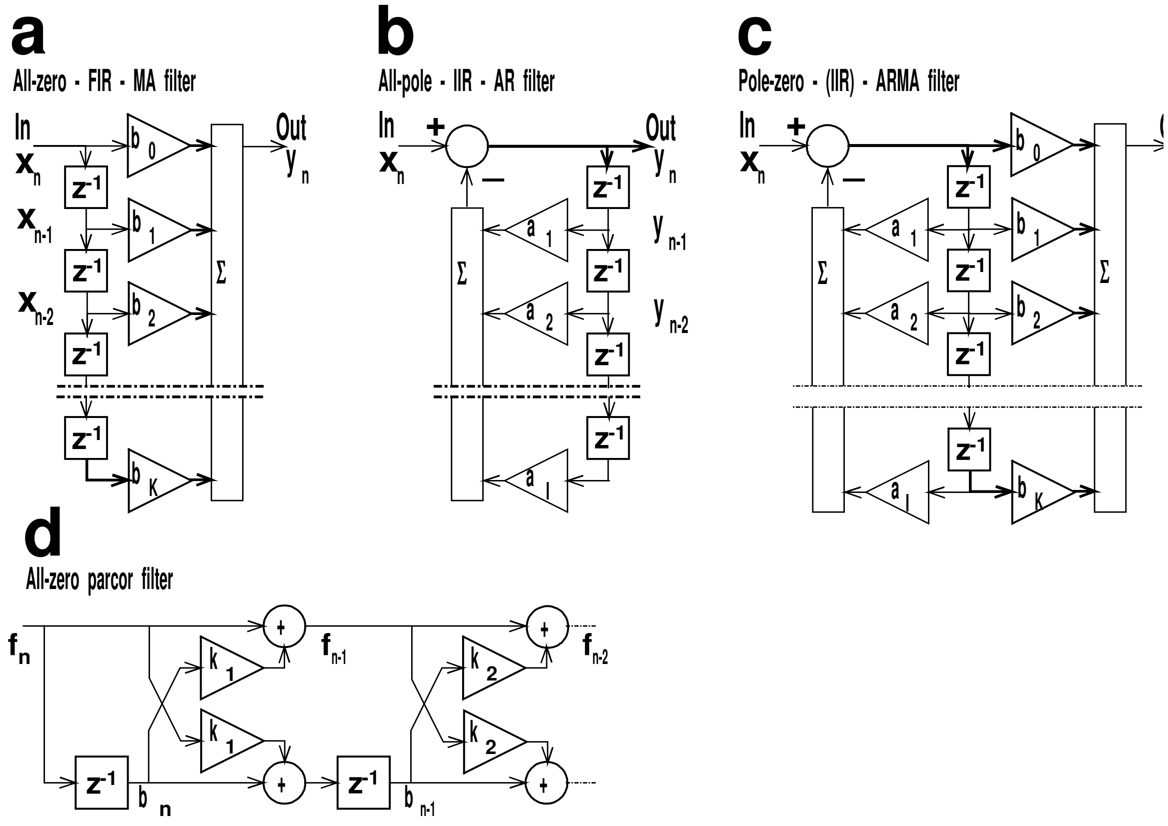

Fig 4. Block diagram representations of the computation formulas for elementary filter types:
a: All-zero - FIR - MA filter,
b: All-pole - IIR - AR filter,
c: Pole-zero - ARMA filter,
d: All-zero parcor filter, one example among several possible lattice filter structures.

As a simplistic example of how to design a FIR filter from a prototype numeric filter response specification in the frequency domain we can obtain the impulse response sequence b_k by IDFT. It is necessary however to understand exactly what we are allowed to specify. Obviously b_k must be real numbers which implies that the frequency specification must have even symmetry from $-f_s/2$ to $+f_s/2$, or 0 to f_s. This frequency band is uniformly sampled at N intervals, so we can not specify any details in the response with any better resolution than f_s/N. This implies that sharp frequency filtering requires such a large N that FIR filters sometimes are impractical for implementation in computers or signal processors. It is also important that the b_k sequence ends gracefully, for instance by the appliction of some window function. Design methods for FIR often become iterative for this reason, after windowing the frequency response is modified and must then be rechecked, conveniently with a DFT of the b_k.

The z transform

A central concept in DSP is the z transform. In discrete time systems this has a meaning correspondent to the Laplace transform in continuous systems. z is defined in complex frequency s and sample interval T by

$$z = e^{sT} = e^{\sigma T} \cdot e^{j\omega T} \quad or \quad z^{-1} = e^{-sT} \qquad (7)$$

This constitutes a conformal mapping of every point in the *s* plane onto the *z* plane. A main feature is that the imaginary *s* axis, representing frequencies $j\omega$, is mapped on the unit circle in the *z* plane, fig 5. The left half *s* plane, the allowed area for the poles of a filter if it is to be stable, is mapped to the interior of the circle. Conversely the *z* plane is mapped on an infinite sequence of horizontal stripes in the *s* plane, representing the baseband and its aliases due to the sampling.

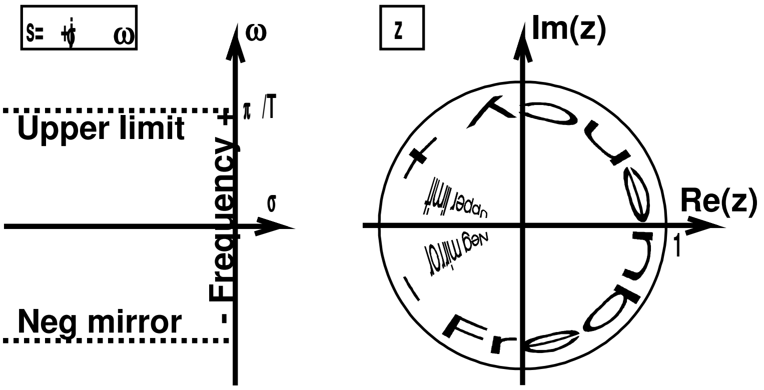

Fig 5. The conformal mapping of the complex frequency plane s, below half the sampling rate, and the z plane.

