Matlab Auditory Demonstrations

Speech and Hearing Research
Department of Computer Science
University of Sheffield

Version 2.0

Document revision 0.1
Stuart Cunningham
10-May-1998

`http://www.dcs.shef.ac.uk/~martin/MAD/docs/mad.htm`

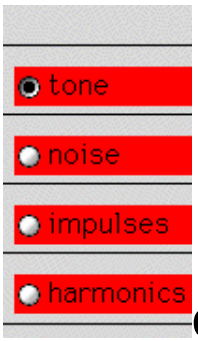
Documented Utilities

- createSig

Interactive Demonstrations

- Autocorrelation
- Basilar membrane
- Binaural masking-level difference
- Cepstral liftering
- Detuned harmonics
- Distorted speech
- Isolated word end-point detection
- Interleaved melody identification
- Linear prediction
- Missing data automatic speech recognition
- Pipeline processing
- Pole zero
- Speech labelling tool
- Strands: auditory time-frequency representation
- Tone streaming
- Sine-wave speech
- Temporal induction
- Simple time-domain processing
- Vowel explorer
- Waveforms and spectra

Documented Utilities

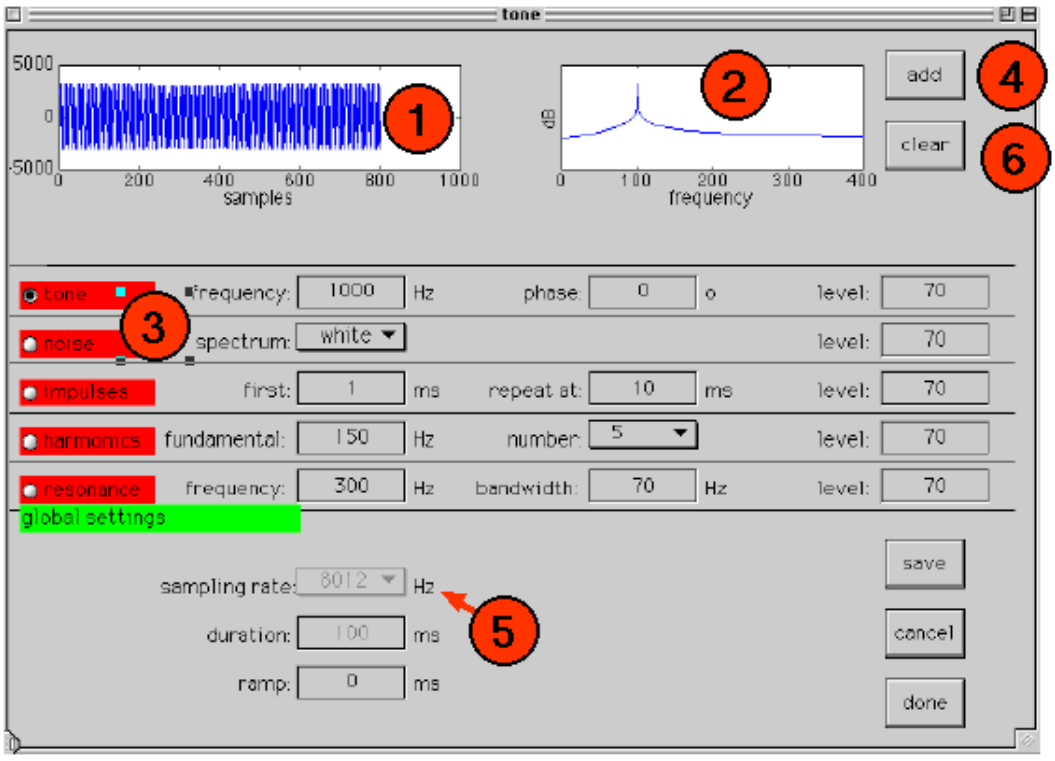


createsig: create signal

Introduction

This tool allows interactive creation of standard signals such as tones, impulses, etc and additive combinations thereof. This tool can be called from within a program, or from the command line.

The tool



From the command line or from within a program, type

```
[sig,fs,name]=createsig;
```

to launch the demo. An interface similar to that above will appear (it is possible that later versions will contain extra signal options at 3).

Initially, choose a sampling rate and duration (5). Once set, these cannot be modified until the signal is cleared (6). Then, select a signal type (3) and choose values for the relevant parameters. To add this signal into the existing signal, click add (4). Alternatively, clear the signal (6) and add. The waveform and its DFT spectrum will appear at 1 and 2. You can listen to the signal by clicking on either waveform or spectrum. A ramp of given on/off durations can be added if desired.

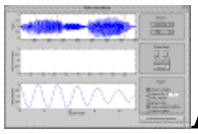
Known limitations

1. The frequency axis probably isn't in Hz just yet.
2. There is no undo option.

Credits etc

Produced by: Martin Cooke, April 1999.

Interactive Demonstrations



Autocorrelation

Introduction

Pitch is defined as *that attribute of auditory sensation in terms of which sounds may be ordered on a musical scale* (American Standards Association). There are a number of theories of pitch perception and these have given rise to computational models which implement them. These models have 3 stages:

1. *peripheral processing*
2. *feature analysis*
3. *pitch determination*

For pitch perception models that use temporal information some mechanism for identifying periodicities in the signal for use in the feature analysis stage is required. This demonstration does precisely this.

The usual method for deciding if a signal is periodic and then estimating its period is the autocorrelation function:

$$acg(\tau)=\sum_{t=1}^{N} x(t)x(t-\tau)$$

Essentially, all that is happening is the signal *x(t)* is being convolved with a time-lagged version of itself. To obtain a useful set of results, the autocorrelation function is computed over a range of lag values.
It is an important property of the autocorrelation function that it is itself periodic. For periodic signals the function attains a maximum at sample lags of 0, +-P, +-2P, etc. where P is the period of the signal.

One major limitation of the autocorrelation function is that it can retain too much information present in the signal. In speech, numerous peaks present in the autocorrelation function are due to damped oscillations of the vocal tract response. If these peaks happen to be bigger than the peaks due to periodicity, the simple procedure of picking the largest peak to be the period will fail.

Therefore, the signal needs to be pre-processed in some way to make the periodicity more prominant while suppressing other features which may cause distracting peaks. Such pre-processing techniques are sometimes called *spectrum flatteners*. Many techniques have been proposed but **centre clipping** [2] appears the best for this situation.

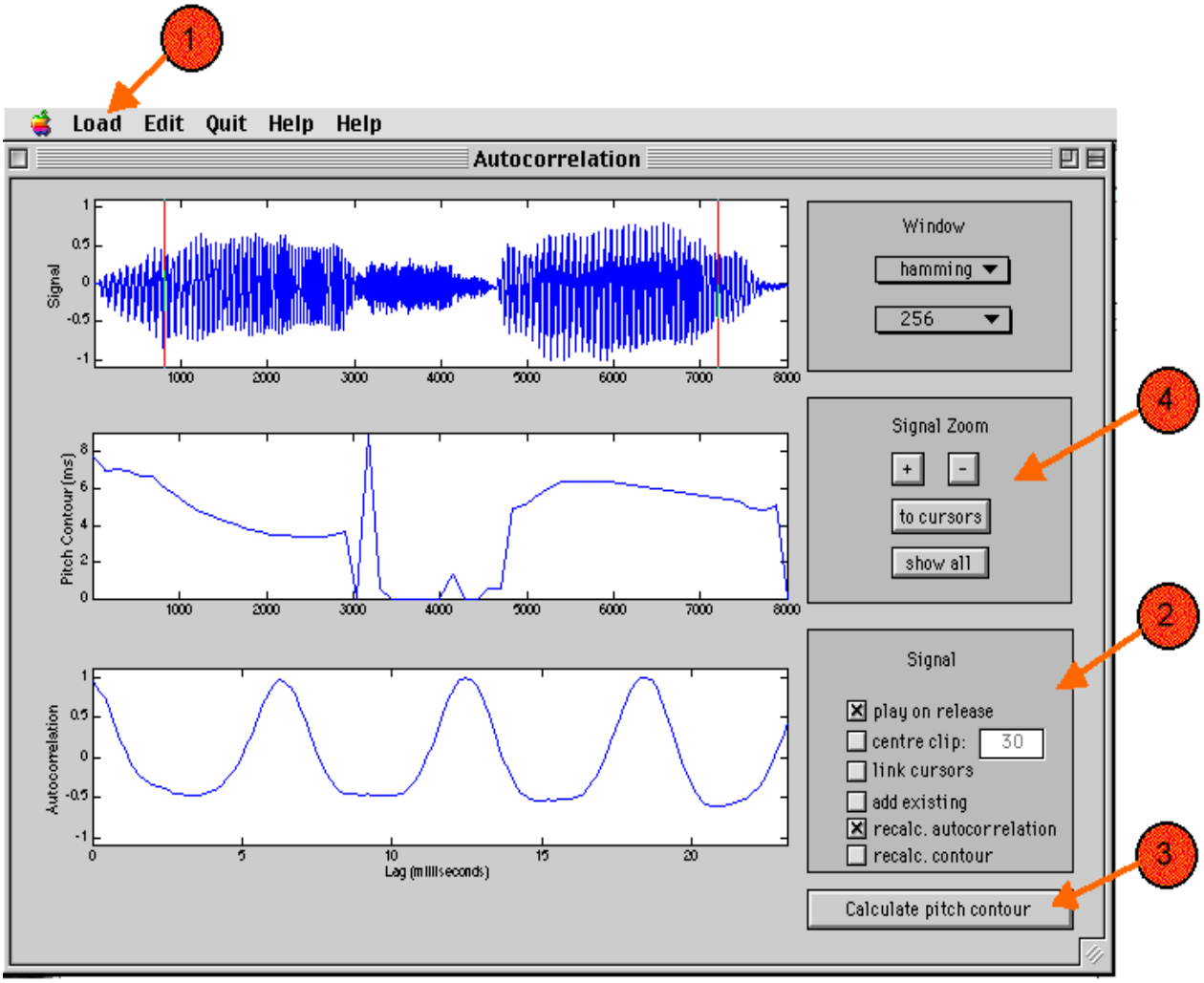
Centre clipping works by clipping a certain percentage of the waveform. Let Amax be the

maximum amplitude of the signal and CL be the clipping level. CL is a fixed percentage of Amax (say 30%). Therefore, the output from the center clipper is as follows:

$$y(n) = x(n)-CL \quad [x(n)>CL]$$
$$y(n) = 0 \quad [x(n)<=CL]$$

ie, for samples below the clipping level, the output is zero, and for samples above the clipping level, the output is equal to the input minus the clipping level. See [1, page 151] and [2] for more information.

The demonstration



Type 'auto' to launch the demo. When the window appears, use the load menu (1) to load a either a tone, a noise or a sound file. The signal can be played by clicking anywhere within the signal axes. Once the signal has been loaded and displayed, a set of cursors appear. These can be moved about in order to select various parts of the signal. They can be linked

together if necessary (2). By default, the segment of the signal contained within the cursors is played after movement of the cursors. This can be turned off if desired (2). The use of centre clipping (see above) and the clipping level can also set in (2).

When the left hand cursor is moved, the autocorrelation plot will refresh showing the current autocorrelation function output. An overall estimate of the signal's pitch - its *pitch contour* can be displayed by clicking the button at the bottom right of the demo (3). This can be automatically updated when the window type or size is changed(2)

The zoom panel (4) allows the signal (and the pitch contour if present) to be zoomed in on.

Things to investigate

1. When you place the cursors over a vowel, what shape is the autocorrelation function output?
   Now place the cursors over a fricative, for example, an /s/ sound. What shape is the output now?
   How can your observations be explained?
2. Display the pitch contour. Whilst slowly tracking along the signal with the left cursor, study how increases and decreases in the pitch shown on the pitch contour relate to changes in the autocorrelation function output.
   Are there any discontinuities (spurious peaks) in the pitch contour? What is causing these?
3. What effect does turning centre clipping on have?
4. Does altering the window type alter the plots? If so, why?

References

[1] Rabiner, L.R. and Schafer, R.W., "Digital Processing of Speech Signals". Prentice-Hall, 1978.

[2] Sondhi, M.M., "New Methods of Pitch Extraction". IEEE Trans. Audio and Electroacoustics, Vol. AU-16, No.2, pp.262-266, June 1968.

Further reading

For a frequency-domain method of pitch estimation, see also the demonstration for pipeline processing. (**pipeline**).

Credits

Produced by: Stuart N Wrigley

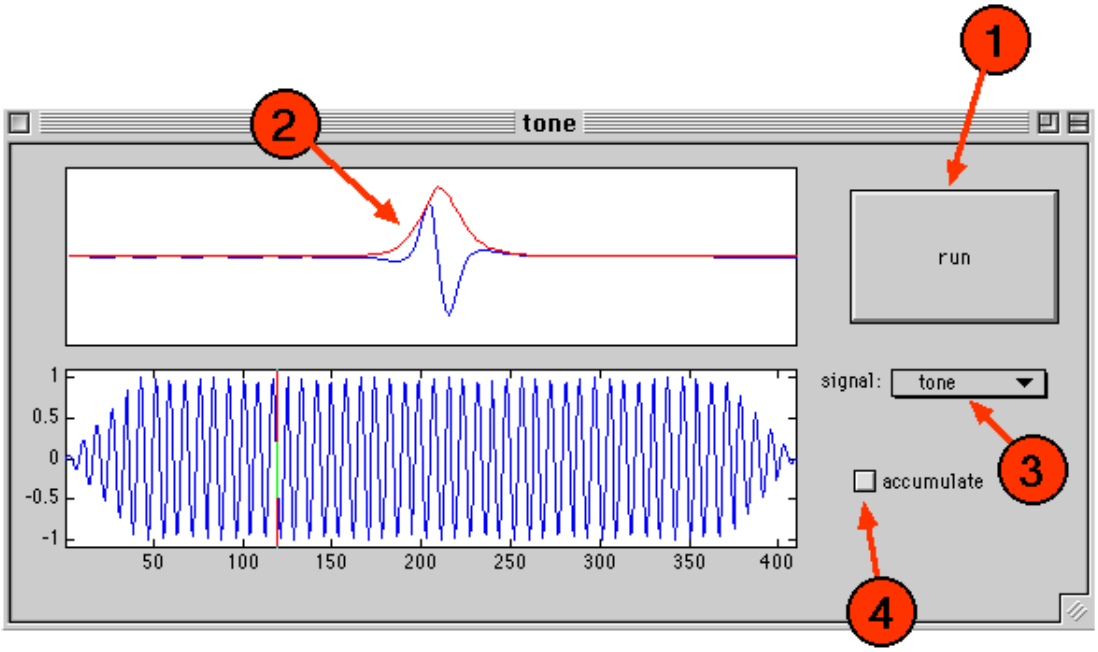
Release date: January 20 1999

bm: simple basilar membrane

Introduction

This demonstration shows the response of a simple basilar membrane (BM) simulation to tones, impulses and noise (and combinations thereof). The underlying BM model is a linear, passive system based on a bank of gammatone filters.

The tool



Type ’bm’ to launch the demo. Now select a signal from the menu at (3). You will be presented with a dialog box asking for various parameters such as duration, level etc. The resulting signal will appear in the lower display. Press the big ’run’ button (1) to start the simulation. The upper display will start to vibrate. At the same time, a cursor will move along the lower signal window to indicate the current input being processed. The left end of the BM display corresponds to low frequencies -- this will become obvious once you see the thing responding to a noise stimulus (or indeed any stimulus via the illusion of propagation from right to left). The red line stores the maximum BM displacement.

Additional signals can be added in to existing ones if the accumulate box (4) is checked. This allows the response of several tones of different frequencies/levels, or tones in noise, to

be investigated.

The 'params' menu allows the parameters of the BM filterbank to be modified. Specifically, the upper and lower frequencies, and the number of filters, can be changed.

