CSpeech Version 3.X
Waisman Distribution

Paul H. Milenkovic
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
1415 Johnson Drive
Madison, Wisconsin 53706

Charles Read
Department of Linguistics
University of Wisconsin-Madison 53706

Abstract

CSpeech, pronounced "See-Speech," is the name for the computer speech waveform acquisition, display, editing and analysis software package. The name is intended as a pun because CSpeech is useful in visualizing temporal and spectrographic properties of the acoustic speech signal. While the analysis functions of CSpeech are tailored to the requirements of speech, CSpeech is useful in other applications such as digital filter design and physiological data recording.

This version of CSpeech is distributed to lab computers of investigators of the Communication Processes Unit of the Waisman Center as well as the Speech Acoustics Lab of the Department of Electrical and Computer Engineering. This version contains enhancements to the commercial release of CSpeech Version 3.1 that represent experiments in the user interface.

This document covers

1. CSpeech commands
2. Inverse filtering
3. Jitter analysis
4. Filtering signals
5. Getting analysis results into text form
6. Hardcopy of CSpeech data

1 Introduction

CSpeech maintains a *waveform buffer* for recording, displaying, and performing operations on signal waveforms. The waveform buffer may contain one or more *channels* of signal waveforms sampled at equal rate. CSpeech maintains *work files* in the *current directory*, the disk drive and directory selected under DOS when CSpeech is started. The configuration of the waveform display is saved in the work file `cspeech.sav` while the waveform buffer is saved in the work file `cspeech.adc`.

The keyboard cursor keys (or optionally a mouse) may be used to place *waveform cursors* on the waveform display. The *selected interval* is the time span from the *initial cursor* to the *final cursor*. The *screen interval* is the time span from the left edge of the waveform display to the right edge of the display. Until the waveform cursors are selected and placed on the screen, the initial cursor position is by default at the left edge of the screen while the final cursor is at the right edge of the screen, and the selected interval is the same as the screen interval. To save keystrokes, some CSpeech commands operate on the selected interval while others operate on the screen interval.

The *time base* of the display may be controlled by zooming in on the selected interval (there are eight levels of zoom) or by zooming back out to the original screen interval. The channel *display scale* controls the apparent waveform amplitude on the screen. The vertical position of the oscillographic trace for each channel may be moved up or down on the display screen. Changes in time base, display scale, and vertical position affect only the appearance of waveforms on the display without changing the signal waveform sample values.

The *selected channel* may be chosen with the cursor keys. The number of the selected channel appears on the status line along with the display scale of the selected channel. Selecting and moving a waveform cursor obtains the waveform sample value in the selected channel at the cursor position and places the value on the status line, replacing the display scale readout. CSpeech commands that operate on a single channel operate on the selected channel. The selected channel is CH 1 until you use the cursor keys (`Home` and `End`) to select a different channel.

To use CSpeech, it is important to know that with DOS, you are always "in" a particular drive and directory: the current drive and directory. The current drive is indicated in the DOS command prompt; `C:>` indicates that drive C is the current drive. The current directory may be displayed with the command `cd`. Please review the DOS commands

`cd` Change directory

`mkdir` Make directory (or `md`)

`rmdir` Remove directory (or `rd`)

`dir` List Directory.

1.1 Starting CSpeech

The CSpeech program may be invoked with the command

```
cspeech
```

entered on the DOS command line. The DOS command line is indicated by a prompt, typically `C:>`, and commands are invoked by typing in the command and pressing the `Enter` key. This command invokes the commands found in the batch file `cspeech.bat` that has been prepared as part of the installation procedure.

When it is activated, CSpeech looks for the work files `cspeech.sav` and `cspeech.adc` in the current directory. If it finds the work files, CSpeech restores the number of channels, signal duration, sampling rate, and the screen display to the way they were the last time CSpeech was active in that directory. Thus, you can exit a CSpeech session and return to that same session easily. If it does not find these files, CSpeech asks for the number of channels, sample rate, and signal duration so that it may create a new set of work files. If these files are present, but you wish to reset these values, exit CSpeech with the `Quit` command from the main menu.

The allowed combinations of sampling rate and number of channels depend on the A/D D/A card used. Some cards (the ACPA, DACA, and DAS 8) permit recording only a single channel. The waveform buffer may contain multiple channels, but recording will take place into the first channel only while the other channels will be filled with zeroes during recording. For the ACPA card, digitizing will take place at a fixed 22 kHz rate, regardless of the sample rate specified.