By virtue of the delay theorem z^{-1} can be seen as a *delay operator* that signifies a delay with the sampling interval time *T*. Each term in (6) can then be transformed as

$$b_k \, x_{n-k} \Rightarrow b_k \, X \, z^{-k} \qquad (8)$$

such that the *z* transform of the time series expression (6) can simply be seen as

$$Y(z) = X(z) \sum_{k=0}^{K} b_k \, z^{-k} \qquad (9)$$

where thus the sum is the *z* domain transfer function *B*(*z*)=*Y*(*z*)/*X*(*z*).

This is an *K*th order polynomial in *z* with the corresponding number of coefficients b_k. One of the things you can do with a polynomial is to set it to zero and solve the resulting equation. Then you get *K* (generally pairwise complex conjugate) values for *z*, and for each of them the transfer function is zero, they are referred to as the *zeroes* of the transfer function. Each value of *z* is represented by equivalent *s* domain values in the base band below half the sampling rate, and an infinite number of higher aliases.

In working with discrete systems it is often helpful to forget about the time domain and instead use the *lag* (z^{-1}) domain. Correspondingly the *z* domain is used as an, albeit distorted, replacement for the frequency domain. When you ultimately want some result expressed in frequency it is easy enough to compute that from *z* using the definition (7).

A very important trick that can be done with any kind of filter or other signal transmitting device is to connect it in a feedback loop as in fig 4b. Doing this with our FIR device the configuration renders a new transfer function which is simply found by inspection as

$$A(z) = Y(z)/X(z) = 1/(1+B(z)) \qquad (10)$$

We now instead use coefficients a_i to distinguish the feedback case, and the convention is to put $a_0 = 1$ to settle a scale factor such that we can write

$$1/A(z) = 1 + \sum_{i=1}^{I} a_i \, z^{-i} \tag{11}$$

We then again have an expression with a polynomial with the important distinction that it is in the denominator of $A(z)$. Again solving for the roots, the values of z where the denominator polynomial is zero, we now get the *poles* of the new transfer function $A(z)$.

A specific consequence of the feedback mechanism is that once a signal sample has entered the system it will circulate through the network back to the input and generate new output samples for all future. This lies behind the term *infinite impulse response*, *IIR*, for this class of filters. This feedback device alone, with no zeroes in the numerator, is also called an *all-pole* filter, or in statistics terminology, an *AR* filter, then with reference to that its coefficients may be established by use of autoregressive methods.

Any filter can be modelled in terms of the two prototypes FIR and IIR in combination. The general recursion formula on how to compute the output samples y_n from the input samples x_n is then

$$y_n = \sum_{k=0}^{K} b_k \, x_{n-k} \; - \sum_{i=1}^{I} a_i \, y_{n-i} \tag{12}$$

visualized in fig 4c. In the z domain, the filter transfer function $H(z)=Y(z)/X(z)$ is seen from

$$Y(z) \bullet (1 + \sum_{i=1}^{I} a_i \, z^{-i}) = X(z) \bullet (\sum_{k=0}^{K} b_k \, z^{-k}) \tag{13}$$

The AR part of the filter gives I complex conjugate poles and the MA part gives K complex conjugate zeroes. The higher of I and K define the *order* of this pole-zero, or ARMA filter.

Even if we can implement any filter directly with two such polynomials there is often a practical reason to refine the technique. Especially with higher order systems the coefficients may need to be specified with an accuracy that can not be reached in processors with moderate word length. This can be overcome by various manipulations on the formulas to write them in alternative, but equivalent forms.

One way is to solve the polynomials for their roots. This may be a costly operation with high order systems since it must then be done iteratively. Once the roots are found the polynomial can be factored as the product of a number of first and second order polynomials. In the implementation this corresponds to that number of such low order filters, connected in sequence such that the output of one is input to the next. An example is the *cascade* formant speech synthesizer where each formant is implemented with a second order filter. Knowing the roots the polynomial can alternatively be expanded into a sum of partial fractions and this corresponds to a set of *parallel* filters, another speech synthesizer classic. Here all the filters are given the same input and the total output is the sum of the filter outputs. A more recent and technically advantageous development is the *lattice filter* structure to which we return below.

A sampled-data filter can be developed from well known templates in continuous theory, like the *Butterworth* (maximally flat frequency response) all-pole filters, and *elliptic filters* having poles and zeroes combined to render sharp filter cutoff between the pass and rejection frequency bands. Such descriptions in terms of poles and zeroes do not always perform well when transposed to the z domain because of interaction with the frequency aliases. One popular remedy is to pre-warp the frequency description with a *bilinear* transformation such that infinite frequency is mapped to $f_s/2$, that is, $z=-1$. In the handbook literature there are also numerous other methods on how to design the filter coefficients from given specifications in time or in frequency, some available as commercial programs.

4. Spectrum analysis

Spectrum analysis has always been a fundamental tool for description and parameter extraction in speech research. Two basic representations are predominant, the *spectrum section*, fig 6, that pertains to a specified time interval and shows level versus frequency, and the *spectrogram*, fig 7, with frequency versus time and the level portrayed as shade of gray or color. The standard method of spectrography is to multiply the signal with a time window of suitable length and shape, Fourier transform this, and finally find the spectral level by the logarithm of the squared magnitude. To make a spectrogram this procedure is repeated with partly overlapping windows until the desired total time is covered. The most central issue in spectrography is to select the parameters of time resolution and frequency resolution. From the spectrogram we can localize an event timewise with a resolution that corresponds to the duration of the analysis window. The frequency resolution is inversely related to this duration as defined in (4). These resolutions define a 'logon' within which no further detail can be found. Classical selections for speech work from the time of the Sonagraph are 'narrow band' to resolve the voice harmonics, and 'wide band' to suppress these and instead reveal the formant structure and the timing details. Examples in fig 7a and c show 4ms*250Hz and 20ms*50Hz.