Things to investigate

1. Load an impulse. Run the simulation. Note the propagation from right (high frequency) to left (low frequency). This propagation is an illusion since the model components are quasi-independent filters in each frequency band. The higher frequency filters respond faster.
2. Load a noise signal. Observe the response. Now click the accumulate box and load a 1 kHz tone. Note the presence of the tone in the BM response.
3. Load a tone on its own and observe the location of the peak in the BM displacement. Now change the tone frequency and note the change in location of the peak displacement. Add in a further tone of a different frequency. What cues apart from peak location might be used to signal tone frequency?

Further reading

- The BM model upon which this simulation is based is the impulse invariant transform approximation to the gammatone derived in Cooke (1993) *Modelling Auditory Processing & Organisation*, Cambridge.

Credits etc

Produced by: Martin Cooke

Release date: October 5th 1998

Binaural Masking Level Difference

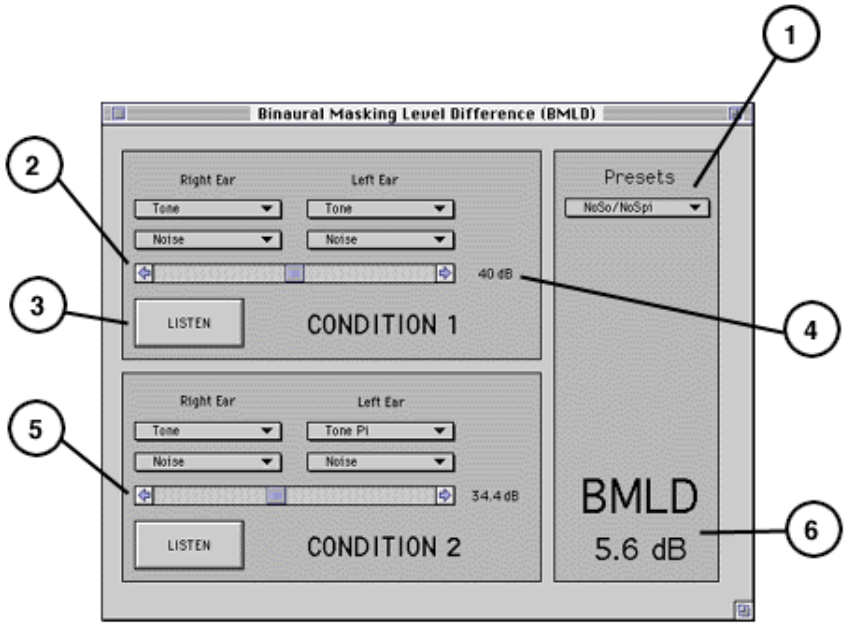
Introduction

The masked threshold of a signal can sometimes be lower when listening with two ears rather than one; this is demonstrated by the phenomenon of the binaural masking level difference (BMLD). The BMLD can be summarised as follows; the detection of a signal in noise is improved when either the phase or level differences of the signal at the two ears are not the same as the masker. An implication of this is that the signal and masker appear to originate from different locations in space; hence, the BMLD appears to be related to the well-known 'cocktail party effect'.

In general, we can describe a particular stimulus using the symbols S (for signal) and N (for noise), each followed by a suffix to indicating the relative phase in the two ears; 'o' for same phase (so-called homophasic) and 'pi' for a 180 degree (pi radians) phase difference (so-called antiphasic). For example, NoSo means that the noise and signal both have the same phase in each ear, and NoSpi means that the noise has the same phase, but the signal is 180 degrees out of phase. Nu means that the noise is uncorrelated in the two ears, and the suffix m indicates monaural presentation (i.e. presentation to one ear only).

BMLDs are largest for low-frequency tones. This demonstration uses a tone frequency of 250 Hz.

The demonstration



Choose one of the presets from the menu (1). Now, move the slider in the CONDITION 1 box (2) until the tone is just masked by the noise, using the LISTEN button (3) to hear the stimulus. The masking level is shown in dB (4). Now do the same in the CONDITION 2 box (5); move the slider until the tone is just masked by the noise. The BMLD (6) is the difference between the masking level in the CONDITION 2 box and the masking level in the CONDITION 1 box.

Things to investigate

Try each of the preset conditions and note the BMLD in each case. Compare your figure with the following, which are typical BMLDs for each interaural condition:

NoSo/NoSpi	15 dB
NoSo/NpiSo	13 dB
NmSm/NoSm	9 dB
NmSm/NpiSm	6 dB
NoSo/NuSo	4 dB
NoSo/NuSpi	3 dB

References

B.C.J. Moore (1997) An introduction to the psychology of hearing (fourth edition). Academic Press.

Further reading

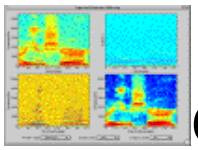
None.

Credits

Produced by: Guy J. Brown

Release date: 6th November 1998.

ÿ



Cepstral Liftering

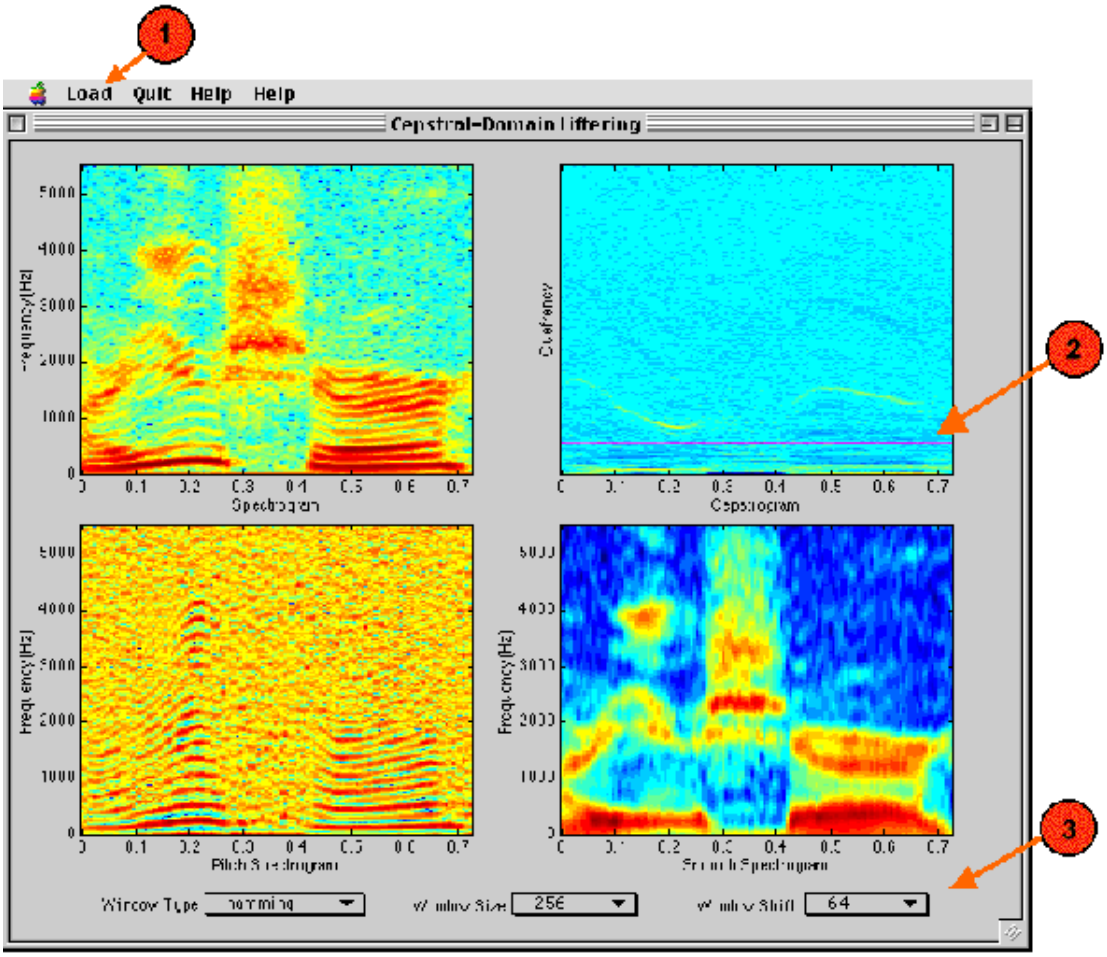
Introduction

A cepstrogram is similar in many ways to a spectrogram. The only difference is that each vertical 'strip' is a *cepstral* slice and not a *spectral* slice as in a spectrogram. The method of obtaining a cepstral slice can be investigated using the Pipeline Processing MAD demonstration (pipeline).

A cepstral slice exhibits a very interesting property: there is a prominant peak in the slice at the point corresponding to the signal's pitch. In the cepstrogram, the series of these peaks appears as a smooth contour. It is known that if the cepstral slice is liftered (the cepstral-domain equivalent of filtering) in a specific place to separate it into two portions, two of the signals characteristics can be examined independently: its spectral envelope and its pitch and harmonic information.

This demonstration allows the user to compare the original spectrogram with the two resulting spectrograms by interactively liftering the cepstrogram in different places.

The demonstration



Type 'ceplift' to launch the demo. When the window appears, use the load menu (1) to load a sound file. After a short wait the spectrogram, cepstrogram together with the smooth and pitch spectrograms will appear. There is a cursor on the cepstrogram which defines the liftering position (2). Move this cursor to see how the differing positions affect the smooth and pitch spectrograms. The parameters of the windowing function can also be altered (3).

Things to investigate

1. At what point does the smooth spectrogram begin to contain pitch and harmonic information?
2. What happens when the liftering point is moved close to the bottom of the cepstrogram? Why is this?
   Conversely, what happens when the liftering point is move close to the *top* of the cepstrogram? Why is this?
3. Identify the pitch contour. Above this, it may be possible to see a fainter contour with a similar shape. What is this contour?
4. Does altering the window type alter the plots? If so, why?

References

Brown, G.J. and Cooke, M.P., "COM325 Speech and Hearing" Course Notes. Department of Computer Science, University of Sheffield.

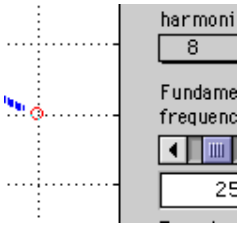
Further reading

See also the demonstration for pipeline processing (**pipeline**).

Credits

Produced by: Stuart N Wrigley

Release date: January 20 1999

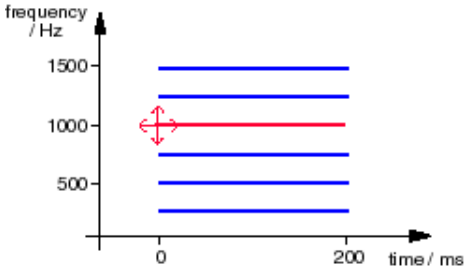


Detuned harmonics

Introduction

Any periodic waveform can be constructed by adding together sinusoids at integer multiples of the fundamental frequency corresponding to the period, known as *harmonics*. Frequency selectivity in the auditory system separates out these Fourier components at least for the lower portion of the spectrum, yet we perceive a single complex tone whose pitch corresponds to the fundamental frequency and whose timbre depends principally on the relative strength of the harmonics. The theory of Auditory Scene Analysis (Bregman 1990) interprets this integrated percept as the *fusion* of simultaneous tones on the basis of their common fundamental frequency (i.e. their *harmonicity*) as well as the common onset (and offset) times introduced when the sinusoids are turned on simultaneously.

This naturally leads to the question of what happens if these cues of harmonicity and common onset are degraded or absent. In broad terms, shifting the frequency of one harmonic -- i.e. *detuning* it -- will cause it to be segregated from the rest of the complex, and heard as a separate tone; shifting it in time so that it starts earlier or later than the remaining tones will similarly inhibit fusion. These manipulations are illustrated in the figure below, where the red harmonic can be made to stand out from the remaining blue harmonics by moving it in either or both of the dimensions indicated by the small arrows.



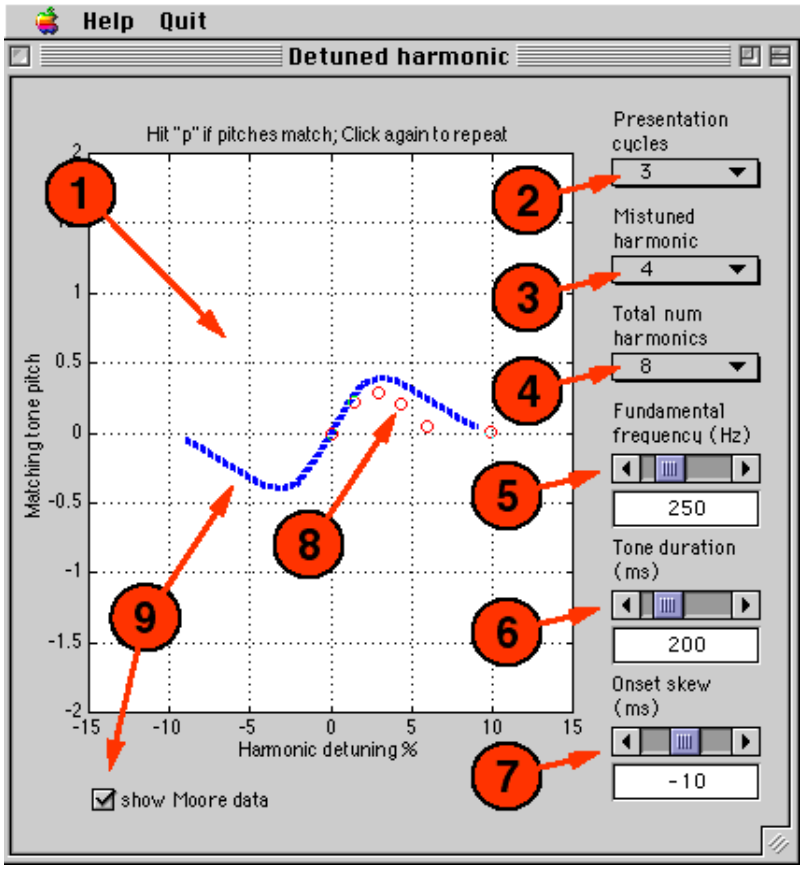
Rather than simply asking subjects if they hear one sound or two, a more sensitive measure of integration can be obtained by asking subjects to match the pitch of the residual complex to a strictly-harmonic tone. Very small mistunings of one harmonic leave the complex whole but produce a measurable shift in the overall pitch; we might also expect that once the mistuned harmonic is perceived as a separate element, it will make no contribution to the pitch of the residual complex.

This situation was investigated by Moore *et al.* (1985). They found that although subjects

were aware of a detuned 4th harmonic at mistunings of 1% or less, it continued to have a measurable effect on the pitch of the residual complex out to about 8% mistuning, with a maximum effect around 3%. Many other experiments have been performed with these or similar studies. In particular, Ciocca & Darwin (1993) found that starting the mistuned harmonic significantly earlier than the remaining harmonics could remove it completely from the pitch effect, but only if the harmonic during the complex was 'organized' as a continuation of the earlier-starting tone (which they were able to defeat with alternative organizations).

Darwin & Carlyon (1995) interpret this gradual removal of the mistuned harmonic from the grouped complex, as measured from the overall pitch, as evidence that grouping is not an 'all-or-nothing' effect, and that different aspects of auditory perception, such as the number of objects and the pitch of each one, may use separate versions of the organization of energy into sources.

The demonstration



Launch the demonstration with the command 'detuning'. Clicking anywhere in the graph (1) results in the delivery of a stimulus consisting of repetitions of a detuned-harmonic complex followed by a purely-harmonic complex; the horizontal co-ordinate at the clicked point determines the degree of mistuning of the single harmonic in the first tone, and the vertical co-ordinate controls the shift applied to the entire matching tone. The task is to match the

pitch of the residual harmonic complex (the lower pitch when two are heard) to the second tone by moving up and down at a particular degree of mistuning.

Other stimulus parameters are controlled by the popup menus and sliders on the right-hand side of the window. The number of repetitions is governed by the top popup menu (2). The second popup (3) specifies which harmonic in the complex is mistuned (where larger numbers indicate increasingly high-frequency components). The third menu (4) selects the total number of harmonics in both detuned and matching complexes, which are always a contiguous set starting from the fundamental. Note that if the harmonic indicated for detuning is higher than the top of the complex, no detuning will be heard.