1.2 Exiting CSpeech

The first way to exit CSpeech is with the `Quit` command, invoked by pressing q. The `Quit` command will erase the CSpeech work files. Reinvoking CSpeech from the DOS command line will cause CSpeech to ask for the sampling rate, number of channels, and buffer duration. The `Quit` command is useful for erasing the CSpeech work files and exiting CSpeech to start over.

The second way to exit CSpeech is with the `Exit` command found in the `Files` submenu. The `Files` submenu is first selected by pressing `f` and the `Exit` command is invoked by pressing `e`. This command saves the waveform buffer in the file `cspeech.adc` and saves the waveform display configuration in the file `cspeech.sav` prior to exiting. The `Exit` command is useful for exiting CSpeech, running other DOS programs, and then reentering CSpeech where you left off. Invoking CSpeech from the drive and directory in effect the last time CSpeech was exited will restore the waveform display.

2 CSpeech Commands

2.1 Cursor keypad functions in CSpeech

The layout of the cursor keys (including the mouse buttons) is given in Table 1 where `Ctrl`-*key* means hold down the `Ctrl` key and press *key* once. The functions CSpeech assigns to these keys are shown in Table 2. The use of the cursor keys may be understood by considering them in the following groups.

Channel selection

`Home` Select lower numbered channel (as in change from 2 to 1)

`End` Select higher numbered channel (as in change from 1 to 2)

↑ Scale up display of selected channel

↓ Scale down display of selected channel.

Waveform cursor movement

`Ctrl` ← Select initial waveform cursor for movement

`Ctrl` → Select final waveform cursor for movement

← Move the selected waveform cursor to the left

→ Move the selected waveform cursor to the right

`LeftMouse` Select and drag the initial cursor

`RightMouse` Select and drag the final cursor.

Holding down `Ctrl` and pressing ← once (`Ctrl` ←) selects the initial waveform cursor. Pressing → moves this cursor to the right starting from the left edge of the screen. Each press of → advances the initial cursor an additional increment. Holding down `Ctrl` and pressing → once (`Ctrl` →) selects the final waveform cursor. Pressing → moves this cursor to the right starting from the position of the initial waveform cursor. Each press of → now advances the final cursor an additional increment.

Screen interval

`PgDn` Zoom in to the interval marked by waveform cursors

`PgUp` Zoom out to a wider screen interval

`Ctrl-PgUp` Scroll right by one screen interval

`Ctrl-Home` Scroll left by one screen interval

`Ctrl-PgDn` Scroll right by one quarter screen interval

`Ctrl-End` Scroll left by one quarter screen interval

`Ins` Restore waveform cursors to initial positions

`Del` Remove message box from the bottom of the screen.

Home Ctrl-Home	↑	PgUp Ctrl-PgUp
← Ctrl ←		→ Ctrl →
End Ctrl-End	↓	PgDn Ctrl-PgDn
Ins		Del

LeftMouse	RightMouse

Table 1: Keys of the keyboard cursor pad and mouse buttons.

LowerChan ScrollLeftFull	ScaleUp	ZoomOut ScrollRightFull
MoveCurLeft SelectInitCur		MoveCurRight SelectFinalCur
UpperChan ScrollLeftPart	ScaleDn	ZoomIn ScrollRightPart
RestoreCurs		DelText

DragInitCur	DragFinalCur

Table 2: CSpeech function assignments to keys of the keyboard cursor pad and mouse buttons.

2.2 Menu commands

The CSpeech menu lists names of commands and names of submenus. To invoke a command, press the first letter of the command name. It does not matter whether you press a lower case or upper case letter on the computer keyboard. To view the contents of a submenu, press the first letter of the name of the submenu. To invoke a command within a submenu, press the first letter of each submenu name leading to the command and then press the first letter of the command. Pressing `Esc` or the spacebar at any submenu returns control to the main menu.

`Screen`

- `Bias` Change the vertical position of the selected channel
- `Visible` Make a hidden channel visible, make a visible channel hidden
- `All` Make all channels visible
- `None` Hide all channels
- `Mark` Mark the selected interval with a text label
- `Find` Position the waveform cursors by finding the text label among the marked intervals
- `Label` Show the text labels for the marked intervals
- `Hardcopy` Output the screen interval of the displayed waveforms to the file `screen.dat` in a form readable by the Grapher program.