The essence of the now fashionable time-frequency distributions is that they, contrasting with the classical spectrogram, reveal time detail within the time span transformed. This has a price in that signals, more composite than for instance sine sweeps, exhibit what could be called intermodulation products or aliases, spurious peaks that make the distribution difficult to interpret. This effect can be moderated by various schemes of smoothing which however counteracts the purported improvement in resolution. Fig 7d shows results from an implementation with individually controllable time and frequency smoothing windows as suggested by Velez and Absher (1989). Here the smoothing is selected to make the spurious components barely visible. An example with less smoothing in fig 7e shows typical spurious patters and alias 'ghost' formants and pitch periods. The use of time-frequency distributions is promising but not yet established in speech. As Cohen remarks: Although it is now fashionable to say that the motivation for this approach is to improve on the spectrogram, it is historically clear that the main motivation was for a fundamental analysis and a clarification of the physical and mathematical ideas needed to understand what a time-varying spectrum is.

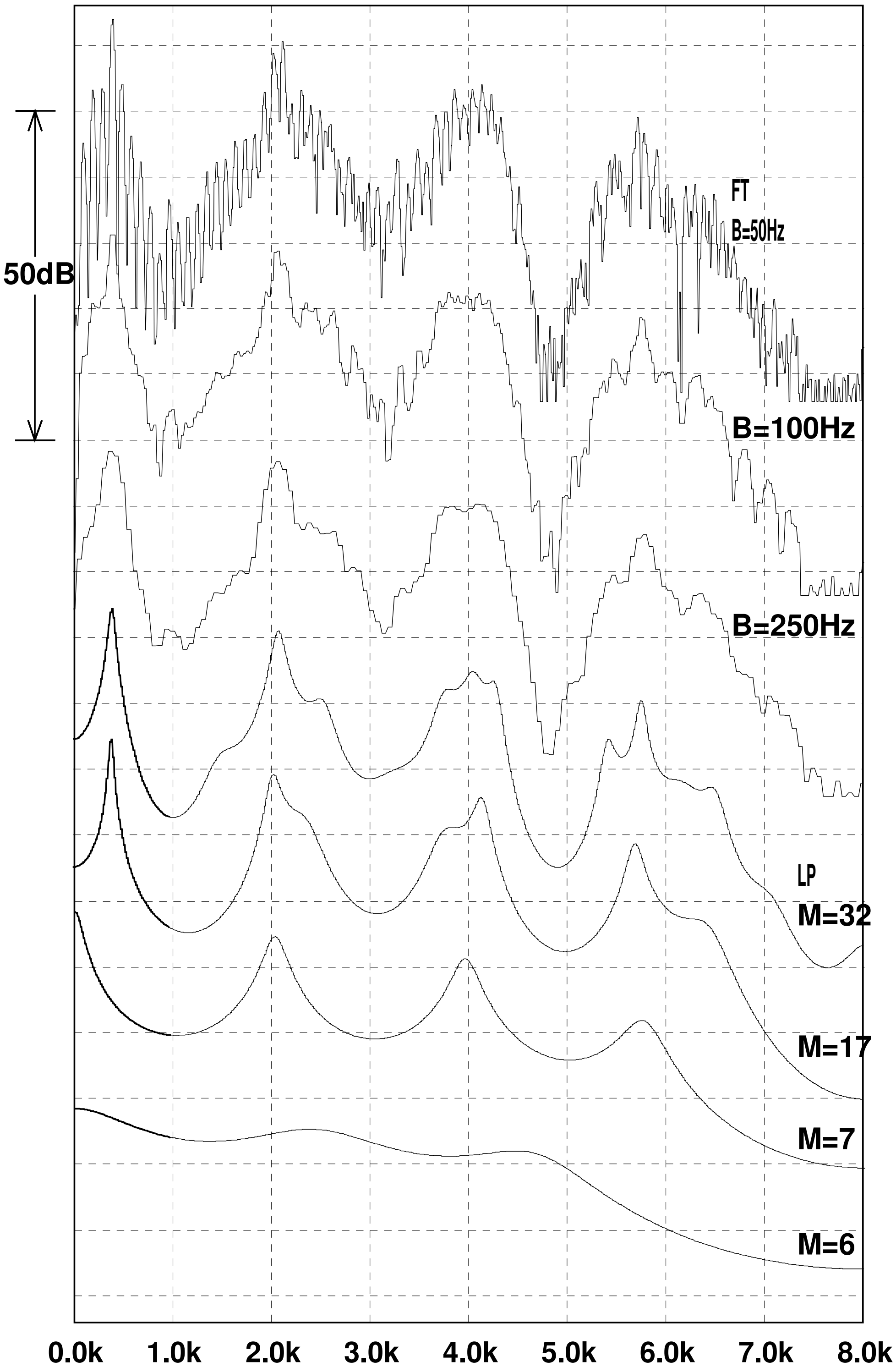

Fig 6. Spectrum sections of a vowel [e:] sampled at 16 kHz. The top three are Fourier spectra with different bandwidths to show or suppress the pitch harmonics. The bottom four are LP spectra of varying order. M=17 would be the more adequate with the sampling rate used if the peaks were supposed to indicate formants.