Control (5) allows the fundamental pitch of the complex to be varied between 0 and 1000 Hz, either by moving the slider or by typing a value directly into the box. Similar control (6) is provided for the duration of each tone, varying between 0 and 1000 ms. Finally, control (7) can cause the detuned harmonic to be started up to 200 ms before or after the remainder of the complex. Negative values correspond to the detuned harmonic starting before the rest of the complex.

After some practice at matching the pitches, you may wish to record your responses. After each stimulus presentation, pressing 'p' will result in a red circle (8) appearing at the co-ordinates corresponding to the parameters. After a number of such identifications, compare your responses with a schematic approximation to Moore et al's results (9).

References

1. Moore, B.C.J., Glasberg, B.R. & Peters, R.W. (1985), Relative dominance of individual partials in determining the pitch of complex tones, Journal of the Acoustical Society of America, 77, 1853-1860.
2. Ciocca, V. & Darwin, C.J. (1993), Effects of onset asynchrony on pitch perception: Adaptation or grouping?, Journal of the Acoustical Society of America, 93(5), 2870-2878.
3. Darwin, C.J. & Carlyon, R.P. (1995), Auditory Grouping, in: The Handbook of Perception and Cognition, Vol 6, Hearing (ed: B.C.J. Moore), Academic Press, 387-424.

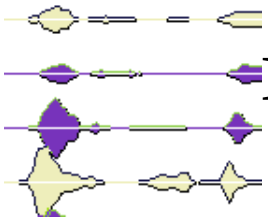
Further reading

The Darwin & Carlyon chapter mentioned above provides a good description of the basic phenomena as well as arguing their particular interpretation in terms of multiple grouping mechanisms.

Credits etc

Produced by: Dan Ellis

Based on: "streamer" demo by Martin Cooke



Distorted speech

Introduction

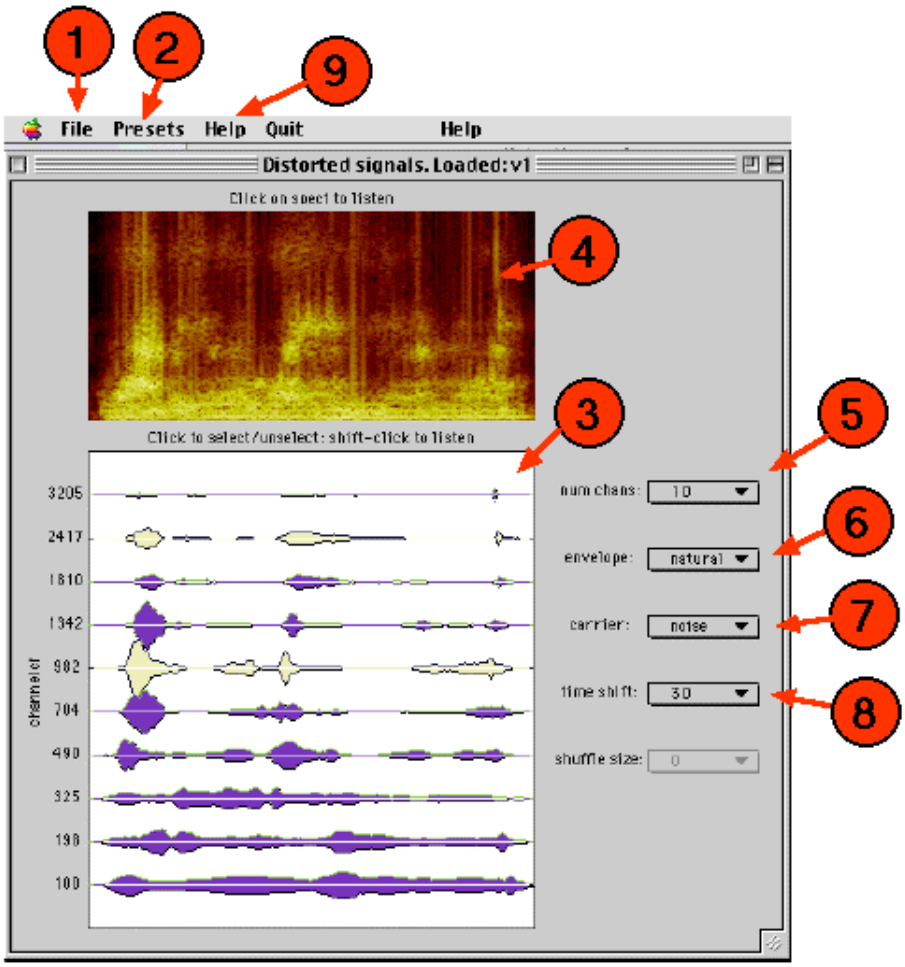
’Speech’ often arrives at the ears of the listener as a rather different entity than on production. The effects of reverberation, communication channel restrictions and failures, and the presence of other sources all contribute to the degraded signal. Listeners possess strategies for handling many types of distortion, and uncovering these strategies is of importance in designing robust systems for computational hearing.

The range of possible distortions is quite wide, and many forms of signal modification have been the subject of psychoacoustic inestigation since Fletcher’s work in the early part of the 20th century (see Fletcher, 1953; Allen, 1994). Speech signals have been subjected to such distortions as:

- specral filtering
- spectral reduction to a small number of sinusoids (see the sine-wave speech demo) or narrow bands
- multi-channel temporal distortion
- envelope and carrier modifications
- periodic or random interruptions and fading (see temporal induction demo)

The current demonstration allows the user to produce some of these forms of distortion.

The demonstration



To start the demonstration, type 'distortion' at the MATLAB prompt. A window like the one above will appear. Choose the load option in the file menu (1) to select a sound file (some speech and music examples are provided with the distribution, but any .au or .snd format files can be loaded). A spectrogram of the sound appears in the top panel (4). As distortions are applied, this display is updated to show the distorted sound spectrogram. To hear the signal, click anywhere in the spectrogram.

Shortly after the spectrogram appears, a series of waveforms will be displayed in the lower panel (3). These correspond to (downsampled) Hilbert envelopes of the outputs of a bank of auditory gammatone filters. The centre frequencies of the filters are shown on the left side of the display. It is advisable to limit the number of channels chosen to 10 or below unless you are dealing with short (< 1 second) signals. On some platforms, memory and compute speed becomes an issue with larger numbers of channels.

> [**ASIDE**: It is important to note that the filter bandwidths are NOT adjusted to ensure equal coverage of all spectral regions. Bandwidths are set to those defined in Glasberg & Moore (1990) and represent estimates of the effective frequency-dependent resolution of the auditory periphery. Future versions may allow proper treatment of this issue, but the redundancy of the speech signal ensures that even an 8 gammatone analysis gives a perfectly adequate 'clean'

baseline for subsequent distortions.]

The spectrogram display has a linear-in-Hz y-axis, whereas the filter centre frequencies (CF) are arrayed on an ERB-rate scale. The latter is approximately logarithmic.

You can listen to individual bands by shift-clicking on their waveform. [On platforms other than the Mac, this might involve using the right button]. Clicking on individual waveforms selects/unselects that signal. An unselected signal does not contribute to the overall output. By selecting/unselecting, you can explore various forms of spectral filtering. Alternatively, use the presets menu (2) to speed up the selection process.

Other distortions are controlled by the popup menus on the right of the display.

Popup menu (6) allows the envelope in each band to be replaced by a noise waveform, or by a constant (set to the mean of the envelope). Popup menu (7) allows the carrier to be a noise signal or a tone. The noise signal results from passing a wide-band noise through each gammatone filter. The tone frequency is set to channel CF.

Popup menu (8) specifies a maximum time shift (in ms) to be applied to each channel. The actual time shift applied is a pseuodrandom delay bounded by this figure.

In all cases, both displays are updated to show the effect of the distortion.

Distortions can be combined in arbitrary ways. Envelope and carrier modifications in each band are applied independently, and the resulting waveform in each band then is temporally-distorted.

Plenty of space exists to add further distortions!

Things to investigate

Since it is easy to 'hear out' utterances once you know what they are, it is advisable to load different utterances frequently, and to start with the most challenging conditions, gradually introducing more information until the utterance can be readily identified.

1. *Basic spectral filtering*. Listen to the effect of lowpass, highpass, bandpass and bandstop filtering. The latter condition has been investigated recently by Lippmann (1996) for consonant identification.
2. *Single bands*. Listen first to the lowest frequency band, then to the highest, then to one in the middle frequencies. Warren et al (1995) recently measured word identification performance for extremely narrow bands as a function of their CF, and found much better performance at 1500 Hz than at 300 or 6000 Hz. However ...
3. *Two bands*. ... Warren et al found supra-additive peformance when the low and high frequency bands are presented together.
4. *Carrier distortion*. Listen to the effect of replacing carriers by noise and by tones. Examine the spectrogram to see how information relating to both voicing and formant structure is disrupted. The noise carrier condition is related to a recent discovery by Shannon et al (1995), although that involved a smaller number of wider bands.

5. *Envelope distortion*.
6. *Temporal distortion*. Starting with the longest delays (1 second!), gradually reduce the amount of temporal distortion until the utterance can be identified. This signal modification is related to the temporal modulation filtering of Drullman (1995).
7. It is instructive to compare the effects of identical distortions on signals other than speech. Some music examples are provided in the standard distribution.

References

1. Allen (1994). *IEEE Trans. Speech & Audio Proc.*, **2**(4), 567-577.
2. Cooke (1996). *Proc. ESCA Workshop on Auditory Basis for Speech Perception*, Keele.
3. Drullman (1995). *JASA*, **97**(1), 585-592.
4. Fletcher (1953). *Speech & Hearing in Communication*. Van Nostrand.
5. Glasberg & Moore (1990). *Hearing Research*, **47**, 103-138.
6. Lippmann (1996). *IEEE Trans. on Speech & Audio Processing*, **4**(1), 66-69.
7. Shannon et al (1995). *Science*, **270**, 303-304.
8. Warren et al (1995). *Perc. & Psychophys.*, **57**(2), 175-182.

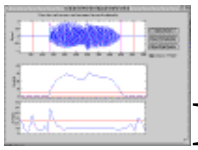
Further reading

- The ability of the auditory system to handle spectral filtering and reduction is one of the inspirations for missing data speech recognition. See the demo for further references to this literature.
- Listeners' resistance to multi-channel temporal distortion is one of the motivating factors behind the multi-stream approach to robust ASR. See Bourlard & Dupont (1997). ICASSP'97, 1251-1254.

Credits etc

Produced by: Martin Cooke

Release date: June 22 1998



Isolated Word Endpoint Detection

Introduction

An important problem in speech processing is to detect the presence of speech in a background of noise. This problem is often referred to as the endpoint location problem [1]. The accurate detection of a word's start and end points means that subsequent processing of the data can be kept to a minimum. Consider the speech recognition technique based on template matching. The exact timing of an utterance will generally not be the same as that of the template. They will also have different durations. In many cases the accuracy of alignment depends on the accuracy of the endpoint detections.
In order to perform well, the algorithm must take a number of special situations into account such as:

- Words which begin or end with low-energy phonemes (weak fricatives).
- Words which end with an unvoiced plosive.
- Words which end with a nasal.
- Speakers ending words with a trailing off in intensity or a short breath (noise).

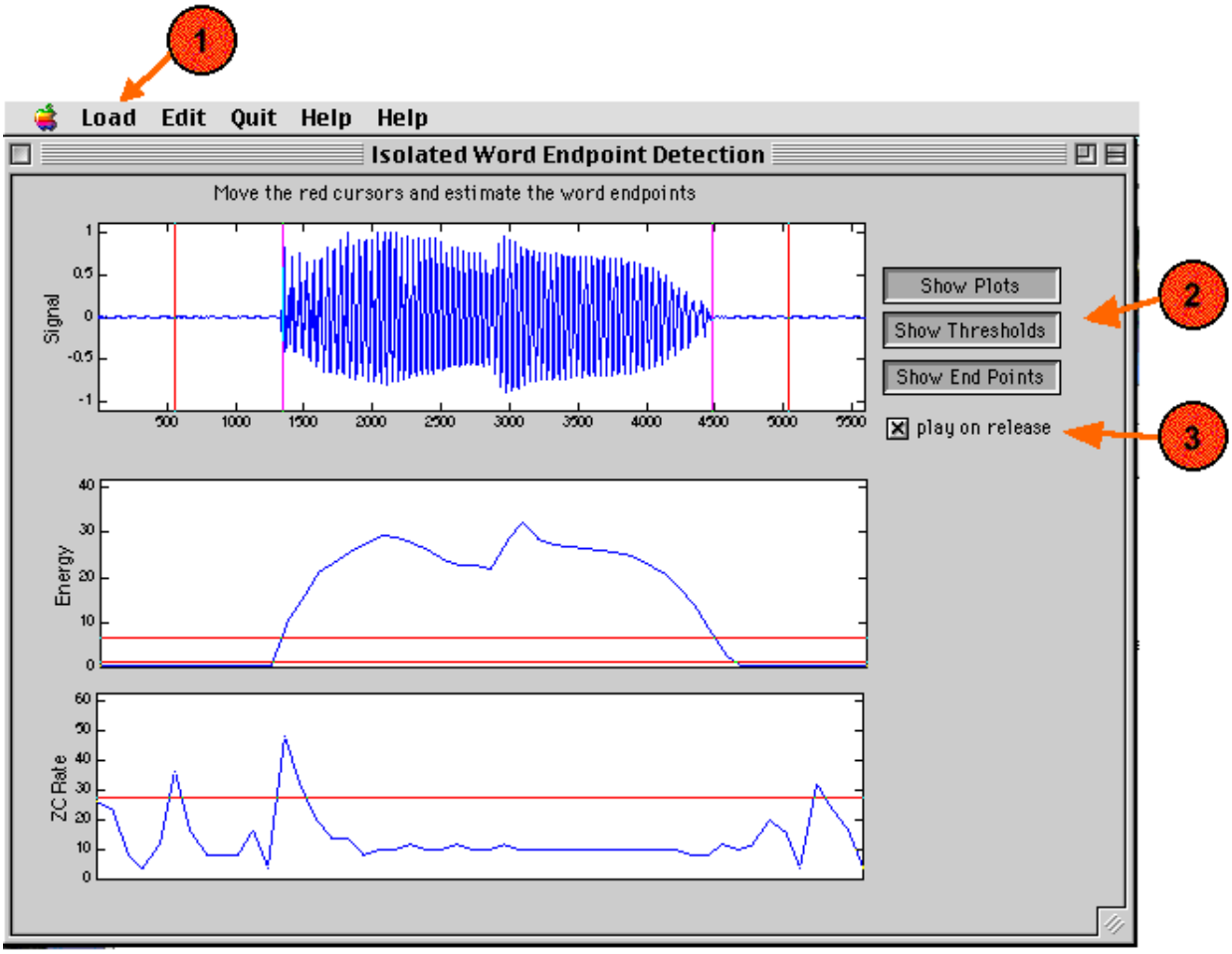
The method proposed in [1] and used in this demonstration uses two measures of the signal - the zero crossing rate and the energy.
Three thresholds are computed:

1. ITU - Upper energy threshold.
2. ITL - Lower energy threshold.
3. IZCT - Zero crossings rate threshold.

For more information on how these are computed, see Rabiner and Sambur [1].The method proceeds as follows. Search from the beginning until the energy crosses ITU. Then backoff towards the signal beginning until the first point at which the energy falls below ITL is reached. This is the provisional beginning point - N1. N2 (the end point) is selected in a similar way. For the beginning point, now examine the previous 250ms of the signal's zero-crossing rate. If this measure exceeds the IZCT threshold 3 or more times, N1 is moved to the first point at which the IZCT threshold is exceeded. N1 is defined as the beginning point. Again, perform a similar method for the end point N2.
For a more detailed explanation of the algorithm refer to Rabiner and Sambur [1].

