Demo 10 Exercises: Vibrato

DSP Lab (ECE 4163 / ECE 6183)

2019

Demo files

`play_vibrato_simple.py`
`play_vibrato_interpolation.py`
`myfunctions.py`
`author.wav`

The demo program `play_vibrato_simple.py` is a simple implementation of the vibrato effect. This implementation is poor because the time-varying fractional delay is implemented by rounding the delay to an integer number of samples. For better audio quality, interpolation is usually used instead, as in the demo program `play_vibrato_interpolation.py`

Exercises

1. Modify the vibrato demo program `play_vibrato_simple.py` so it plays a stereo output signal. Use different vibrato parameters in the left and right channels.

2. Modify the vibrato demo program `play_vibrato_simple.py` so the audio input is from the microphone (not a wave file).

3. The vibrato demo program `play_vibrato_interpolation.py` uses linear interpolation. Write a version that uses quadratic (or cubic) interpolation.

4. Write a Python program to implement the flanger effect. Use interpolation for an improved result. As described in Chapter 2 of *Audio Effects: Theory, Implementation and Application*, the flanger effect is like the vibrato effect but it additionally has a direct path, as shown in the figure. The input signal should be read from a wave file. SUBMIT

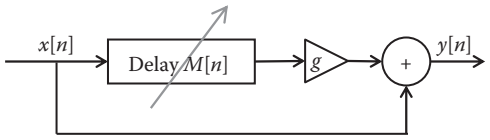

Figure 2.11
Block diagram of a basic flanger without feedback. The delay length $M[n]$ changes over time.

5. Write a Python program to implement the chorus effect. See Chapter 2 of *Audio Effects: Theory, Implementation and Application*.