vibrato_mod.m

Syntax

[Output ModSignal] = vibrato_mod(Input, FS , ModFreq, Width, Q)

Description

This function applies a frequency modulated vibrato effect to an audio signal by means of putting the signal through a quasi-periodic modulating delay system. The function has been designed with the objective of producing vibrato that is not absolutely periodic or equal in amplitude throughout in an attempt to replicate the vibrato produced by humans.

Input

Input = Input audio signal (vector column)
FS = Sampling rate of the input audio signal (in Hz)
ModFreq = Modulation frequency of the Vibrato Effect (in Hz)
Width = Maximum delay of the delay modulation signal (in seconds) (i.e. how much pitch variation there is in the vibrato)
Q = Quality Factor. Determines how strictly the vibrato effect adheres to the specified modulation frequency parameter (ModFreq)

Output

Output = Processed input signal
ModSignal = Vector of the modulating delay signal
[Chart] = A graphic representation of the modulating delay signal

The modulating delay signal is generated by filtering fractal noise at the specified modulation frequency (ModFreq parameter). The extent to which the modulation signal adheres to the ModFreq parameter is determined by the quality factor of the filter applied to the fractal noise, represented by the ‘Q’ parameter. The higher the ‘Q’ value the more the delay signal will resemble a periodic sine wave of the frequency assigned to the ModFreq parameter. The Width parameter controls the maximum delay time of the modulating delay signal which, in turn, affects how much pitch variation there is in the resulting vibrato.

The function automatically calls the ‘proclpc’ function to separate the audio signal into its source excitation and resonant components. The modulating delay signal is applied only to the source excitation, which is then resynthesised with its resonances, using the ‘synlpc’ function, to produce the resulting output signal.
Figure 1. Examples of unfiltered fractal noise, filtered noise with low Q and filtered noise with high Q. These are the shapes of the modulating delay signals to be applied to the input signal.
Output

The `vibrato_mod` function outputs the processed signal as a vector and sounds it automatically. It also outputs a vector of the generated modulating delay signal and a graphical representation of this delay signal, examples of which can be seen in Figure 1.

Example

```matlab
[Input FS] = wavread('InputAudioSignal')
ModFreq = 5
Width = 0.001
Q = 50
[Output ModSignal] = vibrato_mod(Input, FS, ModFreq, Width, Q)
```

Recommended Parameter Values

- **ModFreq**: 3 – 7 Hz
- **Width**: 0.0003 – 0.002 s
- **Q**: Any value above 20

This function is capable of operating on input signals that are less than ten seconds in length.

Acknowledgements

Many thanks must be extended to William Martens, not only for his suggestions and advice regarding this project, but also for supplying much of the Matlab code that is utilised in this function.

References

Matlab Code

- Vibrato Code


  ‘frand64’, ‘get_LP_filtered_mod_JL’, ‘M_fq2coef’, ‘normsig’, ‘proclpc’, ‘randsig’ and ‘synlpc’ functions were all sourced from William Martens.

Text