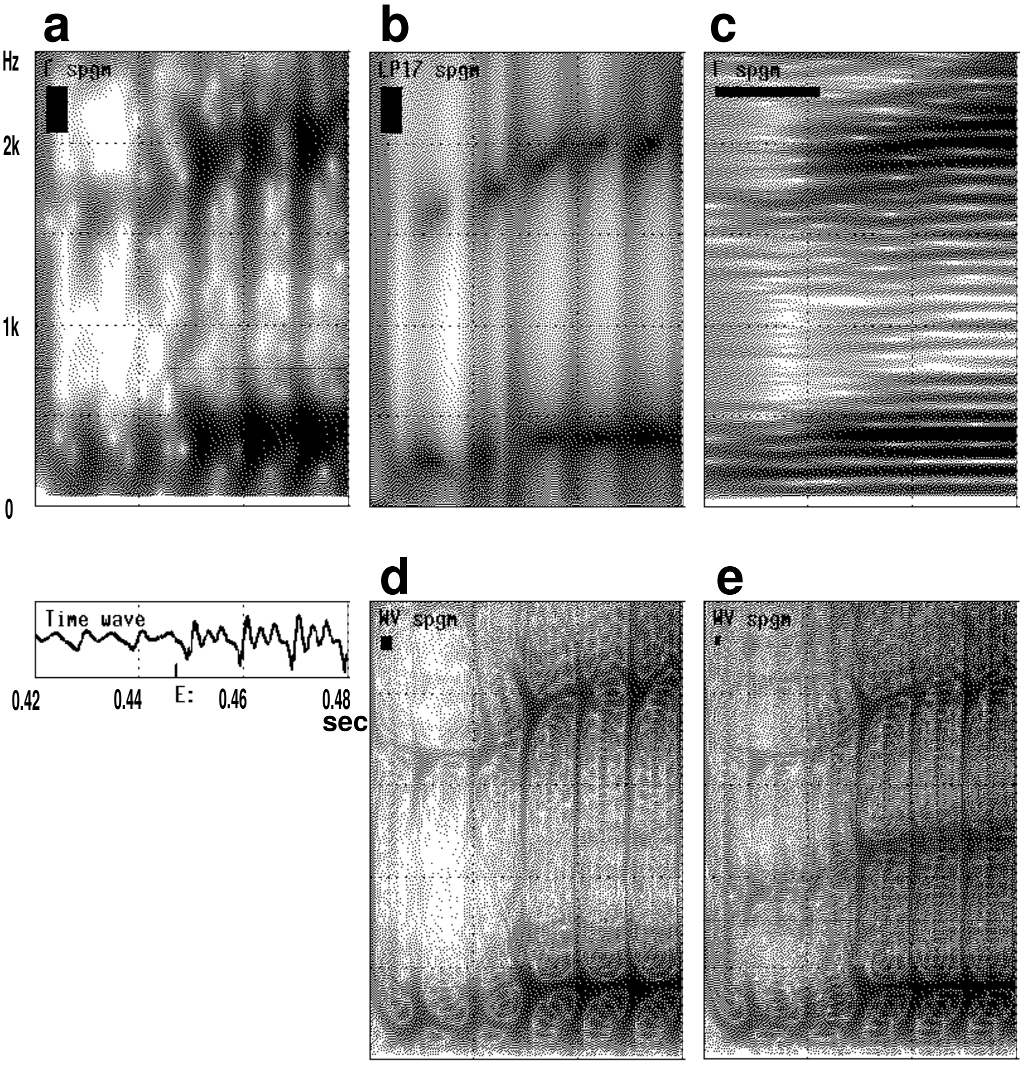

Fig 7. Enlarged details of spectrograms of a transition [le:] sampled at 16 kHz with 'logons' to indicate time and frequency resolutions.
*a: conventional Fourier 'wideband' with 4ms*250Hz,*
b: 17 coefficient autocorrelation LP with the same time window as a,
*c: conventional Fourier 'narrowband' with 20ms*50Hz,*
*d: smoothed Wigner-Ville distribution with nominally 2ms*70Hz,*
*e: same, but 1ms*35Hz giving interlacing artifical pitch pulses and formant track.*

5. Linear prediction analysis

Prediction theory has an origin in statistics for analysis of periodic sequences of data, like daily temperature or population birth rate. One of several interesting aspects is that it gives a way to identify seasonal variations, or correspondingly with speech, for instance to identify the systematic oscillations in the waveform due to the formants. This or similar methods have been applied in many fields, in speech first by Saito and Itakura (1966), and Atal and Schroeder (1967), and became a widespread standard tool in speech research after the comprehensive presentation by Markel and Gray (1976).

An N:th order predictor would use N historic samples, each contributing to the prediction by some weight factor b_i, and then we use precisely the formula (6) for a FIR filter (excluding b_0, otherwise there would not be much of a prediction). This predictor is called *linear* because it uses a linear combination of the samples, but nothing like powers or products of them. The concept of linearity is an important one that implies that superposition is legal: assuming two different inputs to a system which would generate two different outputs - then the sum of the inputs also generates the sum of those outputs uniquely.

There exist several methods to arrive at the predictor coefficients, covered extensively in the standard textbook literature. One way to formulate the problem is to arrange the predictor in the circuit of fig 8. The signal is compared to the prediction of it, and the resulting output difference is the *prediction error*.

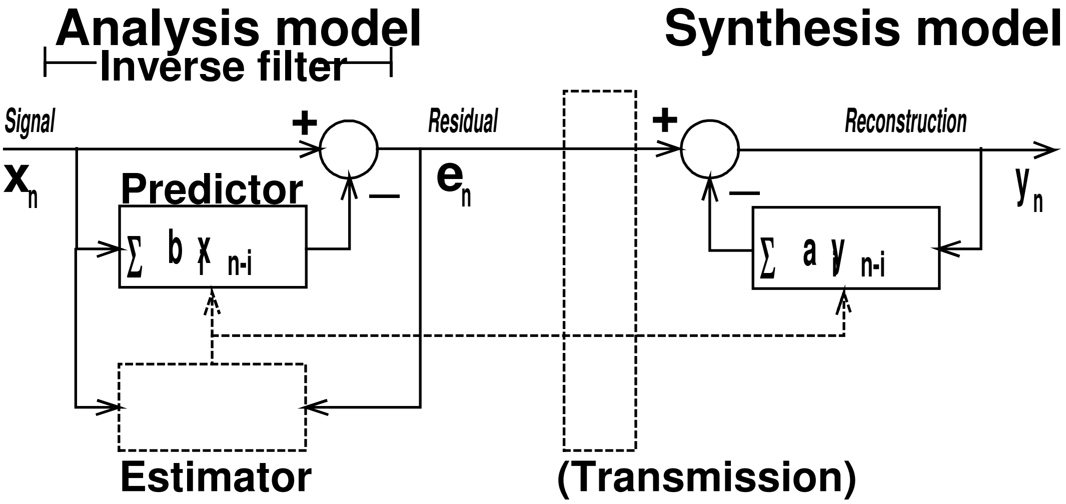

Fig 8. The basic processing blocks in LP modelling. The estimator computes optimal b_i from the signal x and e. In a basic vocoder application a reconstructed version of the signal can be fabricated using the same coefficients in an IIR filter.