Note: For the algorithm to perform correctly, the first 100ms of the speech signal must contain no speech.

The demonstration



Type 'epd' to launch the demo. When the window appears, use the load menu (1) to load a sound file. The signal can be played by clicking anywhere within the signal axes. The set of cursors allow the user to estimate their own endpoints. The option of playing the selected portion of the signal on release of the cursors is available (3). To the right of the axes (2), are 3 toggle buttons. The top two - Show Plots and Show Thresholds - provide the user with clues. The former displaying the energy and zero crossing rates and the latter providing even more information by showing the thresholds ITU, ITL and IZCT. With both of these buttons depressed, it is simple to find the endpoints. Pressing the third button - Show Endpoints - displays the computed endpoints in magenta on the signal axes.
The buttons can only be pressed in the order of increasing information displayed.

Things to investigate

1. Do your estimates always match well with the computed endpoints? If not, why do think this happens? (Consider the cases outlined above)

References

[1] Rabiner, L.R. and Sambur, M.R., "An Algorithm for Determining the Endpoints of Isolated Utterances". The Bell System Technical Journal, Vol. 54, No. 2, February 1975, pp. 297-315.

Further reading

Parsons, T.W., "Voice and Speech Processing". McGraw-Hill, 1987.

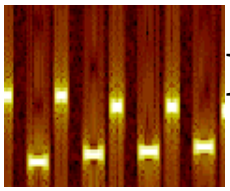
Rabiner, L.R. and Schafer, R.W., "Digital Processing of Speech Signals". Prentice-Hall, 1978.

Simple Time-domain Processing (**timedom**) MAD demonstration.

Credits

Produced by: Stuart N Wrigley

Release date: January 20 1999



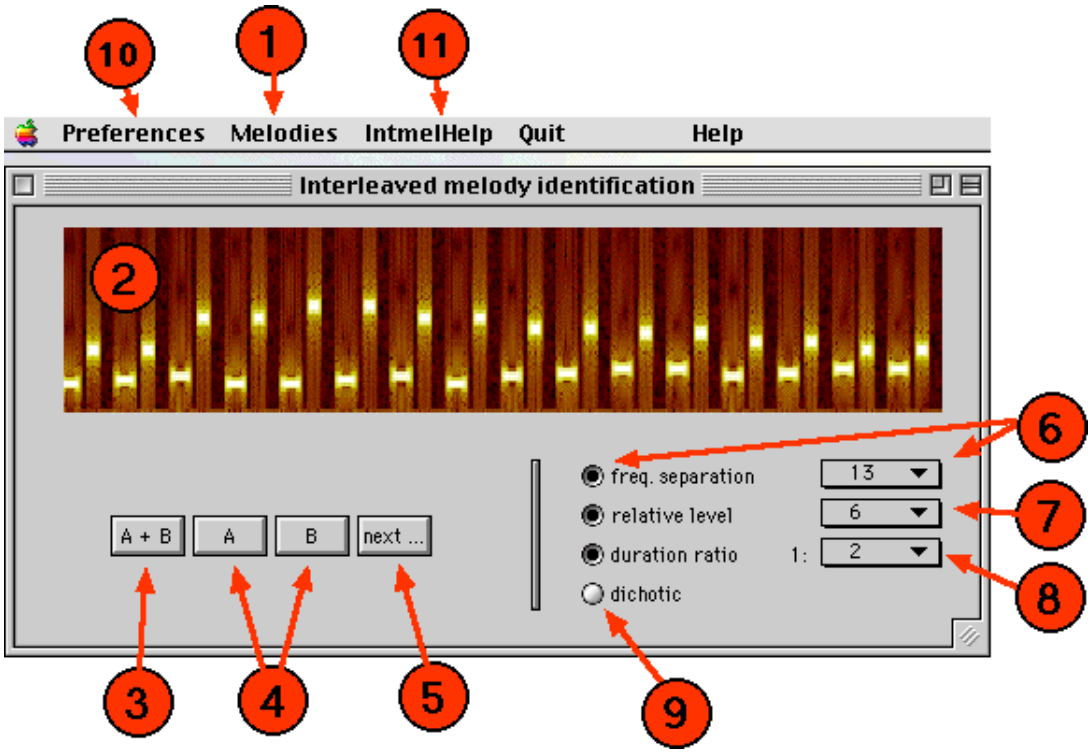
Interleaved melody identification

Introduction

The identification of familiar melodies is a task which engages stored representations or ’schemas’ (Bregman, 1990). However, melodies which are readily identifiable when presented in isolation can be much more difficult to recognise when interleaved -- that is, when notes from one melody alternate with notes from another (Dowling, 1973). Interleaved melodies are important in studies of auditory scene analysis since they can become much easier to recognise when tones from one melody are acoustically distinguished from tones from the other.

The extensive study of Hartmann & Johnson (1991) motivated the current demonstration (which uses a subset of both the melodies and the conditions used by them). They studied the identification of interleaved melodies whose elements differed in such factors as frequency range, ear of presentation, level etc. They were able to rank the effectiveness of 10 such factors in promoting melodic stream segregation.

The demonstration



Load the demonstration by typing 'intmel' at the MATLAB prompt. A dozen melodies are preloaded into the tool, so there is no need to load any further data [Hartmann & Johnson's paper lists many more melodies which we will get round to incorporating one day!].

The first step is to select which melodies you are familiar with (there is no point attempting to recognise unfamiliar tunes, although you can learn them on the fly if you like). The **melodies** menu (1) lists those loaded. Choosing any of these toggles its selection state. Initially, two melodies are indicated as being known. If you don't know these, unselect them. However, there must be at least two selected for the demonstration to be fully enabled, for obvious reasons.

When you are familiar with the melodies, you are ready to identify interleaved pairs. Pairs are delivered by pressing the **A+B** button (3). A spectrogram of the stimulus appears in the top panel (2). You may press this button as many times as you like. To hear the individual melodies, use buttons **A** and **B** (4). When you wish to hear another pair randomly-selected from the subset of melodies you have selected as familiar, press the **next...** button (5).

The default is to present the melodies in their base form, which means that they are identical in all key elements (except for the melody...). Hartmann & Johnson chose the base melodies to occupy similar frequency ranges, so identification is quite difficult in this condition. To examine the effect of potential cues to allow melody discrimination, use the options 6-9:

- *frequency separation*: a specified number of semitones can be added to one or other melodies to increase the mean frequency separation between the pair of melodies.
- *relative level*: use popup menu 7 to specify the relative level (in dB) between the two melodies.

- *duration ratio*: the duration of melody B tones relative to those melody A is specified by popup menu 8.
- *ear of presentation*: melodies can be send to different ears by selecting the dichotic radio button (9).

[Aside: in a future release, the radio buttons accompanying options 6-8 will be removed since they are redundant.]

The presentation rate (in tone pairs per second) defaults to 4, but can be changed via the preferences menu (10). Similarly, the tone duration can be altered from its default of 80 ms.

References

1. Bregman (1990). *Auditory Scene Analysis*. MIT Press.
2. Dowling (1973). *Cognitive Psychology*, **5**, 322-337.
3. Hartmann & Johnson (1991). *Music Perception*, **9**(2), 155-184.

Credits etc

Produced by: Martin Cooke

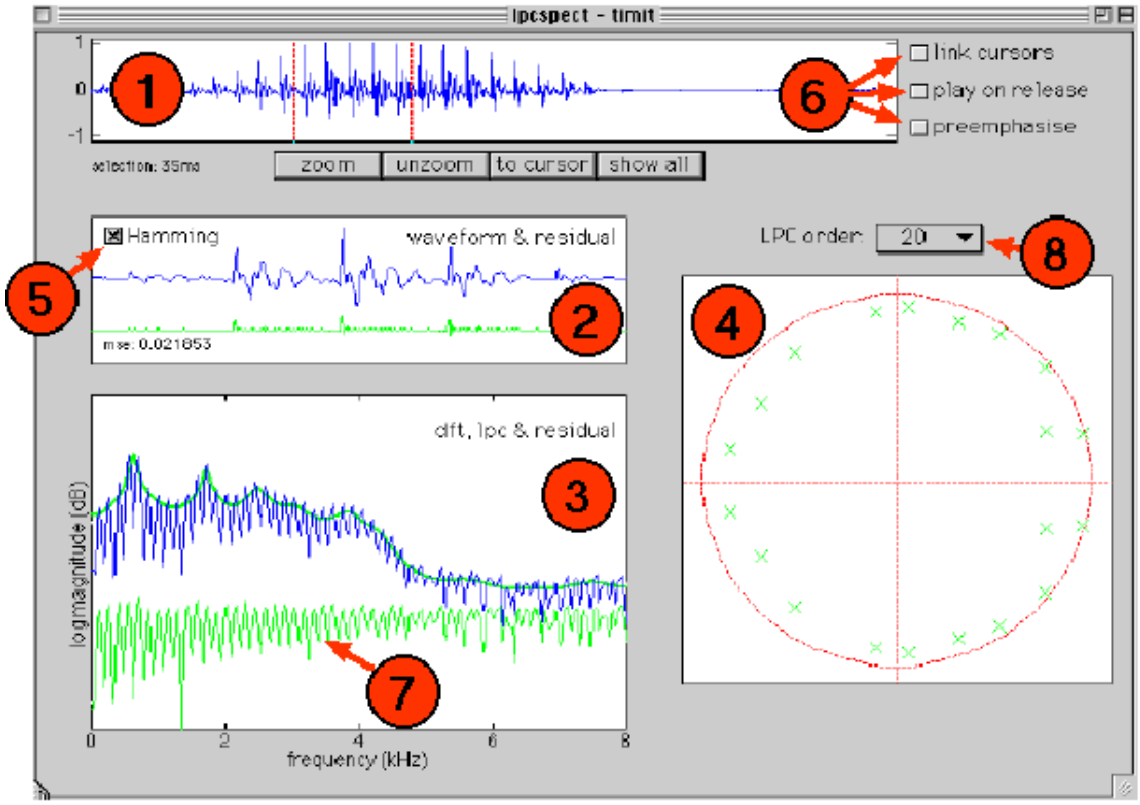
Release date: June 22 1998

lpcspect: linear prediction

Introduction

This tool allows users to explore linear prediction of speech and other signals. It also reinforces the concepts of windowing, preemphasis and frame size selection in speech analysis. Users can select regions for analysis and see the results as lpc-smoothed spectra and poles. The residual signal and its spectrum is also displayed. This tool is not a substitute for learning about linear prediction. To understand what the various panels mean, you should read one of the texts listed below. The demo requires the MATLAB signal processing toolbox.

The tool



Type 'lpcspect' to launch the demo. The *file* menu is used to load an existing signal file or to create a new one using the createsig tool. Supported formats currently include .wav, .snd and .au sound files. The signal will appear in panel (1), and associated linear prediction signals for the region between the cursors will be displayed in the other panels. Use the cursors to select any segment. On cursor release, the segment appears in panel (2) along with linear prediction residual (error signal). Panel (3) shows the DFT spectrum of this segment, overlaid with the LPC-smoothed spectrum and the DFT of the residual (7). The poles corresponding to the linear predictor are shown in panel (4).

The LPC order can be changed using menu (8). The signal can be preemphasised by checking the relevant box in cluster (6). Cursors can be linked (so that a constant segment length is maintained). The waveform segment chosen will be played on release if the associated checkbox is checked.

Clicking on any signal waveform (incuding the residual) causes it to be played. Holding the mouse button down in the z-plane panel (4) will cause the frequency at the selected z-plane angle to be displayed in the top left corner of the panel.

The chosen waveform segment can be Hamming-windowed (5). Unchecked, a rectangular window is applied.

Things to investigate

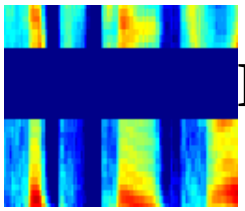
1. Choose a vowel segment of speech. Examine what happens to the residual signal (and its mean square error) as you increase the number of poles. What happens to the LPC-smoothed spectrum? How many poles provide a comfortable fit to a vowel segment.
2. Now repeat this analysis for unvoiced sounds such as fricatives. How does the error compare?
3. Again, perform this analysis for a nasal sound.
4. Can you predict where the poles will appear for a given analysis order, and how they will change when you modify the order?
5. Explore the effect of preemphasis on the LPC spectrum and the number of poles required to obtain a given residual error.
6. Listen to the waveform segment and the residual signal in sequence. Can you detect a change in timbre? How does this change manifest itself as the LPC order increases?
7. What is the effect of changing LPC order on the spectrum of the residual signal?
8. Choose a vowel sound and a relatively low LPC order (10, say). Read off the pole locations (frequencies) by holding the mouse down in panel (4). Now use the polezero tool to recreate just these pole locations and listen to the resulting sound.
9. Use the *create* option on the file menu to create a harmonic series with, say, 5 components. Do you expect to be able to fit it well with a 10th order model? Try it.
10. Now add some noise (using *create*) to your harmonic series. Do the pole locations change? (Read about the peak-hugging properties of LPC).

Further reading

- A good introduction to LPC is provided by Makhoul, J. (1975), IEEE, 63, 561-580.
- This tool was written partly as an example of the interface issues in speech and hearing demonstrations, and is described further in Cooke et al (1999) The interactive auditory demonstrations project, Eurospeech'99, Budapest, September.

Credits etc

Produced by: Martin Cooke, April 1999.



Missing Data ASR

Introduction

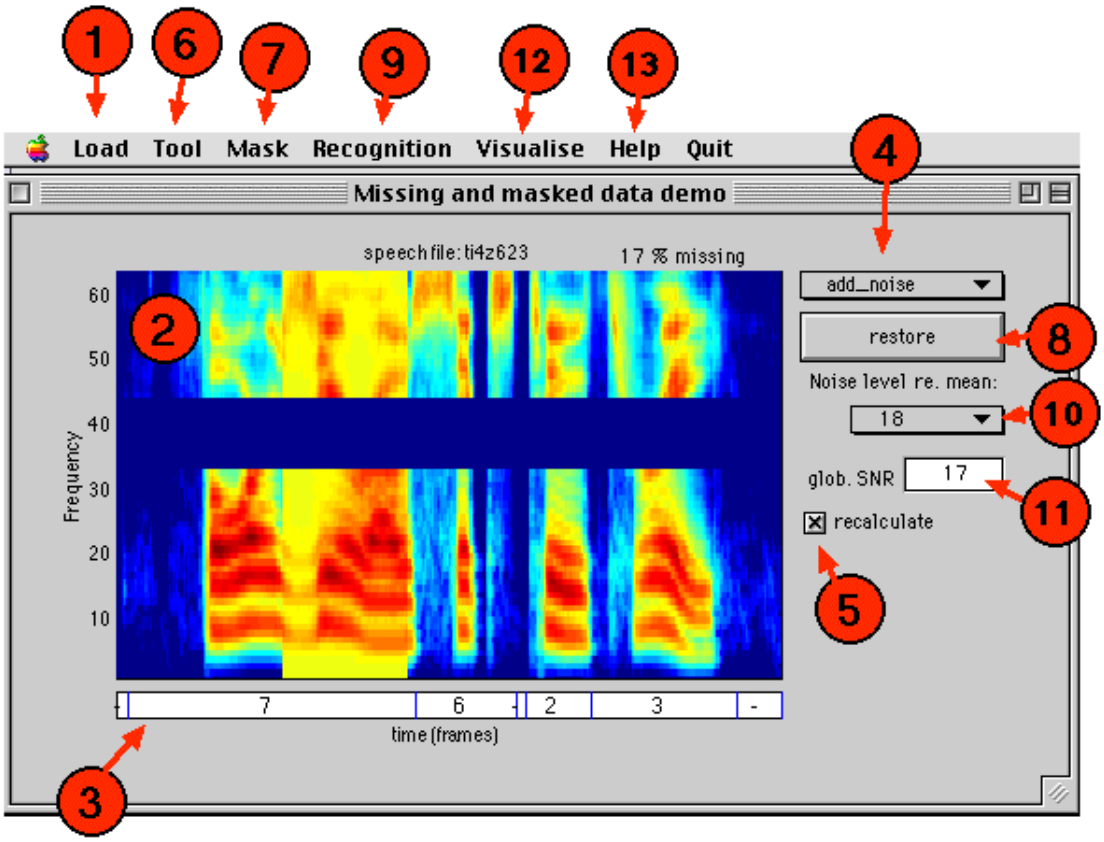
Listeners are capable of perceiving speech in the face of quite severe distortions (see demo). Common aspects of many distortions include the absence of spectro-temporal regions, or the presence of additive noise. Missing data occurs naturally in conditions such as telephone speech and channel fade-outs. Missing data speech recognition is a recent approach (Cooke et al, 1994) to robust ASR based on solutions to the following to subproblems:

- identification of reliable evidence; and
- adapting recognisers to handle missing and masked data.

