DESC9115 DIGITAL AUDIO SYSTEMS 2012
LAB REPORT 2 (LPC Cross Synthesis & Distortion)
Instructor: William L. Martens
Student: Tsun Kit Lam
SID: 420088455
Date: 8 May, 2011

LPC Cross Synthesis & Distortion

The LPC cross-synthesis between two sounds can also be achieved by using one filter to remove the spectral envelope (which is the resonance) of the input signal. The excitation signal of the input signal is filtered by the spectral envelope of the excitation sound.
The prediction coefficients of the input signal are used to whiten the sound by using the FIR filter. The prediction coefficients of excitation are used in the feedback path of a synthesis filter, which performs filtering of the inout signal with the spectral envelope derived from excitation sound.
In addition, the distortion simulates a clipping function to the signal during the process, resulting in harmonic added to the signal, giving it a fuller and harsher sound.

Syntax

```
xy = lpcdistort(inwave, p1,  gain, mix, exsignaltype, p2)
```

Description of Function

`lpcdistort.m` cross synthesizes the filter resonance of the input signal and excitation signal that is chosen. In addition, distortion can be applied on the resonance of the input signal.

Parameter

Input

```
inwave = Input Signal (provided by user)
p1     = numbers of Filter Coefficients of inwave
gain   = amount of distortion on the resonance extracted from inwave
mix    = mix of original and distorted sound, vale: 0 = no distortion
                                                    1 = only distortion
exsignaltype = Type of Excitation Signal
               (to be chosen in "switch" using different cases)
p2           = numbers of Filter Coefficients of chosen Excitation
```

Output

```
xy           = cross-synthesized signal with distortion applied
```

Excitation Signal (Using Different Cases)

Users only have to import the inwave (input signal), as for the excitation signal, 11 sets of excitation signal data are provided in the folder. Allowing users to experiment with a wide range of resonance characteristics. Users do not have to import the data sets manually, when the type (case) of excitation signal is chosen, the particular data will be imported automatically by the function.

exsignaltype

```
case 0 = high resonate organ
case 1 = hallow drone (film sound)
case 2 = hard driven synth bass
case 3 = UFO (Sci-Fi film sound)
case 4 = alien scream
case 5 = elephant scream
case 6 = fuzz synth
case 7 = human scream
case 8 = Tibetan chant singing
case 9 = cymbal
case 10 = punchy synth with high resonance
```

Tips For Using This Function

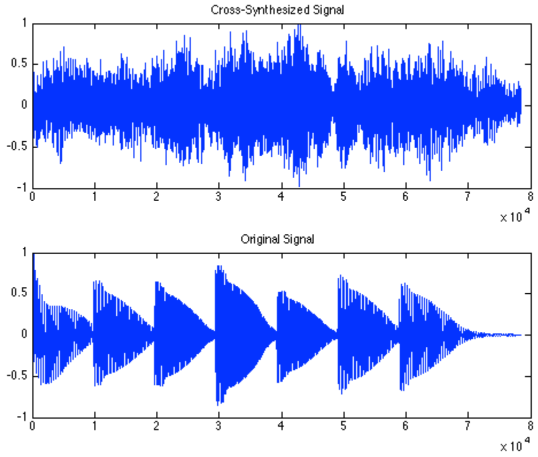
For achieving extreme (noticeable) effect, set the p1 and p2 up to at least 200. In addition, the higher the amount of distortion gain, the longer each note will sustain. For reverb alike effect, set p2 way higher than p1, e.g. p1=5 , p2=10000

Example

Transform between the input synth bass wave file and the fuzz synth excitation signal (case 6), into a synthesized sound that sounds like a distorted bell with false reverb (due to high resonance)

```
[inwave fs] = wavread(‘bass.wav’);
xy = lpcdistort(inwave, 10000,  300, 1, 6, 10000);
```

This function outputs the affected signal as a data vector, a wave file written to disk, a playback function in matlab, as well as a plot that compares the xy (output signal) with the inwave (input signal).



Reference

Matlab Code

David, M *MATLAB Digital Audio Effects Processor*
Cardiff University
Cardiff School of Computer Science & Informatics

See Also

`lpc, filter, freqz, wavread, sound, sign, exp, max, abs, switch`