The solution amounts to finding values of the coefficients such that this error becomes as small as possible, normally using a minimum squares criterion. Two standard methods, known for historical rather than mathematical reasons as the *covariance* and the *autocorrelation* methods, differ in details and in the range of samples used. We do not enter the mathematics of them, this is well published, down to the level of source codes for their central subroutines, for instance in Markel and Gray (1976) or in the publication by the IEEE DSP committee (1979).

The covariance method is efficient in extracting a maximum of relevant information from a minimal amount of data, but may give problems in accuracy and stability. The more reliable results come with good quality input data, not corrupted by noise, and that reflect a stationary system. Even with a static articulatory position the vocal system is not stationary in this sense because the glottis presents a loading impedance that varies considerably over the glottal period, not to speak of the disturbing discontinuity at glottal closure which normally constitutes a major part of the excitation. The covariance method is considered to perform best when the data come from the part of a voiced speech period where the glottis is closed. An inherent property of the method is that the solution does not necessarily represent a stable filter, the roots of the predictor polynomial may be located outside the unit circle.

With the autocorrelation method one normally uses a larger number of samples for the computations, for speech typically at least 256, covering more than one pitch period. To reduce truncation effects the signal segment is normally weighted with a window like a Hamming window. Part of the processing, as its name suggests, is to compute autocorrelations of the samples which also includes an averaging process which will reduce the influence of noise. Moreover this method inherently gives a stable result.

Having found the coefficients that minimize the prediction error the configuration of fig 8 will represent an *inverse filter*. This removes the spectrally prominent features (formants) in the signal and delivers a spectrally white residual at its output. The customary practice is to regard this as the inverse to the production filter in a source-filter production model. Whatever residual that remains of the speech wave after inverse filtering is categorized as the source. Ideally this white spectrum residue should be a pulse train in case of voiced speech, or random noise with unvoiced. Usually it does exhibit prominent peaks at the instances of excitation of the vocal tract, and this in itself makes it a much used input for pitch determination algorithms. It also serves as a fundamental raw material for studies of the vocal source. To account for the effect from radiation in speech production the speech signal is normally high frequency preemphasized with a first order filter prior to LP analysis.

The impulse response of the inverse filter is an initial unit value followed by the predictor coefficients b_i. By Fourier transformation of this we obtain its frequency response, and turning this upside down we get the estimated 'LP spectrum' of the signal. Being a filter characteristic this lacks information on the signal level which however can easily be found from the signal directly.

Fig 6 compares Fourier and LP spectra of different orders. It is important to recognize that although the LP spectrum appears clean and regular as compared to the Fourier spectrum, and more like an ideal textbook shape of a spectrum with formants, it does not necessarily represent the speech signal in a more truthful way than the Fourier spectrum. Rather the LP spectrum *is* the ideal textbook shape, it is precisely a formant model of the signal. The number of formants that can appear is a priori decided by the number of LP coefficients. If you make an LP analysis with M coefficients you will get no more than $M/2$ formants, independent of what number of formants the signal may really have. The merit of LP is that you can prescribe M from your knowledge or desire of how many formants to find, and then the LP analysis will deliver the best matching model within this restriction.

The frame fig 7b shows 17th order LP spectra in the format of a spectrogram. This representation is rarely used for visual display but is the more common as raw material for the estimation of formant patterns.

To get knowledge of the dimensionality from the signal alone has not been a primary concern for classical speech research but is intensely treated in the field of *system estimation*. Here speech serves as example of a most challenging application because of its rapid variation in number and values of its system parameters. Numerous alternative methods have been developed to find LP parameters in a sequentially *adaptive* manner, and simple forms have reached enough maturity to be incorporated in mobile telecommunications systems, like ADPCM (adaptive predictive PCM), see e.g. Jayant and Noll (1984).

Batch or frame processing vs. such sequential processing is an often encountered dichotomy. The Fourier and LP analyses are typical examples of the first where the spectrum or the LP coefficients are determined for a frame of signal samples. The data of the whole frame is treated in one comprehensive and relatively complicated process, and also the result is a vector, a composite set of data. The data for the next execution of the algorithm is taken some frame step time later in the input sequence, with speech perhaps in the range 2-50 ms depending on application. The IIR and FIR filters with their simple recursion formulas exemplify sequential processes. Although the input is several samples the process delivers an output for only one sample time slot, and the process is then subsequently repeated in its entirety, each time displaced by one sample interval. Sometimes the desired processing can be obtained either way. You can for instance do a spectrum analysis framewise using the FFT or you can do it sequentially with a bank of parallel bandpass filters. Which is the computationally more efficient way can be inferred from the requirements of output temporal and spectral resolution. The inherent efficiency of FFT can in such cases justify apparently wasteful solutions like pooling several adjacent spectrum samples into one wider band analysis channel. Numerous methods have been developed to bridge such gaps, for instance special variations of the FFT with 'pruned' inputs or outputs, or the constant Q transform (Brown, 1991). The chirp-z transform (Rabiner et al, 1969) is for evaluation away from the frequency axis. Frequency warping can at instances be integrated within a process like in the warped LP outlined by Strube (1980).

6. Pitch extraction techniques

The extraction of pitch is one of the perennial problems of speech research, reviewed for instance by Hess (1983). The literature describes methods by the hundreds, constantly reporting improvements, still an ideal method remains to be found. The reasons why a pitch determination algorithm (PDA) may fail are manifold, but prominent ones are that speech has a highly variable spectrum - the base for its information carrying capacity - as well as it shows considerable variation between speakers. These problems are likely to create difficulty when one attempts to measure or display accurate individual pitch periods within the laboratory. In practical applications like speech coding for communications we have the additional difficulties of band limiting, non-linear and phase distortions due to filtering and reverberation, and external noise. The more fundamental problem is however one of definition: what exactly is pitch? Most PDAs work on some specific feature connected with periodicity, and perform well if this feature is actually present. Suppose for instance a pitch meter that isolates the fundamental with a lowpass filter in order to measure its period. This will inevitably fail in case the fundamental has been removed by bandlimiting as in telephone speech, or in presence of low frequency noise, even such that may not be perceived by a human listener.