Solutions to the first subproblem range from existing noise-reduction techniques such as spectral subtraction and local SNR estimation (Hirsch, 1993) through to more general-purpose approaches such as computational auditory scene analysis (Rosenthal & Okuno, 1998).

One approach to the second problem is demonstrated here. The purpose of the demo is to allow the user to explore the effect on recognition of removing spectro-temporal regions or adding noise to speech.

The demonstration



Type ’md’ at the MATLAB prompt. Use the speech submenu of the load menu (1) to select a file to load. The speech files supplied with the demo are strings of digits (length 4 or 5) from the TIDigits corpus (ref). The identity of the digit sequence is apparent from its name.

> [**Aside**: File names are of the form *ti9o82*, which signifies that this is the string "nine-oh-eight-two" from the TIDigits corpus. Recognition models exist for the 9 digits 1-9, together with ’o’ for ’oh’, ’z’ for ’zero’ and ’-’ indicating silence. For this demonstration, recognition is performed using rather simple single mixture HMMs with no use of derivatives, so performance is not perfect even without deletions!]

On loading, the system displays an auditorily-motivated spectrogram which is used as the sole basis for recognition (2). After a short while (longer on slow systems, but a few seconds on a PowerMac running at 200 MHz), the recognised result will appear in the lower panel (3). Displayed boundaries are those obtained by tracing back through the best sequence of models in the usual way.

EDITING THE DISPLAY

At this point, clicking anywhere on the spectrographic image results in an editing function determined by the popup menu (4). For instance, if the mode is **erase**, an area under the cursor is removed from the display each time the user clicks on the spectrogram. At this

point, it is worth remarking that the demo acts like a spreadsheet: changes made restart the recogniser, which updates the recognised result 'almost instantaneously'. If you are using a slow machine (defined here as one which takes more than a few seconds to perform the recognition), you may wish to untick the *recalculate* checkbox (5) until you wish recognition to take place.

The extent of the region affected by clicking in the display is governed by the tool menu (6). Here, you can select the region affected by an edit operation by specifying the frequency region (in channels) and the duration (in 10 ms time frames).

Other edit options are **add_noise**, which adds noise at the level specified (relative to the signal mean) by popup menu (10), and **restore**, which restores the original values. In addition, the whole display can be reset at any time by pressing the **restore** button (8).

Removing or restoring data has the obvious effect on the display. A status line at the top right of the display indicates the percentage of missing time-frequency regions. The visual effect of adding noise depends on the noise level relative to the speech background. The global SNR is changed by any noise addition, and is indicated on the display. In addition, the user can set a required global SNR and the noise background will be modified to meet this criterion (11).

RECOGNITION

The manner in which the missing and masked data is recognised is controlled by the options available under the **mask** (7) and **recognition** (9) menus. The default recogniser is a conventional strategy which treats the data as being complete. That is, regions deleted by the user are treated as having value zero (energy). Similarly, added noise is part of the data fed to the recogniser. Setting the **missing data** option via the recognition menu instructs the recogniser to treat certain regions as missing. In the simplest case, these regions are just those which the user has deleted from the display. However, for the case of added noise, the user can force the recogniser to treat those regions with a negative local SNR as missing by selecting the **snr > 0** option via the mask menu. The display is immediately updated to reveal those regions which meet this criterion.

A further recognition option is to use just spectral peaks (or the subset of them not missing or masked by noise) in the recognition process. This option is obtained via the **peaks** options under the masks menu. Again, the display is updated to show the points upon which recognition takes place.

INSPECTING THE HMMS AND OUTPUT PROBABILITIES

Certain aspects of the recognition models and process can be visualised (12):

- **Visualise>HMMs** plots the means/variances in each HMM state
- **Visualise>Output probs** displays the per-frame output probabilities. This is updated when changes are made to the recogniser's input.

PRESET DELETIONS AND NOISE

It is possible to apply certain deletion patterns without using the mouse to select regions. These are available under the load deletions submenu and include

- random deletion at the 50% and 90% level
- random removal of frames or channels at 50% or 90%
- removal of all low energy points
- lowpass, bandpass or highpass filtering
- removal of all but a couple of single frequency bands

In addition, noise patterns (at present, these are just other speech files) can be loaded from the **load>noise** submenu.

In the near-future, the demo will be enhanced to include auditory induction effects and perhaps include a range of automatic local SNR techniques.

Things to investigate

1. Load a speech file and observe the recognition result. Apply a preset deletion pattern. Again, look at what happens to the recognition performance. Turn missing data recognition on. This ought to improve the recogniser's performance.
2. Examine the effect of adding noise. Again, load a speech file, but this time add some noise patches with the mouse. Recognition should deteriorate (if it doesn't, you're not adding enough noise!). Then, set the display to **snr > 0** from the **Mask** menu and switch missing data recognition on. This ensures that the system uses only those points whose SNR is positive.

References

1. Cooke et al (1994). *ICSLP*, 1555-1558.
2. Hirsch & Ehrlicher (1995). *ICASSP*, 153-156.
3. Rosenthal & Okuno (1998). *Computational auditory scene analysis*. LEA.

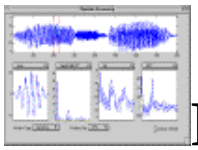
Further reading

- An easy introduction to the technique is given in Cooke et al (1997), ICASSP, 863-866.

Credits etc

Produced by: Martin Cooke

Release date: June 22 1998



Pipeline Processing

Introduction

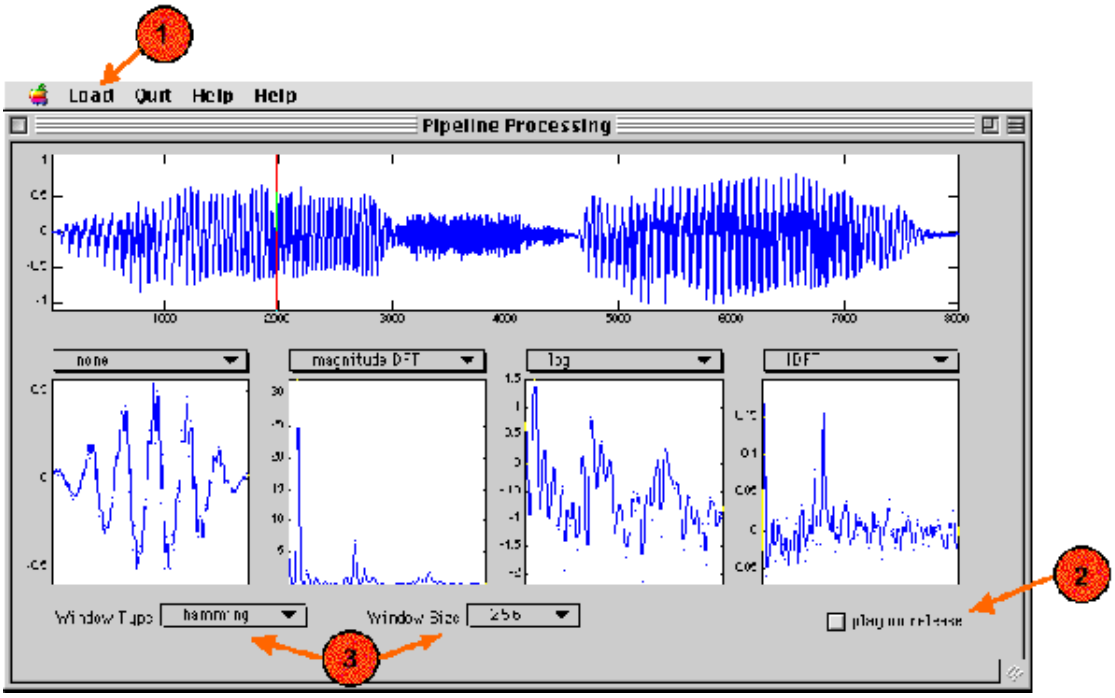
Many alternative representations of (speech) signals require a number of basic operations to be performed in sequence. One way of thinking about such sequential processing is as a pipeline. Generally, the start involves the windowing of the signal to select a specific part. Each operation is then performed on the output of the previous point in the pipeline.

This demonstration provides 4 pipeline windows with a choice of 4 possible operations each:

- none
- magnitude discrete fourier transform (DFT)
- inverse DFT (IDFT)
- log

As might be expected, *none* does not perform an operation on the previous output with one exception: when *none* is selected in the first pipeline window, the output is in fact the **windowed** segment selected by the cursors.

The demonstration



Type ’pipeline’ to launch the demo. When the window appears, use the load menu (1) to load a sound file. The signal can be played by clicking anywhere within the signal axes. The set of cursors allow the user to estimate their own endpoints. The option of playing the selected portion of the signal on release of the cursors is available (2). Various windowing types and sizes are available (3).
The dropdown menu above each of the pipeline windows is used to select the operation to be performed at that point in the pipeline.
To force one of the pipeline axes to rescale, reselect the existing output from the dropdown menu for that window.

Things to investigate

1. Use the pipeline to create a cepstral slice. What does the peak in the cepstral slice signify?
   What happens to this peak when the cursors are placed over a fricative? How do you explain this?
2. Use the pipeline to create a power spectrum. What does the fine structure of the output represent?
   What does the overall shape represent?
3. Does altering the window type alter the plots? If so, why?

References

[1] Parsons, T.W., "Voice and Speech Processing". McGraw-Hill, 1987.

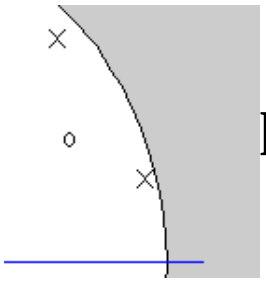
Further reading

Cepstral Liftering (**ceplift**) MAD demonstration.

Credits

Produced by: Stuart N Wrigley

Release date: November 1998

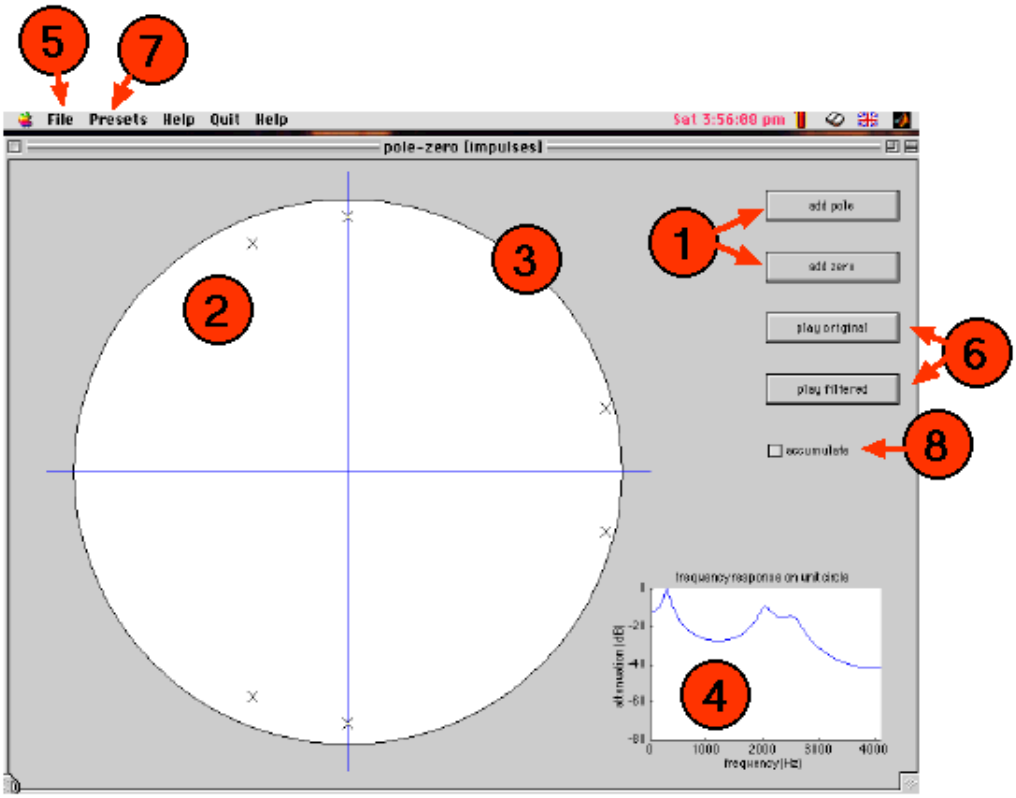


polezero

Introduction

This tutorial tool allows users to create linear systems via a pole-zero diagram. The magnitude response of the linear system is displayed and updated as the user drags individual poles or zeroes . The linear system can be applied to signals and the filtered and unfiltered signals can be played back. Tones, uniform noise and impulse trains can be generated. In addition, preset pole-zero diagrams (in fact, all-pole systems) for vowels can be loaded.

The tool



Type ’polezero’ to launch the demo. You will be presented with a display dominated by the unit circle. Initially, it contains no poles or zeroes. Use the add pole or add zero buttons (1) to add new poles or zeroes to the figure (2, 3). They appear in conjugate pairs. You can move them by selecting and dragging, and delete them by selecting and hitting the delete (or similar) key. Dragging either one of a pair does the appropriate thing to its partner.

As they move, the magnitude response display (4) is updated to show the response evaluated on the unit circle. [At present, no phase response is shown, but it could be easily added]. The response of the system is normalised with a maximum of 0 dB.

To plug a signal into the linear system, use the file menu (5). From here, you can load sound files, or generate tones, uniform noise or impulse trains. Once an input signal is available, the playback controls (6) are enabled, allowing playback of the input and output of the linear system. Note that the sampling frequency reflects that of the most recently loaded signal, and the interpretation of the unit circle and magnitude response axis changes accordingly.

Signals can either replace those already loaded, or by added in. To accomplish the latter, click the accumulate checkbox (8). If a generated signal is loaded *after* a soundfile, it will have a duration to match the soundfile.

Preset pole configurations can be loaded using the presets menu (9). Currently, presets for selected vowel sounds are available.

Things to investigate

1. Using a single pole pair, observe the effect on the frequency response as you drag it. What happens as it nears the unit circle? What is the relationship between the peak in the frequency response and the location of the pole pair?
2. Add a further pole pair and experiment with its position.
3. Now add a zero pair. Again, what happens when the zero nears the unit circle?
4. Load in a speech signal, then tick the ’accumulate’ checkbox and load a tone. Listen to the signal you obtain. The tone should be quite loud in relation to the speech signal. Now attempt to build a linear system which reduces (or removes completely) the effect of the tone.
5. Load one of the vowel presets. Move one of the poles and observe the effect on the spectrum. Note how the bandwidth of each ’formant’ is related to the associated pole’s distance from the unit circle. To listen to the vowel, you need to load a suitable excitation signal. Start with a noise signal and listen to the filtered response. Does it sound like the specified vowel? Load a different preset vowel to hear the difference. Now load an impulse series. The filtered vowel should now sound more like a voiced sound. You can experiment with the impulse repetition rate to change the fundamental frequency of the vowel.
6. Load in a sentence and use the pole-zero editor to create a frquency response with a high-pass shape (that is, one rising from low to high frequencies). What effect does this have on the speech signal? Now produce one with a low-pass shape (falling from low to high frequencies).
7. Simulate telephone bandwidth speech.