`Files`

- `New` Create a zero-valued waveform buffer of the specified time duration
- `Get` Create a waveform buffer containing waveforms from one or more files created with the `Put` command
- `Put` Output one or more channels into waveform files
- `Leader` Specify the offset from the initial (left) cursor for waveform input or output
- `DOS` Enter a window to run DOS commands
- `Exit` Exit CSpeech, saving the displayed waveforms for later redisplay
- `Filter` Invoke a DOS command, redirecting standard input and output to waveform buffer channels
- `Turbo` Invoke the Turbo Pascal compiler (available on machines where Turbo Pascal has been installed)

`Menu` Invoke the Quick Edit shareware editor to view or update the menu script file (available on machines where `qedit` has been installed).

`Edit`

`Get` Input a single waveform file into the selected channel at the initial (left) cursor position

`Put` Output the selected interval of the selected channel into a waveform file

`Read` Input an ASCII (text) file into the selected channel at the initial (left) cursor

`Units` Change the units name (such as Volts) or the units scale (such as 20 PP) of the selected channel

`Copy` Copy the screen interval of the selected channel into one of the other channels of the waveform buffer

`Zero` Zero the selected interval of the selected channel

`Window` Apply a raised cosine taper to the initial and final 200 ms of the selected interval of the selected channel.

`Analysis`

`Filter`

`Hi pass` Apply a zero phase 20 Hz high pass filter to the screen interval of the selected channel

`Integrate` Integrate the screen interval of the selected channel

`Diff` Differentiate the screen interval of the selected channel

`LPC` Apply an LPC inverse filter to the screen interval of the selected channel

`APF` Apply an LPC all-pole (forward) filter to the screen interval of the selected channel

`FIR` Apply a Hamming window design lowpass FIR filter to the selected interval of the selected channel. The cutoff frequency is designated in kHz, and the maximum number of coefficients is 255.

`Spectrum`

`Fourier` Compute a Hamming window Fourier spectrum of the selected interval of the selected channel

`LPC` Compute LPC coefficients and spectrum of the selected interval of the selected channel

`Zero spec` Zero the spectrum display buffer

- `Decibels` Compute the RMS value of the selected interval of the selected channel in dB
- `Hardcopy` Output computed spectra to the file `spec.dat` in a form readable by the Grapher program.

`Display` Display computed spectra

`Ncoef` Change the number of coefficients for LPC analysis and the preemphasis coefficient for LPC and Fourier spectra

`Time/freq` Compute and display a time/frequency spectrogram – bandwidth, frequency range, minimum displayed intensity, and screen interval may be changed

- `Screen`
  - `Color` Toggle between a gray scale rendering and a false color rendering of the time/frequency spectrogram
  - `Mark` Mark the selected interval with a text label
  - `Label` Show the text labels for the marked intervals.
- `Analysis`
  - `Spectrum`
    - `Fourier` Compute a Hamming window Fourier spectrum of the selected interval of the selected channel
    - `LPC` Compute LPC coefficients and spectrum of the selected interval of the selected channel
    - `Zero spec` Zero the spectrum display buffer
    - `Decibels` Compute the RMS value of the selected interval of the selected channel in dB
    - `Hardcopy` Output computed spectra to the file `spec.dat` in a form readable by the Grapher program.
  - `Display` Display computed spectra
  - `Ncoef` Change the number of coefficients for LPC analysis and the preemphasis coefficient for LPC and Fourier spectra
  - `Formants` Compute and display formant tracks on the time/frequency display
  - `Time/freq` Recompute and display the time/frequency spectrogram
  - `Voice`
    - `RMS` Compute the RMS amplitude envelope of the selected interval of the selected channel

`Pitch` Compute the pitch contour of the selected interval of the selected channel using the FPRD algorithm

`Main menu` Exit time/frequency display and restore main menu

`hi/lo Cursor` Change between high and low increments of cursor movement.

`Voice`

`RMS` Compute the RMS amplitude envelope of the selected interval of the selected channel

`Pitch` Compute the pitch contour of the selected interval of the selected channel using the FPRD algorithm

`Jitter` Compute jitter, shimmer, and aperiodicity SNR for the selected interval of the selected channel.