Most practical measurement algorithms, not only for pitch, can be logically partitioned into a preprocessor, an estimator proper, and a postprocessor. The task of the preprocessor is to condition the signal, for instance by dynamic control of spectral preemphasis and gain, to create an optimal signal for the estimator. The postprocessor typically identifies and corrects errors, it could for instance replace a deviant sample with a value somehow derived from its neighboring samples. In all three subsystems the degree of sophistication can be arbitrarily selected which in part explains the vast number of PDAs in the literature.

The historically earlier PDAs were based on direct processing of the time signal, typically detecting peaks (Dolansky, 1955) or zero crossings. Another popular time domain idea is that successive pitch periods should be of approximately equal shape.

This lies behind the autocorrelation and average magnitude difference methods (AMDF, Ross et al, 1974). If you compare two signal segments, located some time apart, then they should be maximally correlated (alternatively show a smallest difference) when this time equals the pitch period. The weakness of the time methods is that successive periods are not always very similar when articulation varies, or in the presence of reverberation or other disturbances. Some have as special merit that they reasonably locate some anchor point in each individual pitch period.

The other classical mainstream PDAs operate in the frequency domain and exploit the fact that a periodical signal has a number of evenly spaced harmonics. An initial process is then to Fourier transform the signal, and the PDA looks for the repetitive spectral peaks. The spectrum must then have sufficient resolution to show the individual harmonics which implies that the time interval transformed must have sufficient duration - you cannot detect a periodicity unless you look at more than one period. One way to detect the spectral periodicity is to compute the sum or product spectrum (Schroeder, 1968) or computationally efficient variations like subharmonic summation (Hermes, 1988). The principle is then that you make several versions of the spectrum with frequency axes rescaled by factors 1, 2, 3, etc., and then combine them by summing or multiplication. This way the harmonic peaks of a spectrum will give rise to a sharp coincidence peak while inharmonic components average out into a low floor.

Another way to identify the periodicity in the spectrum is to take the spectrum of the power spectrum. According to the Wiener-Kinchin theorem this amounts exactly to finding the autocorrelation function of the signal. A successful related variant is to first take the logarithm of the power spectrum - this makes all spectral lines have the same shape even if their levels vary with the formant envelope. The spectrum of the log power spectrum is called the *cepstrum*, with an abscissa of *quefrency,* the same dimension as time. This method and its vocabulary of permuted variants of the classical terms was conceived of in seismology by Bogert et al (1963), and was applied as a highly successful speech PDA by Noll (1967). A principal merit of this method is that it is largely insensitive to level variation and band limiting in the input - these only reflect as shift and truncation in the log spectrum. Thus the cepstral peak is largely of constant cepstral amplitude. A weakness appears when the signal has only a few harmonics (e.g. in the occlusion phase of a voiced plosive) or when the pitch is rapidly changing. In the latter case the higher harmonics in the log spectrum are smeared out and may overlap such that the periodic structure disappears.

Fig 9 shows the cepstrum of a voiced sound. Its contents at quefrencies shorter than the pitch peak (the short-pass liftered cepstrum) is a Fourier description of the grosser features of the log spectrum like level, slope and formant density. These first few cepstral values are often used as input data to speech recognizers. In that application it is also customary to reshape the log spectrum on a warped frequency scale before Fourier transformation of it into the cepstrum. This is mostly done on the basis of the Bark scale which is applied to account for the frequency selective properties of the hearing mechanism. The intention is to represent the perceptually relevant information with a minimal amount of data, a prime interest both in communications applications and research.

Modern PDAs often use the machinery of an LP inverse filter as preprocessor to remove the formant structure from the speech wave, for instance Markel (1972), Un and Yang (1977), Ananthapadmanabha and Yegnanarayana (1979), Cheng and O'Shaughnessy (1989). The accuracy and robustness of modern PDAs are often sufficient for use in

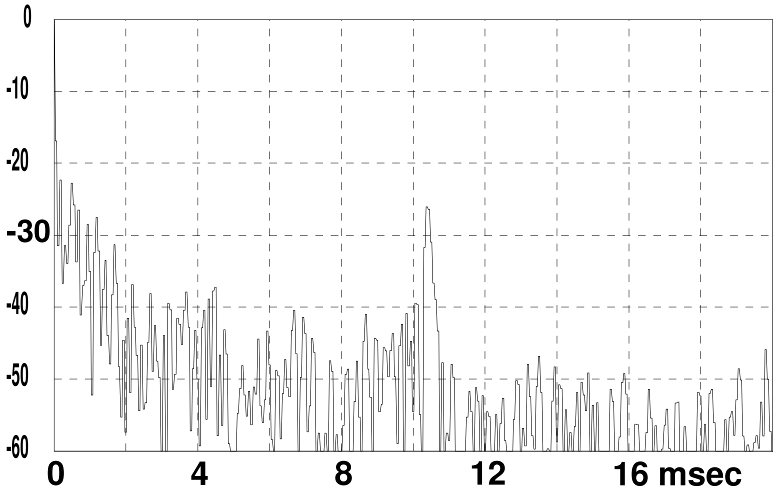

Fig 9. High resolution cepstrum of the same vowel [e:] as in figs 6 and 7.

communications systems, but pitch extraction remains a dynamic field now trying to incorporate more complete modelling of the human auditory system, see Hermes (1993).

7. Speech synthesis and coding

Knowing the inverse filter it is elementary as outlined above to set up the inverse of it, that is the synthesis filter. Another important variation is that as part of the LP coefficients solution we find what in different terminologies is called, *reflexion*, *parcor* (Itakura and Saito, 1972), or *k* coefficients. They have correspondent filter implementations in the form of *lattice filters* (Makhoul 1978, Friedlander 1982) which can be set up in a variety of equivalent forms by use of mathematical transformations and such tricks as introducing backward prediction. These are similar to reflexion line filters, or wave filters (Fettweis and Meerkötter 1975, Strube 1982) and relate back to classical vocal tract modelling as a set of abutting transmission line sections of varying areas. From *k* one can thus construct such an area function that would give rise to the observed filtering. Yet another form is the logarithm of the area ratios, LAR, which have been found to be effective and robust descriptors in speech coding for communications. The possibility to estimate vocal tract area functions reliably is however limited. It depends critically on proper handling of such side factors as preemphasis and that assumptions of the model (for instance that of one-dimensional wave propagation) are valid in the frequency band defined via the sampling rate.