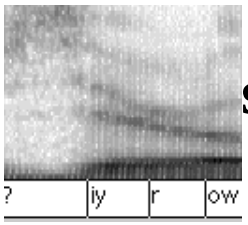
Further reading

- Any linear systems textbook (e.g. Lynn) will contain examples of pole-zero diagrams for you to implement here.

Credits etc

Produced by: Martin Cooke, based loosely on Perry Cook's (no relation :-) Resolab application on the NeXT.

Release date: October 17th 1998

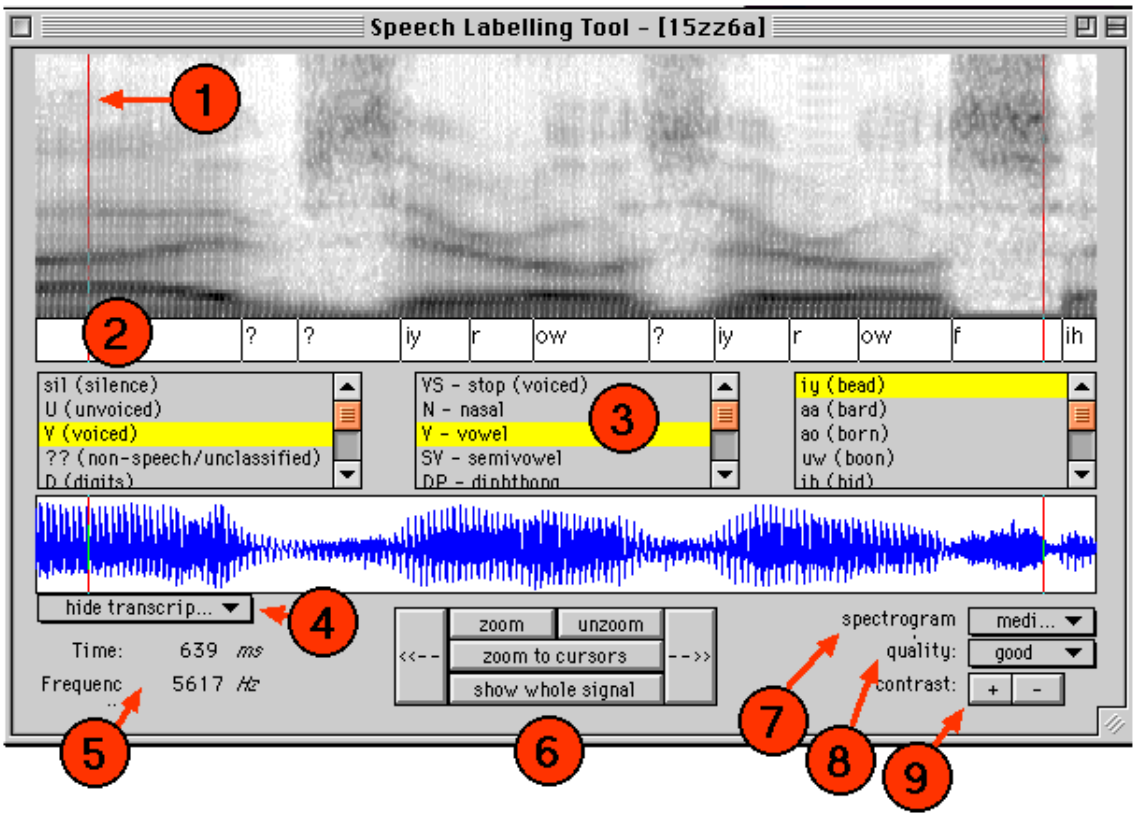


slt: speech labelling tool

Introduction

This tool allows users to display speech signals and their associated spectrograms and transcriptions. Users can create new transcriptions, edit existing ones, and view reference transcriptions. Narrow, medium and broadband spectrographic displays are supported, as is playback of signal segments. A hierarchical transcription browser is incorporated.

The tool



Type ’slt’ to launch the demo. When the window appears, use the load menu to load a speech signal file. Supported formats currently include .wav, .snd and .au sound files. The signal and its spectrogram will appear. The type and quality of the spectrogram can be

altered via the controls (7-9) in the lower right of the tool. Medium, broad and narrowband spectrograms can be produced, and the quality of the display can be varied. Initially, the spectrogram quality is set to rough (the fastest to compute), but is probably best viewed at good quality. The brightnesss/contrast of the spectrogram can be altered using the buttons at (9). Increasing the contrast will also remove the lower amplitude regions from the display, which can be useful when looking for speech-related events.

If any existing transcriptions are available, the menu (4) will appear. It defaults to 'hide transcription', but the user can select any of the available transcriptions for this utterance to be displayed in the waveform window. Note that any transcriptions displayed here cannot be edited, and are for 'reference' or comparison purposes.

To **listen** to a the signal, click the mouse anywhere in the signal or spectrogram windows. If the mouse click is between the cursors (shown as red vertical lines -- (1) above), only that segment will be heard. If a user transcription (as opposed to a reference transcription) is loaded, clicking between pairs of transcription markers will play out just that segment. This can be useful during labelling.

Us the control pad (6) to modify which segment is displayed. The '-->' and '<--' buttons move the portion of the signal displayed left and right (by 40%).

As you move the cursor over the displays, the time and frequency under the current mouse location is shown at the lower left of the display (5). The frequency varies when the mouse is over the spectrogram (whose upper frequency is set to 6 kHz). This can be useful for extracting measurements (eg of formant frequencies) from the spectrogram.

Transcription is performed in the user transcription window (2). Transcriptions can be loaded via the file menu, or developed from scratch. To insert a new transcription marker, double-click near one of the two cursors. A new boundary marker, and '?' label will appear in the transcription window at the location of the cursor clicked. The new segment can be selected, moved, labelled or deleted:

- A label is *selected* by clicking near the boundary or label. The boundary will be highlighted.
- *Deletion* is performed by selecting the object to the deleted, then hitting the delete key.
- Labels are *moved* by dragging with the mouse.
- To *add* or *modify* a label, the label browser (3) is used to select a particular label, whereupon double-clicking in the label browser will modify the selected transcription label. Any label in the browser can be selected, whether it be a fine phoneme label or a broad class or other. This allows various types of spectrogram labelling.

Mystery sounds (those whose name starts with 'myst') disable the play commands.

Things to investigate

1. If you are new to spectrogram reading, start off by identifying voiced, unvoiced and silent regions of the signal. Use the top-level of the label browser to apply these labels. Compare the cues available in the broad and narrowband spectrogram displays. Choose

an example to load in which reference transcriptions at the voiced-unvoiced-silence level are available (the annotation file will probably have the .suv extension), and reveal them when you've had a go yourself.
2. How easy is it to spot word boundaries from the signal alone?
3. Load an example which has phoneme labels, and compare the spectrographic display of these with those contained in any textbook on acoustic-phonetics (e.g. Ladefoged; Rabiner & Juang).
4. Find the vowel portions and measure the frequencies of the first 3 formants. Compare these with tabulated values in the above texts.
5. Explore co-articulation.
6. Load one of the mystery spectrograms. In the current release, these are digit sequences containing 4 or 5 digits. Have a go at identifying them.

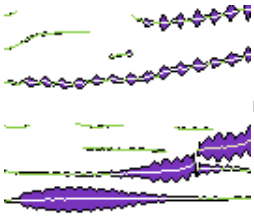
Further reading

- You will get most out of this tool if you have a suitable acoustics-phonetics text handy. Ladefoged's course in phonetics or any speech technology book (eg Deller, Proakis & Hanson; Rabiner & Juang) will suffice.
- For other visualisations of speech signals, see the edited collection in Cooke, Beet & Crawford (1993). *Visual Representations of Speech Signals*. Wiley.

Credits etc

Produced by: Stuart Wrigley and Martin Cooke

Release date: October 5th 1998

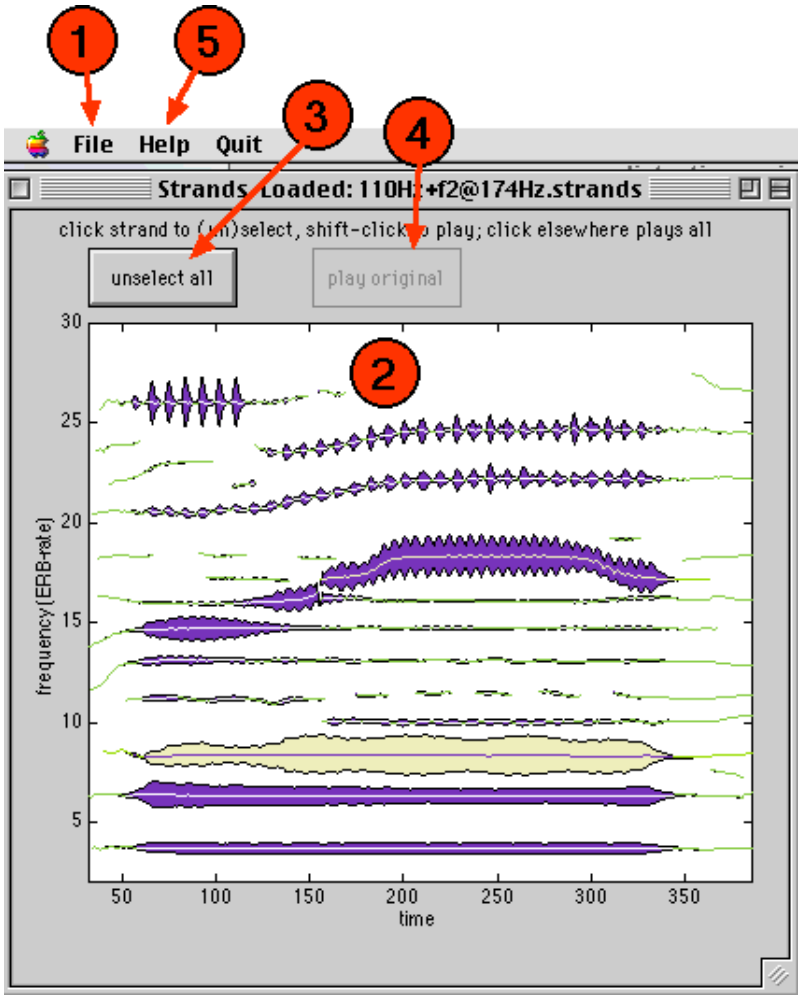


Strands: auditory time-freq rep

Introduction

Strands (Cooke, 1991) are time-frequency descriptions which attempt to capture the important features of speech signals in a form amenable to the application of auditory grouping principles (Darwin & Carlyon, 1995; Bregman, 1990). Individual strands attempt to track individual spectral dominances through time. Such dominances can correspond to resolved harmonics, formants, or other relatively-narrowband components of signals. Strands can be converted to sound (’resynthesised’), individually or collectively.

The demonstration



Type ’strands’ to launch the demo. When the window appears, use the load menu (1) to load a strands file. If the associated sound file exists, the ’play original’ button (4) will be enabled. Once the strands are loaded, they can be switched on or off by clicking. Clicking elsewhere in the display panel (2) plays out the sound resulting from adding the selected strands resyntheses together. Button (3) makes selection or unselection of all displayed strands easier.

Shift-clicking (on the Mac, at least - try right buttons on other platforms?) on individual strands plays them out.

Things to investigate

1. The starting point for strand production is a model of the auditory periphery based on gammatone filters (Patterson & Holdsworth, 1990). A consequence of the frequency-dependent bandwidth of the modelled auditory filterbank is the resolution of harmonic components at low-mid frequencies, and the lack of resolution (leading to the appearance of formants) at mid-high frequencies. Load any strands file to observe this distinction.
2. One set of strand examples provided with the distribution come from the "ru"-"li"

stimuli generated by Chris Darwin (Darwin, 1981; Gardner et al, 1989) and examined computationally in Cooke (1991). These are synthetic syllables with 4 formants. In one condition, the formants are synthesised on the same F0 (110 Hz). In the other conditions, the second formant has a different fundamental from the other 3 (which remain on 110 Hz). The effect of this F0 difference can be seen by loading the associated strand files, which have names like *ru112* (signifying an F2 on 112 Hz). By manually removing the F2 strand(s), you can hear "li" instead of "ru".

3. A subset of the corpus used to evaluate speech separation in Cooke (1991) is available in the standard distribution. Relevant files have names starting with *v0*. File *v0* itself is a single-source utterance, while *v0n** denote *v0* with added 'noise'. By manually attempting to group speech strands, you can get an idea of what performance might be possible by a system for grouping based on strands. [It is hoped that something similar to Cooke's original system will be implemented as a demo in the next release].

References

1. Bregman (1990). *Auditory Scene Analysis*. MIT Press.
2. Cooke (1991). *Modelling auditory processing and organisation*. PhD Thesis. Published by Cambridge University Press, 1993.
3. Darwin (1981). *Q. Jnl. Exp. Psych*, **33**(1), 185-207.
4. Darwin & Carlyon (1995). *Auditory Grouping*. In: *Hearing*, Academic Press, 387-424.
5. Gardner et al (1989). *JASA*, **85**(3), 1329-1337.
6. Patterson & Holdsworth (1990). In: *Adv. in Speech, Hearing & Language Proc*., Vol. 3 (ed: Ainsworth), JAI Press.

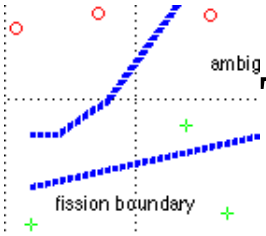
Further reading

- For other visualisations of speech signals, see the edited collection in Cooke, Beet & Crawford (1993). *Visual Representations of Speech Signals*. Wiley.

Credits etc

Produced by: Martin Cooke

Release date: June 22 1998



Tone streaming

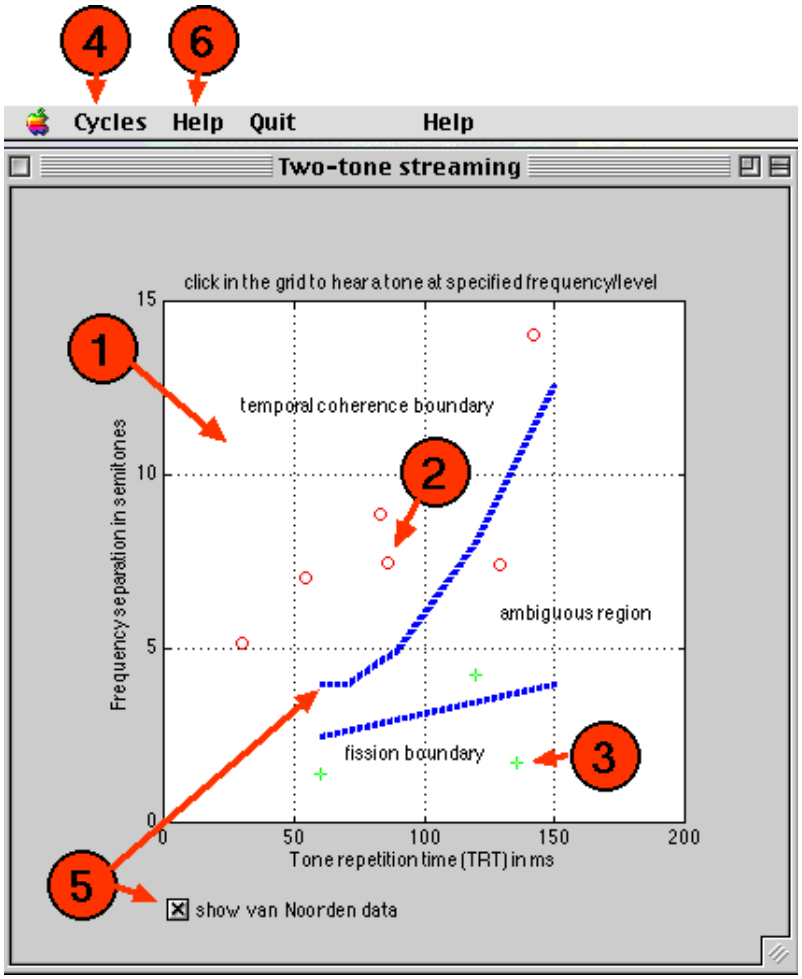
Introduction

Repeated sequences of tones can be heard as belonging to a single perceptual organisation (fusion) or to two groupings (fission). The frequency difference between successive tones plays a key role in determining which organisation prevails, as first demonstrated by Miller & Heise (1950) using alternating pairs of tones with increasing frequency separation. The point at which the sequence 'breaks' into two is termed the 'trill threshold'.