`Waterfall`

`Fourier` Compute and display Fourier spectra (uses Hamming window) of the selected channel starting at the initial cursor and ending at the final cursor

`LPC` Compute and display LPC spectra of the selected channel starting at the initial cursor and ending at the final cursor

`1.BarkF` Compute and display Fourier spectra on a Bark transform frequency scale

`2.BarkL` Compute and display LPC spectra on a Bark transform frequency scale.

`Record` Digitize analog waveforms with the A/D converter card and store the samples in the waveform buffer

`Play` Play back the selected interval of the selected channel through the D/A converter card

`Grapher` Invoke the Grapher presentation graphics program (available on machines where Grapher has been installed)

`Quit` Exit CSpeech without saving the waveform buffer and waveform display configuration

`hi/lo Cursor` Change between high and low increments of cursor movement.

Find prompt

The commands `Files Get`, `Files Put`, `Edit Get`, `Edit Put`, and `Edit Read` employ the Find prompt to locate and select files. The Find prompt may be used to 1) change to a new drive or directory, 2) search the contents of a directory, 3) specify a file name, or 4) take no action.

The Find prompt indicates the drive and directory you are "in." To change to a new drive or directory, simply type a valid DOS path name and press the `Enter` key. The Find prompt will reflect the change. Home base for the drive and directory selection is the CSpeech current directory: the drive and directory selection in effect when CSpeech was invoked as well as the drive and directory containing the CSpeech work files. You may always return to the current directory by changing to directory "`.`" (type a period and press `Enter`).

To search the directory you are "in," type a DOS wildcard specification and press `Enter`. Information on DOS wildcards may be found in the DOS manual. An example wildcard is `*.wav`, meaning search for any file name with the extension `.wav`. The wildcard `*.*` means search all files. In the Find prompt, the effect of `*.*` may be obtained by pressing ↓. The file and directory names turned up in the search will be displayed in the Find prompt window. You may use the cursor arrow keys to scroll through the window and select one of the names. To accept a selection, press the `Enter` key. Selecting a directory name in this manner will cause CSpeech to change to that directory. Selecting a file name in this manner will select that file name for `Get`, `Put`, or `Read`.

You may specify a file by typing the file name and pressing `Enter` at the Find prompt. You may also specify a file name by accepting a file name selection as part of a directory search. Either way, `Get`, `Put`, or `Read` will use that file. You may exit the Find prompt without specifying a file by pressing the `Esc` key.

3 Common Types of Analysis

3.1 Performing spectrum and inverse filter analysis

CSpeech is capable of computing the narrowband Fourier spectrum (Hamming window) and the LPC spectrum a single frame at a time. Computing the LPC spectrum causes CSpeech to store coefficients for use in the inverse filter. The inverse filter has the effect of flattening the spectrum of the signal being inverse filtered. Applying the inverse filter to a speech waveform in effect removes the resonances of the vocal tract, allowing us to estimate the glottal waveform.

CSpeech may display an LPC spectrum and perform inverse filtering according to the following proceduer. First, make a copy of the screen interval of the speech waveform in CH 1 in CH 2 by invoking `Edit Copy`. Select CH 2 by pressing `End` once to change from CH 1 to CH 2 as the selected channel. Next, high pass filter the copy of the original acoustic waveform with the command `Analysis Filter Hi pass`. The high pass filter removes any DC offset or room rumble noise from the acoustic signal.

Position the cursors to select a 20 ms interval within the vowel portion of the acoustic signal. LPC analysis is performed by the command `Analysis Spec LPC`. The default preemphasis `alpha = 1` and the default number of LPC coefficients (22 in the case of 20 kHz sampling) are used. The preemphasis setting and number of LPC coefficients may be modified with `Analysis Ncoef`. The LPC spectrum may be displayed with the command `Analysis Display`.

Having computed a set of LPC coefficients with the `Analysis Spec LPC` command, apply the LPC inverse filter to the screen interval of the selected channel using `Analysis Filter LPC`. This produces the first derivative of the glottal flow signal. Integrate this signal with `Analysis Filter Integrate` to obtain the glottal airflow signal itself.

The above procedure is suitable for applying inverse filter analysis to the acoustic signal recorded with a microphone. Accurate inverse filter analysis requires a microphone and preamplifier combination with a high pass cutoff frequency at or below 2 Hz. Most commercially available microphones are not suitable for inverse filtering. It is still possible, however, to obtain meaningful LPC and Fourier spectra with these microphones.