An early application of LP was in vocoders where the transmitted data were the coefficients, the pitch and the level. These exhibited a 'reedy' and somewhat unnatural character in the resynthesized voice, much due to the use of a simple pulse generator for synthesis excitation. For high quality vocoding different schemes have been developed to transmit the residual in more detail, ranging from simple low-rate PCM over more complicated representations with sets of multiple unequal pulses each pitch period, up to schemes using vector quantization, so called code-book excited vocoders, CELP.

Vector quantization is an important topic in itself, also with several other applications for speech. See Makhoul et al (1985) or Gersho and Gray (1992). Assume that we have collected, from some learning material, a large number of *vectors*, that is ordered *sets* of numbers. They may for instance represent pitch period residuals, spectrum shapes, area functions, or sets of formant frequencies. A purpose of vector quantization is to extract the essentials of such a large amount of empirical data and classify it. The vectors are compared to each other with some pertinent distance measure and they are grouped into classes, generally much fewer classes in number than the vectors. The members of each class are then sufficiently close in value that they can be averaged into a single representative entry in a codebook. Later, using the codebook, we have as input one vector and its closest correspondent is searched for. The quantity subsequently recorded or transmitted is not the composite vector itself, but only its ordinal number in the codebook. To hint the order of magnitude involved, a codebook to represent with reasonable fidelity the various spectral patterns used by a population of speakers might contain some 1000 vectors.

8. References

Ananthapadmanabha, T. S., Yegnanarayana, B. (1979): Epoch extraction from linear prediction residual for identification of closed glottis interval. IEEE Trans ASSP-27, 309-319

Atal, B. S., Schroeder, M. R. (1967): Predictive coding of speech signals. Proc. 1967 Conf. Commun. and Process, 360-361.

Bogert, B. P., Healy, M. J. R., Tukey, J. W. (1963): The quefrency alanysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe cracking. In Rosenblatt, M, (ed.): *Proc. Symp. Time series analysis*, John Wiley & Sons, NY, 209-243.

Brown, J. C. (1991): Calculation of a constant Q spectral transform. JASA 89(1), 425-434.

Cheng, Y. M., O'Shaughnessy, D. (1989): Automatic and reliable estimation of glottal closure instant and period. IEEE Trans ASSP-37, 1805-1815.

Cohen, L. (1989): Time-frequency distributions - a review. Proc IEEE, 77(7), 941-981

Cooley, J. W., Tukey, J. W. (1965): An algorithm for the machine calculation of Fourier series. Math. Comput., 19, 297-301.

Dolansky, L. O. (1955): An instantaneous pitch period indicator. JASA 27, 67-72

Fettweis, A., Meerkötter, K. (1975): On adaptors for wave digital filters. IEEE Trans ASSP-23, 516-525.

Friedlander, B. (1982): Lattice filters for adaptive processing. Proc IEEE 70(8), 829-867.

Gersho, A., Gray R. M. (1992): *Vector quantization and signal compression*. Boston: Kluwer Academic.

Hermes, D. J. (1988): Measurement of pitch by subharmonic summation. JASA 83, 257-264.

Hermes, D. J. (1993): Pitch analysis. In Cooke et al (1993) (pp 3-25).

Hess, W. (1983): *Pitch determination of speech signals: Algorithms and devices*. Berlin: Springer.

IEEE Digital Signal Processing Committee (eds) (1979): *Programs for digital signal processing*. New York: IEEE Press.

Itakura, F., Saito, S. (1972): On the optimum quantization of feature parameters in the PARCOR speech synthesizer. In Flanagan, J. L., Rabiner, L. R. (eds) (1973): *Speech synthesis* (pp 289-292). Stroudsburg, PA: Dowden, Hutchinson and Ross.

Jayant, N. S., Noll P. (1984): *Digital coding of waveforms. Principles and applications to speech and video*. Englewood Cliffs: Prentice-Hall, Inc.

Markel, J. D., Gray, A. H. (1976): *Linear prediction of speech*. Communication and cybernetics 12. Berlin: Springer.

Makhoul, J. (1978): A class of all-zero lattice digital filters: properties and applications. IEEE Trans ASSP-26, 304-314.

Makhoul, J., Roucos, S., Gish, H. (1985): Vector quantization in speech coding. Proc IEEE 73-11, 1551-1588.

Markel, J. D. (1972): The SIFT algorithm for fundamental frequency estimation. IEEE Trans AU-20, 367-377

Noll, A. M. (1967): Cepstrum pitch determination. JASA 41, 293-309

Rabiner, L. R., Schafer, R. W., Rader, C. M. (1969): The chirp z-transform algorithm. IEEE Trans AU-17(2), 86-92.

Ross, M. J., Shaffer, H. L., Cohen, A., Freudberg, R., Manley, H. J. (1974): Average magnitude difference function pitch extraction. IEEE Trans ASSP-22, 353-362

Saito, S., Itakura, F. (1966): The theoretical consideration of statistically optimum methods for speech spectral density. Rep. no 3107, Electr. Comm. Lab, NTT, Tokyo (in japanese)

Schroeder, M. R. (1968): Period histogram and product spectrum: New methods for fundamental-frequency measurement. JASA 43, 829-834

Strube, H. W. (1980): Linear prediction on a warped frequency scale. JASA 68(4), 1071-1076.

Strube, H. W. (1982): ?

Un, C. K., Yang, S. C. (1977): A pitch extraction algorithm based on LPC inverse filtering and AMDF. IEEE Trans ASSP-25, 565-572.

Velez, E., Absher, R. (1989): Transient analysis of speech signals using the Wigner time-frequency representation. Proc ICASSP '89, 2242-2245.