Van Noorden (1975) investigated the role of other factors such as presentation rate in streaming. In one condition, he used repeated HLH_ stimuli, where H and L represent high and low frequency tones, and _ indicates a silent interval of the same duration as the H and L tones. This stimulus uses rhythmic change as an indicator of fission or fusion. If the low and high tones fuse, listeners should hear a galloping (HLH_) rhythm. However, if the L and H tones segregate, two isochronous rhythms should appear. One, formed by the H tones, is twice as fast as the other, formed by the L tones.

Van Noorden discovered that some conditions produced an 'ambiguous' percept: i.e. listeners could choose to hear fusion or fission, and switch between these percepts.

The demonstration



Launch the demonstration with the command 'streamer'. Clicking anywhere in region 1 results in the delivery of a stimulus with the tone repetition time and high-low frequency separation specified by the position in the grid. The number of cycles delivered is governed by a menu option (4). Ten or more cycles may be required to hear one or other organisation.

After some practice at hearing the two rhythmic possibilities, you may wish to record your responses. After each stimulus presentation, pressing 's' or 'f' will result in a red circle (2) or green cross (3) appearing at the click location. After a number of such identifications, compare your responses with those summarising van Noorden's subjects by checking the checkbox (5).

References

1. Miller & Heise (1950). *JASA*, **22**, 637-638.
2. Van Noorden (1975). PhD Thesis, Eindhoven University of Technology.

Further reading

- Bregman's book (*Auditory Scene Analysis*, MIT Press, 1990) contains an extensive

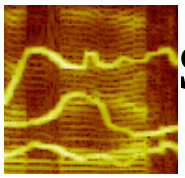
discussion of auditory stream segregation.

- A number of computational models which attempt to explain fusion/fission have appeared lately:
  - Beauvois & Meddis (1995). *JASA*
  - Brown & Cooke (1998). Ch 7 in Computational Auditory Scene Analysis (ed: Okuno & Rosenthal, LEA).
  - McCabe & Denham (1997). *JASA,* **101**(3), 1611-1621.
  - Todd (1996). *Network: Computation in Neural Systems*, **7**, 349-356.
- Various theories have been put forward to explain these and other streaming effects. Some of these are discussed in:
  - Rogers & Bregman (1993). *Perc. & Psych*., **53**(2), 179-189.
  - Hartmann & Johnson (1991). *Music Perception*, **9**(2), 155-184.

Credits etc

Produced by: Martin Cooke

Release date: June 22 1998



Sine-wave speech cocktails

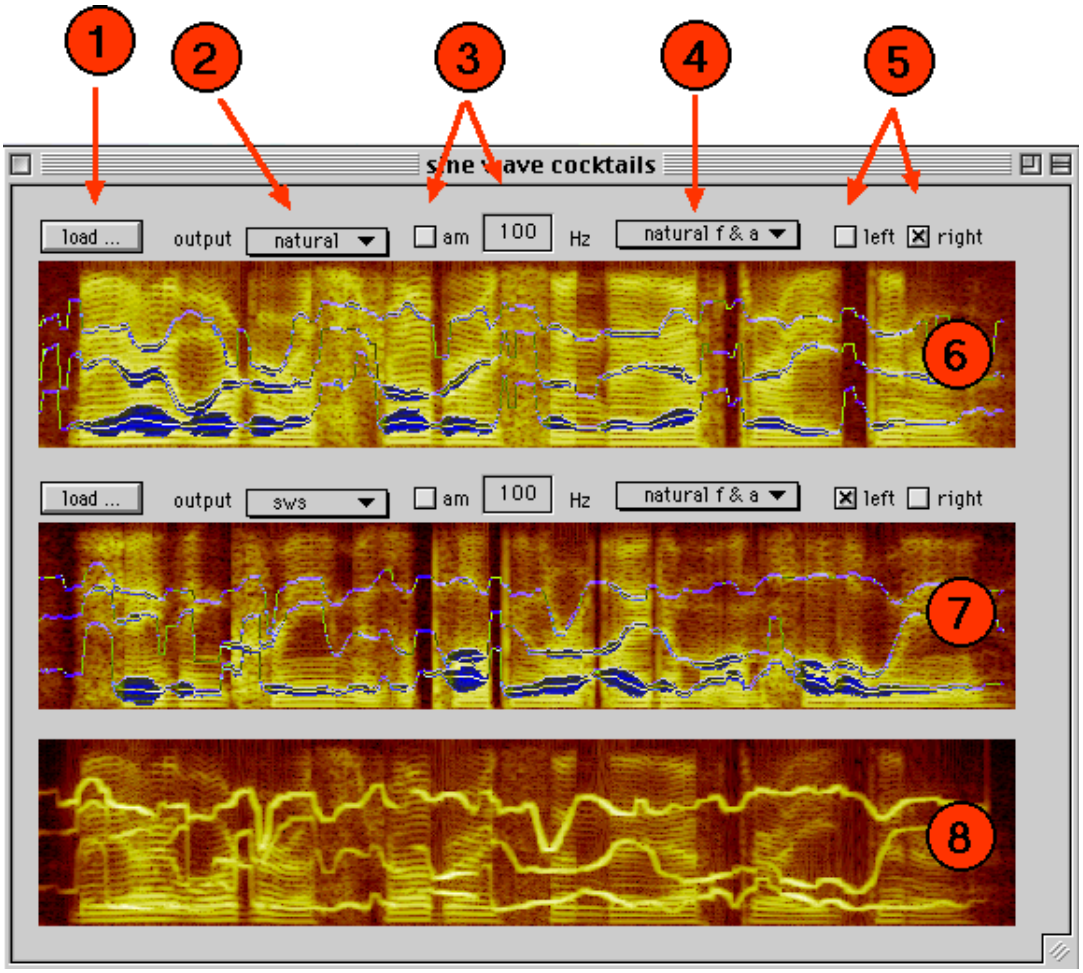
Introduction

Sine-wave speech (SWS - Bailey et al, 1977; Remez et al, 1981) is a synthetic analogue of natural speech produced by a small number of time-varying sinusoids. Listeners perform well at transcribing sine-wave replicas of utterances (Remez et al, 1981; Barker & Cooke, in revision). It has been argued that SWS demonstrates the special status of speech in auditory perception (Remez et al, 1981). Of late, proponents of this hypothesis have used SWS experiments (Remez et al, 1994) to suggest that speech is beyond the reach of the 'gestalt' grouping processes which motivate the auditory scene analysis (ASA) account of sound perception (Bregman, 1990).

The present demonstration was motivated primarily by studies into the perception of simultaneous sine-wave speech utterances (Barker & Cooke, 1997). In these experiments, listeners were asked to transcribe pairs of sine-wave sentences presented simultaneously. Results were compared against (phoneme-level) transcription scores for pairs of natural utterances.

Other experiments have examined the effect of dichotic presentation (Remez et al, 1994), reduced numbers of sine-wave 'formants', further reduction of SWS to constant amplitudes or frequencies (Remez & Rubin, 1990) and the role of amplitude modulation (Carrell & Opie, 1992; Barker, 1998). The demonstration allows all of these manipulations to be explored.

The demonstration



The demo is launched with the command ’sws’, which brings up a window similar to the one above (without the spectrograms, initially). The window contains three display panels (6,7,8). The top two (6,7) are used to display spectrogams and SWS tracks for a pair of utterances, which are loaded via the buttons (1). The lower panel (8) displays a spectrogram of the mixture.

Once spectrograms and SWS tracks are loaded, clicking on the spectographic image results in the associated signal being played. SWS formants can be selected and unselected by clicking on the tracks. Unselected formants do not contribute to the sound output, and their absence can be noted in the mixture spectrogram.

Popup menu (2) selects which signal is used in playback. Options are ’natural’, ’SWS’ and ’silent’. The latter prevents the signal from contributing to the mixture.

Amplitude modulation can be added to the SWS waveform. If checkbox (3) is checked, AM at the specified rate is applied to the SWS signal. Sidebands will be visible in the mixture for all but the lowest rates of AM.

By default, SWS tracks use frequency and amplitude values extracted from the natural utterance (for details on the mainly-automatic procedure used, see Barker, 1998). Optionally,

via popup menu (4), the listener can select constant amplitude or constant frequency SWS tracks.

Finally, the two signals can be presented diotically or dichotically via checkboxes (5).

The distribution comes with a selection of 20 SWS + natural utterances. The natural utterances come from the TIMIT CDROM.

Things to investigate

1. If you have never heard SWS before, load one of the example utterances and select SWS output from (2). Can you identify the utterance? Before listening to the natural utterance, try applying amplitude modulation (3). Does this make things easier?
2. Choose a different utterance, and attempt to identify it in some of these more difficult conditions:
    - with one or more formants unselected
    - with constant amplitude or frequency
    - with an interfering SWS utterance
3. If the SWS cocktail is too lethal, try the following:
    - apply AM to one of the signals
    - apply different rates of AM to the two signals
    - send the signals to different ears (on the same head)
    - use a single SWS formant as a distractor for the other complete SWS signal.
4. Listen to pairs of natural utterances. Note the ease of separation when presented dichotically, as noted long ago by Cherry (1953).
5. Examine the spectrogram of a single SWS source (you will have to turn one of the two off at this point). What cues present in the natural speech spectrogram are missing?
6. Examine the spectrogram of the mix. Are there any cues which might be used to separate the source without speech knowledge? [See Barker (1998) for some suggestions].

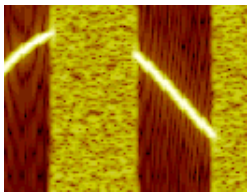
References

- Bailey et al (1977). *TR SR-51/52*, Haskins Labs.
- Barker & Cooke. *Speech Communication* (in revision).
- Barker (1998). *PhD Thesis*, University of Sheffield.
- Bregman (1990). *Auditory Scene Analysis*. MIT Press.
- Carrell & Opie (1992). *Perc. & Psych.*, **52**, 437-445.
- Cherry (1953). *JASA*, **25**, 975-979.
- Remez et al (1981). *Science*, **212**, 947-950.
- Remez & Rubin (1990). *Perc. & Psych.*, **48**(4), 313-325.
- Remez et al (1994). *Psych. Review*., **101**(1), 129-156.

Credits etc

Produced by: Martin Cooke. SWS data provided by Jon Barker.

Release date: June 22 1998

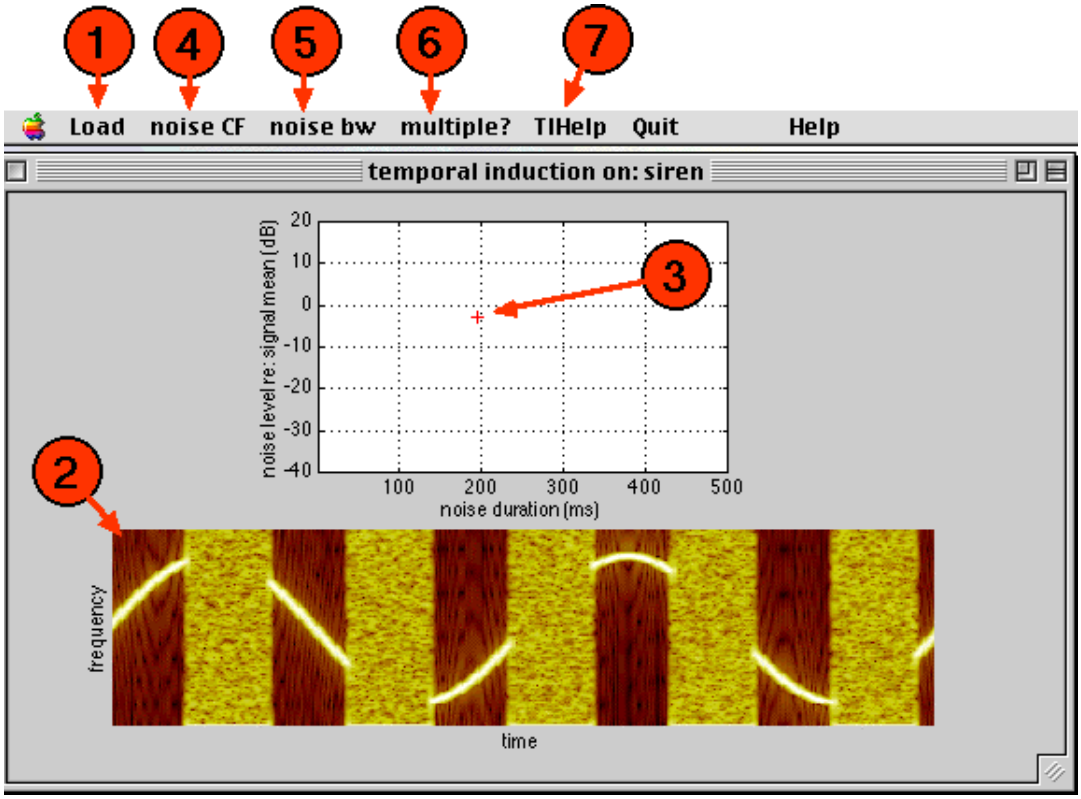


Temporal induction

Introduction

Under certain conditions, signals may be perceived as continuous even when temporal gaps are deliberated inserted and filled with noise (Miller & Licklider, 1950). This phenomenon applies to a wide range of signals from tones to speech, and has been termed *temporal induction*, *apparent continuity*, or the *phoneme restoration effect* when applied to speech signals (Warren, 1970). Temporal induction has since been shown to be one of a class of effects which includes contralateral (Warren & Bashford, 1976) and spectral induction (Warren et al, 1996). The current demonstration concerne temporal induction and allows the main stimulus properties of noise level and duration to be direectly manipulated for a variety of signal types.

The demonstration



Start the demo by typing 'ti' at the MATLAB prompt. Then load a sound file (1). A spectrogram of the sound appears in the lower panel (2). Click on the spectrogram to hear it.

To produce speech with noise-filled gaps, simply click in the upper grid (3). This produces a stimulus with noise bursts of level and duration determined by the location of the click. For instance, the cross indicates that a level of around -3 dB and a duration of just under 200 ms has been chosen. A spectrogram of the resulting stimulus is produced, and the sound is produced. Your task is to determine the range of (duration,level) pairs for which the underlying signal appears continuous.

By default, the noise burst are wideband, but the centre frequency and bandwidth of the noise can be changed using relevant menu options (4,5).

Noise bursts can be regular or irregular, with a noise/signal duration of 1:1, or singular. This option is chosen via the multiple? menu (6).

Things to investigate

1. Start with a tone signal and adjust the level for a constant duration of, say, 200 ms, until the tone appears to continue behind the noise. Now, keeping the level constant, change the duration. Repeat the test with irregular noise bursts.
2. Repeat the procedure above for the other signal types. Are the (level,duration) pairs similar for each signal type?
3. Use a non-wideband noise masker, and explore the necessary conditions for induction to occur.
4. What other factors are involved at extreme durations and at high noise levels?
5. What factors govern the intelligibility of interrupted speech (that is, speech with no noise in the gaps) as a function of gap duration?

References

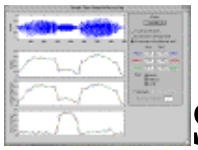
1. Miller & Licklider (1950). *JASA*, **22**, 167-173.
2. Warren (1970). *Science*, **167**, 392-393.
3. Warren & Bashford (1976). *Percep. & Psychophys.*, **20**, 380-386.
4. Warren et al (1997). *Percep. & Psychophys.*, **59**(2), 275-283.