The inverse filter may be applied to an airflow mask signal by modifying the above procedure. On a working copy of the airflow signal, position the cursors on a 20 ms interval in the middle of the waveform to be analyzed and perform

```
Analysis Filter Differentiate
Analysis Spec LPC
Analysis Filter Integrate
Analysis Filter LPC
```

Differentiating the flow signal makes it look spectrographically like a microphone pressure signal in order to do the LPC. Integrating after having performed LPC analysis recovers the original flow signal. The inverse filter is applied to the flow signal, and we dispense with the high pass filter and the integration step needed with the microphone signal because the flow signal is already an integrated signal. Inverse filtering the flow signal will provide an absolute DC reference on flow not found in the microphone signal.

3.2 Voice perturbation analysis

The `Analysis Voice Jitter` command of the CSpeech menu invokes the voice perturbation analysis procedure described by P. Milenkovic (1987): Least mean square measures of voice perturbation, *Journal of Speech and Hearing Research*, Volume 30, pp. 529-538. This procedure analyzes a speech waveform for fundamental frequency, jitter, shimmer, and aperiodicity signal-to-noise ratio (SNR). Jitter is the measure of fluctuation in the pitch period while shimmer is the measure of fluctuation of waveform amplitude between pitch periods. The aperiodicity SNR is the ratio, expressed in decibels, of the total energy in the speech signal to the energy in the aperiodic component of the speech signal according to a periodicity model that allows for amplitude and pitch period changes.

To perform voice perturbation analysis, use the `Home` or `End` key channel selectors to select the channel to be analyzed. Use the cursors to mark that portion of the waveform to be analyzed. Upon invoking the analysis, it is necessary to enter values for

```
List file name
Summary file name
Number of tokens
Estimated pitch
Update interval
```

Pressing `Ctrl-C` while entering any of these values will return control to the CSpeech menu.

The list file name designates a file to receive analysis results. Type in the file name and press `Enter`. To view analysis results on the screen without saving the numbers in a file, simply press `Enter` without giving a file name. The summary file name designates a file to receive the statistical ranges on the analysis results. Simply press `Enter` if it is sufficient to view the summary on the screen without saving the numbers to a file.

In response to the query for number of tokens, type a positive integer and press `Enter`. Simply press `Enter` to analyze without dividing the waveform into tokens. The number of tokens is the number of 100 ms subintervals into which the data being analyzed is divided. Non- tokenized analysis treats the entire data interval selected by the waveform cursors as a single token.

It is necessary to specify an estimate of pitch. Type in the value and press `Enter`. The voice perturbation analysis uses a pitch extraction method based on the short-term autocorrelation function of the speech waveform. This method gives fine resolution of the pitch period but it requires an accurate (within 10 percent) estimate of pitch to track voice perturbation correctly. Use the `Analysis Voice Pitch` command or measure pitch manually by marking one cycle of the speech waveform with cursors to determine an initial pitch value. To analyze a waveform interval with a large pitch excursion, furnish the value of pitch at the left edge of the interval. The voice perturbation analysis will track the pitch change if given an accurate starting point.

The update interval is the number of samples in each step of the sliding analysis interval used to track pitch changes. To accept the default value, simply press `Enter`. A larger update interval will make the analysis run somewhat faster at some loss of accuracy. The step size should be a small fraction (less than 10 percent) of the number of samples in a pitch period. For example, speech sampled at 10 kHz (.1 ms time step between samples) that has a fundamental period of 100 Hz (pitch period of 10 ms) has 100 waveform samples in each pitch period, so the step size should be no more than 10.

The issue of tokenized and non-tokenized analysis merits further explanation. Tokenized analysis is typically employed for sustained vowels. The number of tokens is the number of 100 ms subintervals upon which to perform the analysis. For example, 10 tokens applied to a 1000 ms interval of a sustained vowel will result in 10 tokens consisting of non-overlapping 100 ms subintervals. The tokens are used to obtain a statistical measure of the variation in the voice perturbation measures across a sustained vowel.

Non-tokenized analysis is typically employed for vowel segments extracted from words and phrases. Vowel segments are typically too short to divide into 100 ms tokens, so they are treated as a single token. Of course, the statistical summary will reflect zero variability when performing non-tokenized analysis.

The analysis interval will produce a `#` character for every 10 ms of the speech waveform that is processed. This indicates that the program is operating. For each token, the pitch period in ms, fundamental frequency in Hz, jitter in ms, shimmer in percent, and SNR in dB will be displayed. We express jitter in ms: divide the jitter by the pitch period to obtain the jitter in percent.