9. Other reading

Atal, B. S., Hanauer, S. L. (1971): Speech analysis and synthesis by linear prediction. JASA 50, 637-655.

Atal, B. S., Remde J. R. (1982): A new model of LPC excitation for producing natural-sounding speech at llow bit rates. Proc ICASSP 82, vol 1, 614-617.

Cooke, M., Beet S., and Crawford S. (eds) (1993): *Visual representations of speech signals*. Chichester: Wiley.

Fant, G. (1960): *Acoustic theory of speech production*. Haag: Mouton & Co.

Flanagan, J. L .(1972): *Speech analysis synthesis and perception*. Berlin: Springer.

Flanagan, J. L., Rabiner, L. R. (eds) (1973): *Speech synthesis*. Stroudsburg, PA: Dowden, Hutchinson and Ross.

Furui, S., Sondhi, M. M. (eds) (1992): *Advances in speech signal processing*. New York: Marcel Dekker, Inc.

Hess, W. (1992): Pitch and voicing determination. In Furui, S, Sondhi, MM, (eds) (1992): *Advances in speech signal processing* (pp. 3-48). New York: Marcel Dekker, Inc.

Martin, P. (1982): Comparison of pitch detection by cepstrum and spectral comb analysis. Proc ICASSP 180-183

Rabiner, L. R., Juang, B. H. (1986): An introduction to hidden Markov models. IEEE ASSP Magazine, 4-16

Robinson, E. A. (1982): A historical perspective of spectrum estimation. Proc IEEE, 70(9), 885-907.

Schafer, R. W., Markel, J. D. (eds) (1979): *Speech analysis*. New York: IEEE Press.

Schroeder, M. R., (ed.) (1985): *Speech and speaker recognition*. Bibliotheca Phonetica 12. Basel: Karger.

10. Handbooks

Akansu, A. N., Haddad, R. A. (1992): *Multiresolution signal decomposition: transforms, subbands and wavelets*. Boston: Academic Press

Baher, H. (1990): *Analog & digital signal processing*. Chichester: Wiley.

Barkat, M. (1991): *Signal detection and estimation*. Boston: Artec House

Beauchamp, K. G. (1987): *Transforms for engineers : a guide to signal processing*. Oxford: Clarendon.

Bellanger, M. (1989): *Digital processing of signals : theory and practice*. 2. ed. Chichester: Wiley.

Blahut, R. E. (1985): *Fast algorithms for digital signal processing*. Reading, Mass.: Addison-Wesley

Blahut, R. E. (1992): *Algebraic methods for signal processing and communications coding*. New York: Springer.

Boashash, B. (ed.) (1992): *New methods in time-frequency analysis*. Sydney: Longman Cheshire.

Bowen, B. A., Brown, W. R. (1982): *VLSI systems design for digital signal processing*, Vol 1: Signal processing and signal processors. Englewood Cliffs: Prentice-Hall, Inc.

Brook, D., Wynne, R. J. (1988): *Signal processing : principles and applications.* London: Edward Arnold.

Combes, J. M., Grossmann, A., Tchamitchian, Ph., (eds) (1990): *Wavelets: time-frequency methods and phase space*. 2. ed. Berlin: Springer.

Crochiere, R. E., Rabiner, L. R. (1983): *Multirate digital signal processing*. Englewood Cliffs: Prentice-Hall, Inc.

Dudgeon, D. E., Mersereau, R. M. (1984): *Multidimensional digital signal processing*. Englewood Cliffs: Prentice-Hall, Inc.

Gold, B. (1969): *Digital processing of signals*. New York: Lincoln Lab. Publ.

Haddad, R. A., Parsons, T. W. (1991): *Digital signal processing : theory, applications, and hardware*. New York: Computer Science.

Haykin, S. (1984): *Introduction to adaptive filters*. New York: Macmillan.

Jackson, L. B. (1986): *Digital filters and signal processing*. 2nd ed 1989. Boston-Dordrecht-Lancaster: Kluwer Academic Publ.

Kraniauskas, P. (1992): *Transforms in signals and systems*. Wokingham: Addison-Wesley.

Kailath, T. (ed.) (1985): *Modern signal processing*. Washington: Hemisphere Publ.

Kesler, S. B. (1986): *Modern spectrum analysis, II*. New York: IEEE Press.

Malwar, H. S. (1992): *Signal processing with lapped transforms*. London: Artech House.

Mitra, S. K., Kaiser, J. F. (1993): *Handbook for Digital Signal Processing*. Wiley.

Oppenheim, A. V., Schafer, R. W. (1975): *Digital signal processing*. Englewood Cliffs: Prentice-Hall, Inc.

Oppenheim, A. V., Willsky, A. S., Young, I. T. (1983): *Signals and systems*. Englewood Cliffs: Prentice-Hall, Inc.

Oppenheim, A. V., Schafer, R. W. (1989): *Discrete-time signal processing*. Englewood Cliffs: Prentice-Hall, Inc.

Parsons, T. (1987): *Voice and speech processing*. New York: McGraw-Hill, Inc.

Priemer, R. (1991): *Introductory signal processing*. Singapore: World Scientific.

Proakis, J. G., Manolakis, D. G. (1992): *Digital signal processing*. New York: Macmillan Publ Co.

Rabiner, L. R., Rader, C. M. (eds) (1972): *Digital signal processing*. New York: IEEE Press.

Rabiner, L. R., Gold, B. (1975): *Theory and application of digital signal processing*. Englewood Cliffs: Prentice-Hall, Inc.

Rabiner, L. R., Schafer, R. W. (1978): *Digital processing of speech signals*. Englewood Cliffs: Prentice-Hall, Inc.

Roberts, R. A., Mullis, C. T. (1987): *Digital signal processing*. Reading, Mass.: Addison-Wesley.

Stanley, W. D., Dougherty, G. R., and Dougherty, R. (1984): *Digital signal processing*. Reston Publ Co, Inc (Prentice-Hall)

Young, R. K. (1993): *Wavelet theory and its applications*. Boston: Kluwer.

Young, T. (1985): *Linear systems and digital signal processing*. Englewood Cliffs: Prentice-Hall, Inc.