Further reading

- A recent summary of these and related phenomena can be found in Warren (1996). *Proc. ESCA Workshop on The Auditory Basis of Speech Perception*, Keele.

Credits etc

Produced by: Martin Cooke



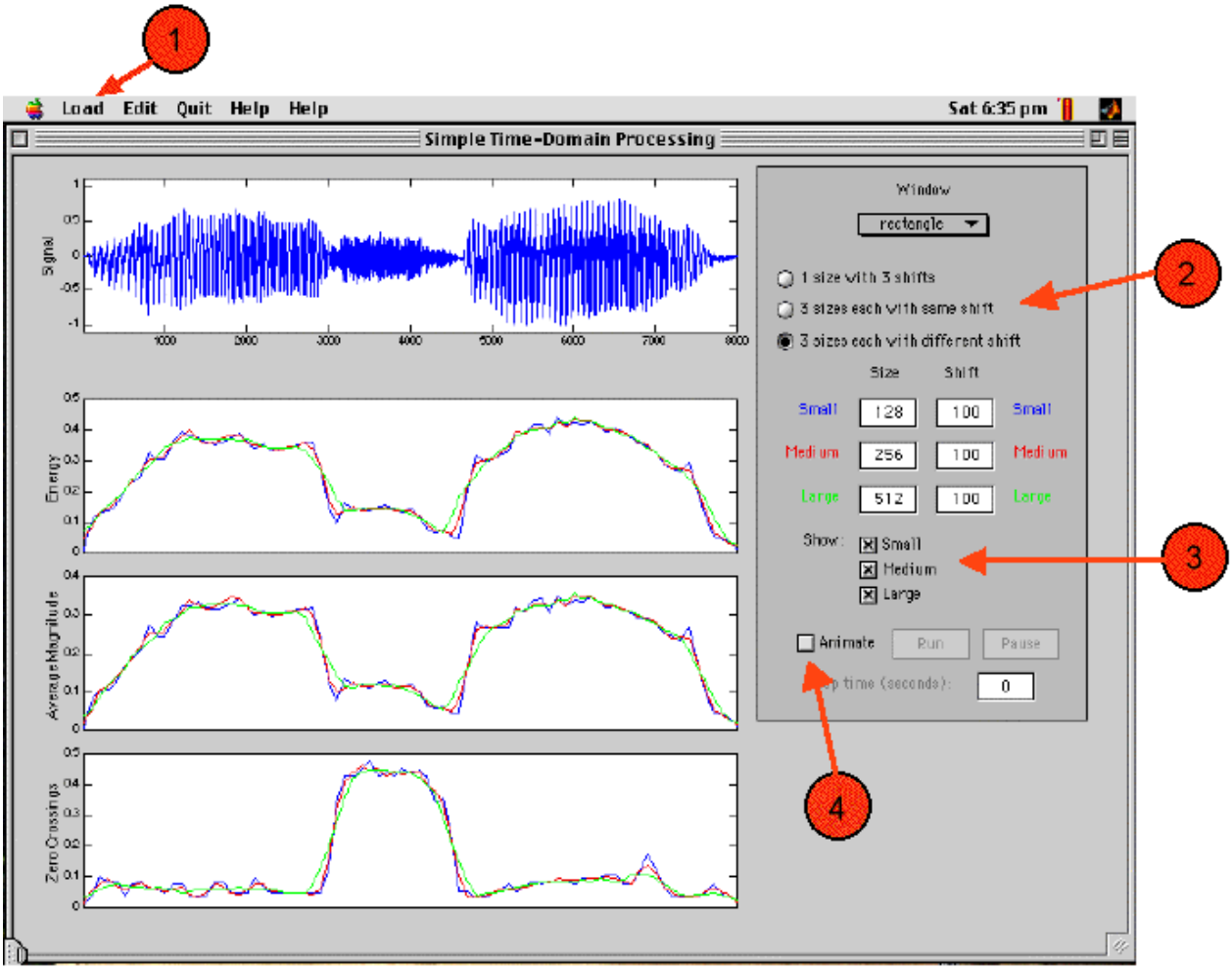
Simple Time-Domain Processing

Introduction

The simple manipulation of a signal's time-domain representation - it's **waveform** - can provide a number of useful properties. Some examples of time-domain measurements include energy, average zero-crossing rate and the autocorrelation function (see the auto MAD demonstration). This demonstration will concentrate on the average zero-crossing rate, energy and average magnitude which is closely related to the signal energy (we shall see later that the information they produce is equivalent).

These properties can either used purely as a new way of viewing the information implicitly buried within the signal or as a basis for more complex analysis (for example, in isolated word endpoint detection or pitch estimation). In signal analysis, the act of **windowing** can have a dramatic influence on the results. The three major aspects are window *type*, *size* and *shift*. This demonstration allows the student to both revise their understanding of the windowing process and also to see how various windowing properties influence the processing techniques.

The demonstration



Type ’timedom’ to launch the demo. When the window appears, use the load menu (1) to load a sound file. The signal can be played by clicking anywhere within the signal axes. As can be seen, the energy, average magnitude and average zero-crossing rate plots have been created. For each property axes there a 3 different plots - blue, red and green. These correspond to the 3 sets of window properties (2). There are a number of windowing options available:

- 1 window size, with 3 different shifts.
- 3 window sizes, each with the same shift.
- 3 window sizes, each with a different shift. (default)

The individual sizes and shifts together with the window type can be set by the student. Any of the 3 colours can be switched off at any point by using the ’show’ checkboxes (3).

The way in which the measurements are made in relation to the windowing of the signal can be investigated by using the animation feature. Click on the ’Animate’ checkbox (4). The measurement axes will be cleared and only the appropriate controls will be enabled. Again, the window type, size and shift can be set. To start the animation, click on the ’Run’ button. If the animation proceeds too quickly, a pause can be inserted between each segment

analysis. A pause of 0 seconds (default) turns off pause-insertion.
Click the 'Animate' checkbox again to revert back to the original interface.

Things to investigate

1. What do you notice about the energy and average magnitude plots? With reference to their respective defintions, how can your observations be explained?
2. What happens to the zero-crossing rate when there is a sharp increase in the energy? What happens when the energy sharply *decreases*? In the context of speech analysis, what is the reason for this and what types of sounds cause such changes?
3. How does the structure of the plots alter when the window sizes and shifts increase? Why?
4. Does altering the window type alter the plots? If so, why?

References

Rabiner, L.R. and Schafer, R.W., "Digital Processing of Speech Signals". Prentice-Hall, 1978, pp. 116-130.

Further reading

See also the demonstration for autocorrelation (**auto**).

Credits

Produced by: Stuart N Wrigley

Release date: January 20 1999

Vowel Explorer

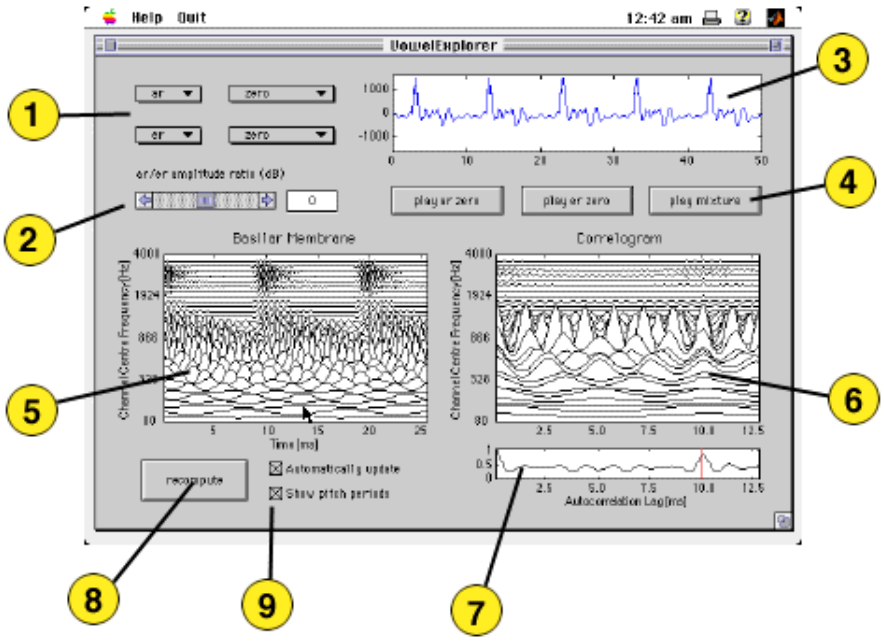
Introduction

Pairs of synthetic vowels that start and stop at the same time (so-called 'double vowels') have been used to demonstrate that listeners can use information about the fundamental frequency (F0) of sounds to separate them. Specifically, listeners are better able to identity the constituents of a double vowel when the two vowels have different F0s, relative to the condition in which they have the same F0.

This application allows you to experiment with the set of five vowels used in the studies of Assmann and Summerfield (1990). Many modelling studies have also used this vowel set (e.g. Meddis and Hewitt, 1992). Each of the five vowels is synthesized on six F0s, corresponding to differences of between 0 and 4 semitones from a 100 Hz fundamental:

Semitone difference	Fundamental Frequency (Hz)	Period (ms)
zero	100.00	10.0
quarter	101.45	9.86
half	102.93	9.72
one	105.95	9.44
two	112.25	8.90
four	126.00	7.94

The demonstration



Use the menus on the upper left of the screen (**1**) to select the constituents of a vowel pair. The amplitude ratio of the two vowels can be adjusted using the slider (**2**). The waveform of the vowel mixture is shown on the upper right of the screen (**3**), together with buttons which allow you to hear the mixture, and each individual vowel component (**4**).

The lower panels of the screen show the rectified basilar membrane response to the vowel mixture (**5**) and the correlogram (**6**). Below the correlogram is the summary (or pooled) autocorrelation function (**7**), which will tend to exhibit peaks of activity at the periods of the two vowels in the mixture. If the checkbox **Show pitch periods** (**9**) is activated, red lines in the summary function will mark the periods of the two vowels.

On slower machines, the update of the basilar membrane and correlogram displays may be sluggish. You can turn off automatic updating of these displays by deactivating the checkbox labelled **Automatically update** (**9**). When automatic updating is disabled, you can force a recomputation of the displays by pressing the **Recompute** button (**8**).

Things to investigate

1. Do you always see a peak in the summary autocorrelation function at the period of each vowel? Do you always see a peak at the period of at least one of the vowels?

2. What does your answer to (1) suggest about possible strategies for segregating double vowels?

References

P. F. Assmann and Q. Summerfield (1990) Modelling the perception of concurrent vowels:

vowels with different fundamental frequencies. JASA 88 (2), 680-697.

R. Meddis and M. J. Hewitt (1992) Modelling the identification of concurrent vowels with different fundamental frequencies. JASA, 91(1), 233-245.

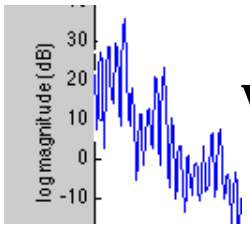
Further reading

See also the demonstrations for vowel segregation using neural oscillations (**VowelSeg**).

Credits

Produced by: Guy J. Brown

Release date: June 22 1998

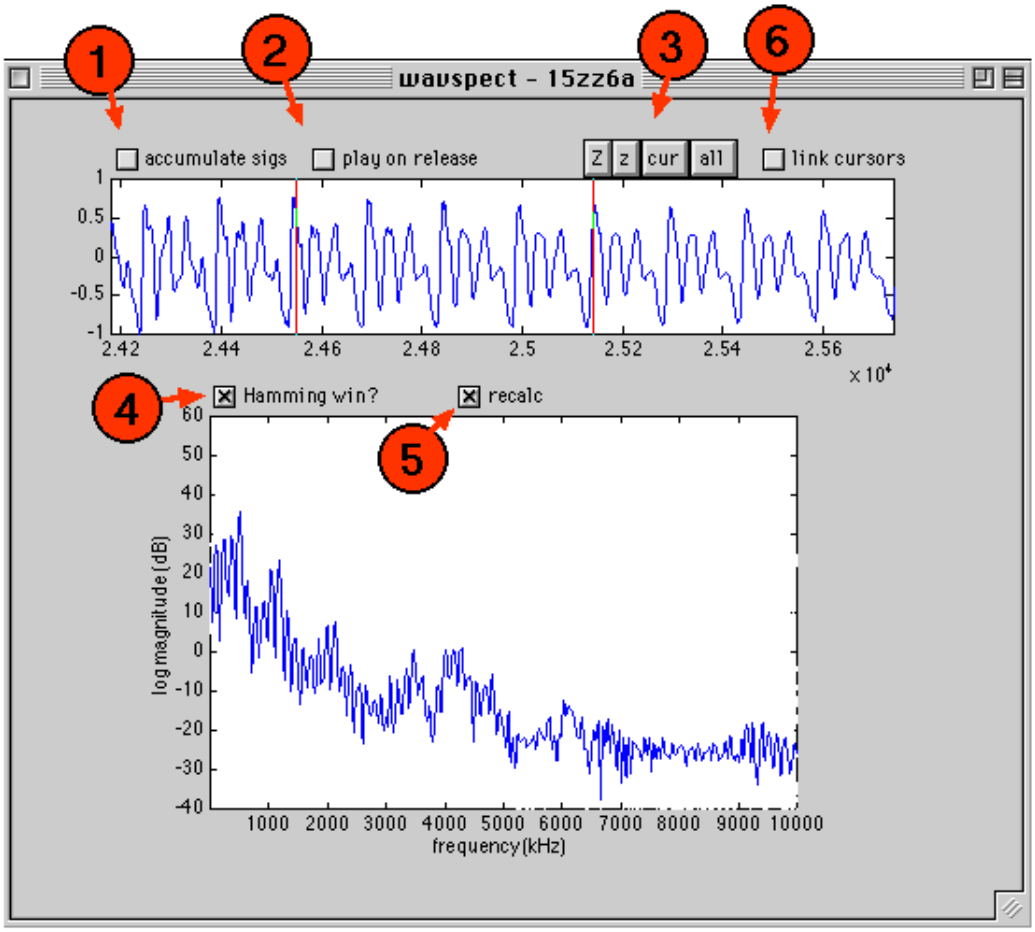


wavspect: waveforms and spectra

Introduction

This tool allows investigation of the relationship between the time and frequency domains. Users load in signals, and then see spectra of selected portions updated in near to real time. In the speech field, this can be used to illustrate the effect of window size, type and placement as well as short and long-term spectra.

The tool



Type ’wavspect’ to launch the demo. When the window appears, use the load menu to load a signal file. Supported formats currently include .wav, .snd and .au sound files. The signal and its spectrum will appear. The spectrum is always computed from the portion of the signal between the two cursors.

The control buttons (3) allow the user to zoom in and out. Additionally, the cursors can be linked (that is, to move together) by checking the control (6). In this mode, the signal displayed will be shifted right or left when one or other cursor reaches the current display boundary.

Cursors are moved by dragging. The portion between the waveform will be played back if the checkbox is ticked (2).

If the ’recalc’ checkbox (5) is ticked, the spectral display will be updated on-the-fly (as fast as possible).

By default, a rectangular window is used to select the signal between the cursors. A Hamming window can be applied instead (4).

When a new signal is loaded, the option exists (1) to add it in to the existing signal. This allows the spectra of signal mixtures to be investigated.

Things to investigate

1. Load a sinusoid and investigate its spectrum as a function of the window size, type and placement.
2. Load a speech signal. Examine its long-term spectrum. Now zoom in on a few ’pitch periods’ and compare the spectra which result from careful placement of the cursors on the precise start and end of the fundamental period. How many pitch periods are necessary to see evidence of the fundamental frequency in the spectrum?
3. How easy is it to find formants in speech spectra? What is the optimal window size/placement to reveal formants?

Further reading

- Any text on digital signal processing, linear systems or Fourier transforms will cover the relationship between the time and frequency domains.

Credits etc

Produced by: Martin Cooke

Release date: October 5th 1998