For each token, the `track` and `err` counts are listed. The `track` count is a measure of gross change in the pitch period, expressed in number of samples. A high track count indicates a large pitch period swing. The `err` count indicates the number of times the analysis algorithm failed to compute a pitch period consistent with the peak of the autocorrelation function. A large `err` count (greater than 10) indicates a loss of pitch track. This condition may arise from an inaccurate pitch estimate or a highly aperiodic voice waveform. It is important to check the pitch estimate against a manual measurement made with the CSpeech waveform cursors.

Upon completion, the voice perturbation analysis will provide a statistical summary of the variation in measures among the tokens. Non- tokenized analysis will result in zero variation. The pitch track is output in a file called `pitch`, the jitter waveform showing the plus and minus fluctuation in pitch period is output in a file called `jitter`, and the shimmer waveform showing the percent fluctuations in pitch period waveform magnitude is output in a file called `shimmer`. These files are found in the current directory, and they may be input into the waveform buffer at the left cursor position with `Edit Get` so they may be displayed.

4 Filters

Filters are programs that take a stream of data as input, transform the data, and output the transformed data. CSpeech employs such programs to implement digital filters, analysis functions, and conversions of data format.

Invoking the CSpeech command `Files Filter` activates the prompt `Command>` on the top line of the CSpeech menu. Type in the name of the filter along with any control parameters and press `Enter`. CSpeech will supply the cursor selected interval of the selected channel as waveform input. For filters producing waveform output, CSpeech will prompt for the channel to put the output. Press `Esc` to return to the main menu without invoking a command.

Filters may be piped: the output of one filter fed into the input of another filter. Piping is denoted by entering a series a filter names at the `Command>` prompt separated by the | symbol. For example, to lowpass filter a speech signal prior to computing its pitch, enter

```
cfir | cpitch
```

at the `Command>` prompt of the `Files Filter` command.

Filters may also be installed on the CSpeech main menu: examples include the commands `Analysis Voice RMS`, `Analysis Voice Pitch`, and `Analysis Spec Hardcopy` as well as the commands in the `Analysis Waterfall` submenu. Invoke the command `Files Menu` to view the menu entry file to see how this is done.

Filters may be also applied to batches of waveform files. This may be done by exiting CSpeech and by using the CBatch program. Enter the command

`cbatch` *fname filter args*

at the DOS prompt. The item *fname* names a waveform file, *filter* names the filter command, and *args* are optional control parameters of the filter. The file name *fname* can be a wildcard (file name employing * or ?. CBatch will apply *filter* to each valid file. Filter output will go into file with the same name but a different extension. The filter CPitch uses the extension `.f0` while CRMS uses the extension `.rms`.

Piping is not yet supported in batch mode. In addition, jitter analysis has not yet been implemented as a filter, meaning that it is not possible to perform jitter analysis in batch mode.

To write your own filters, refer to the source code files `copyw.pas`, `sine.pas`, and `square.pas`. The filter can communicate with CSpeech by writing and reading character strings to standard output and from standard input. Once the filter program has indicated whether it wants input or output and has obtained information such as data sampling rate, the filter program writes an empty string to standard output to signal that it is ready to process the binary waveform data.

Here are some of the filters available in CSpeech.

4.1 CFIR

Invoke the filter `cfir` from the `Files Filter` command prompt to filter the selected interval of the selected channel with a fixed point FIR filter. The default filter is a Hamming window low pass filter with the designated 6 dB cutoff frequency and filter order. The maximum allowed filter order is 255.

It is possible to supply your own filter coefficients by entering `cfir` *fname* at the `Files Filter` command prompt where *fname* is the name of a text file. The coefficients are decimal numbers in the range $-1 \leq d < 1$. One coefficient goes on each line of the text file, and the number of coefficients is determined by the number of lines in the file. Be sure you do not have blank lines in the file.

4.2 CIIR

Invoke the filter `ciir` from the `Files Filter` command prompt to filter the selected interval of the selected channel with a fixed point IIR filter. The default filter is a digital Butterworth low pass filter with the designated 3 dB cutoff frequency and number of second order sections. The filter will have two poles per second order section.

It is possible to supply your own filter coefficients by entering `ciir` *fname* at the `Files Filter` command prompt where *fname* is the name of a text file. The coefficients are decimal numbers in the range $-2 \leq d < 2$. Each line in the file contains the coefficients for one second order section in the order b_0, b_1, b_2, a_1, a_2. The number of second order sections will be the number of lines in the file; be sure the file does not contain blank lines, especially at the end. A practical bound on the number of filter sections is 32. The transfer function of each second order section has the form

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Because the filter employs fixed point arithmetic, it is necessary to distribute the overall filter scale factor among the b coefficients for each stage to prevent numeric overflow.

4.3 CPlot

CPlot is a filter that outputs the selected interval of the selected waveform channel in ASCII (text) form to DOS standard output. The first column of the output is time in milliseconds (ms) and the second column gives waveform values expressed in the units of the waveform display (see the `Edit Units` command).

Invoke the `Files Filter` command and enter

```
cplot >fname
```

at the `Command>` prompt. The file name *fname* designates a text file to receive the output of CPlot. One may alternately enter

```
cplot .ext
```

where *.ext* is a period followed by a three letter extension name. In this case, if the file *fname.fxt* had been input into the selected channel with `Files Get`, the output of CPlot will be placed in the file *fname.ext*. If no input had been made with `Files Get`, the output will be placed in the file `cspeech.`*ext* in the current directory.

The primary use of CPlot is to obtain an ASCII (text) version of CSpeech waveforms. The waveform samples are expressed in floating point values in the units shown on the display. The files created by CPlot may be read back into CSpeech with the `Edit Read` command of the CSpeech editor. The `Edit Read` command regards the time column of the file as column 0, and it regards the data column as column 1, its default column for input.

The optional parameter `/T:`*t* may be used to specify that the waveform be sampled every *t* ms instead of every sample position. This is expecially useful in extracting pitch and RMS contours in ASCII form. For example, ASCII output of the pitch contour sampled every 20 ms may be obtained by entering

```
cpitch | cplot .f0 /T:20
```

at the `Command>` prompt of the `Files Filter` command. This example illustrates the use of piping to perform both pitch analysis and ASCII data conversion in one operation. The pitch data will be in the file *fname*`.f0` where *fname* is the name of the waveform file input with the `Files Get` command. If the `Files Get` command was not used, the data will be found in `cspeech.f0`.

5 Hardcopy

Hardcopy is the process of reproducing data viewed on the computer screen on paper sheets output by a printer. One form of hardcopy is a literal reproduction of a screen image on a printed page. This type of hardcopy is performed by a *screen dump*; the Pizazz Plus program is a screen dump that works with all of the displays and printers available on our computers. Another form of hardcopy is the rendering of waveform and spectrum displays with axes and text labels added. This type of hardcopy is performed by a *presentation graphics program*, Grapher being an example. The last type of hardcopy is the printing of lists of numeric values. The jitter analysis in CSpeech optionally produces files containing analysis numbers. These files may be printed by the command

`copy` *fname* `prn`

invoked from the DOS prompt.

For descriptions of Pizazz and Grapher, please refer to their manuals. If these manuals are not available, please ask your lab director to purchase these programs if they are required for your work.

On lab computers connected to laser printers, we have streamlined the interface between CSpeech and Grapher. The CSpeech command `Screen Hardcopy` produces the file `screen.dat` containing a rendering of the displayed waveforms in a form usable by Grapher. The CSpeech command `Analysis Spec Hardcopy` produces the file `spec.dat` containing a rendering of the spectra produced by `Analysis Spec Fourier` and `Analysis Spec LPC`, spectra viewed by `Analysis Display`.

The files `screen.dat` and `spec.dat` stick around only as long as you are in CSpeech – this keeps the fixed disk from getting cluttered. Grapher may be invoked from the main CSpeech menu to produce output from these files. After you use the Grapher menues to get the plots with the correct axes and labels, use the Grapher menu to "Create the Plot File." This action produces the files `screen.plt` or `spec.plt`. These files will stay on the fixed disk: rename them to meaningful names and save them on floppy disk. These `.plt` files can always be rendered on the laser printer with the `ljplot` command at the DOS prompt. Next, if you use Grapher to "Print the Plot File", the temporary file `screen.hpl` is created. Exiting Grapher at this point invokes the output procedure for the laser printer. If you don't want to take the time for this output, just skip the "Print the Plot File" step, exit Grapher, and save your `.plt` files for later output with `ljplot` when you have time.