MUS424:
Signal Processing Techniques for Digital Audio Effects `Handout #6`
Jonathan Abel, David Berners `April 11, 2019`

Homework #2: Compression Architectures [30 points]
Due Date: April 18, 2019

Submission Instructions

Submit via Canvas. Create **a single compressed file** (`.zip`, `.tar` or `.tar.gz`) containing all your submitted files. Upload the compressed file to the homework submission dropbox. Name the file using the following convention:

```
<suid>_hw<number>.zip
```

where `<suid>` is your Stanford username and `<number>` is the homework number. For example, for Homework #1 my own submission would be named `cavdir_hw1.zip`.

For coding problems (either C++ or Matlab code), submit all the files necessary to compile/run your code, including instructions on how to do it. In case of theory problems, submit the solutions in **PDF format only**. LaTeX or other equation editors are preferred, but scans are also accepted. In case of scanned handwriting, make sure the scan is legible. Illegible homework will not be graded.

Lab2: Compression Architectures

In this lab, you will explore gain computation and level detection by modifying a basic compressor configured as a limiter. You will modify the gain computer to provide control over the compression ratio; you will then modify the gain computer to turn your compressor into a downward expander. In a theory problem, you will compute the static compression characteristic for an analog-style gain computer. You will then modify the peak detector of the basic compressor to have it release to the threshold of compression, and further modify the detector so that its release adapts to the signal being compressed. Finally, you will be asked to design a compressor matching the behavior of a "mystery" compressor.

Note: If a parameter value is not mentioned in the instructions, pick any value you judge reasonable and **indicate in your write-up the chosen value**.

Problem 1. [3 points] Gain Computation.

The basic compressor implements a limiting static compression characteristic. Modify the basic compressor gain computer to produce the static compression characteristic shown on page 28 of the notes,

$$\ell_{\text{out}} = \begin{cases} \ell_{\text{in}} & \ell_{\text{in}} < \ell_{\text{T}}, \\ \ell_{\text{T}} + \frac{1}{\rho}(\ell_{\text{in}} - \ell_{\text{T}}) & \ell_{\text{in}} \geq \ell_{\text{T}}, \end{cases} \tag{1}$$

where ℓ_{in}, ℓ_{out}, and ℓ_{T} are the dB input level, output level and threshold, respectively, and where the compression ratio ρ is controlled by the Compression Ratio slider.

Set the threshold of compression to 0.1, and the attack time to its fastest value, and plot of the output of the compressor in response to a sinusoidal input with an amplitude which exponentially sweeps from 0.01 to 1.0 over about five seconds (a sample audio file is provided in the lab material).

Problem 2. [5 points] Analog-Style Gain Computer.

A number of analog compressors use a combination of a light source and photoresistor to implement a detector and gain computer. An input signal drives the light source, which, for signal levels above a threshold, generates light with an intensity increasing with the signal energy. The input signal (multiplied by a factor of 2) is applied to a voltage divider, with the photoresistor R_p and a fixed resistor R_0 splitting input signal according to the light intensity. For a detected signal level ν, the light-photoresistor element will produce a gain of

$$\gamma(\nu) = \frac{2R_p(\nu)}{R_0 + R_p(\nu)}, \qquad R_p(\nu) = R_0 \left(\nu/\lambda_T\right)^{-0.75}, \tag{2}$$

where λ_T is the compression threshold, and with R_0 about 100 kΩ. In this way, when the signal is strong, the light is intense, causing the photoresistor resistance to become small and the gain applied to the input signal to become small, thus compressing the signal. (By the way, the time constants associated with the photoresistor—and there are several of them, some decently long—determine the temporal dynamics of the compressor.)

Use Matlab to plot the static compression characteristic for the 40 dB range immediately above the threshold of compression when the light-photoresistor element is used in (i) a feed forward compressor and when it is used in (ii) a feedback compressor. In addition to your plots, turn in your Matlab code.

Problem 3. [12 points] Level Detector.

3(a). [2 points] Edit the peak-detection code in the stock plugin so that the detector uses release-to-threshold, rather than release-to-input. Turn in your code.

3(b). [4 points] With the compression ratio set to infinity, the threshold of compression set at $\ell_T = -20dB$, and the input and output gains set to unity, write an expression for the compressor gain $\gamma(t), t \geq 0$, of the basic compressor for the input signal $x(t) = 1 - u(t)$—that is, when the input has been one for sufficiently long that at time $t = 0$, the level tracker is also one, $\lambda(t) = 1$, and that just after $t = 0$, the input $x(t)$ becomes and stays zero. Repeat for the release-to-threshold detector. Turn in Matlab plots of these two release gain trajectories, and note which is more smooth.

3(c). [**6 Points**] Modify the basic compressor to have a program-dependent release. Make the release behavior such that the release time is approximately one second following a sustained high input level, and make the release time following transients controllable using the existing slider for release time. Turn in your code and plots showing the compressor's response to the signal `tdiff.wav` (contained in the `files.zip` file), with the fast release component set to 100ms, and the input gain and threshold set so that the small amplitude portions of the `tdiff.wav` signal are at the threshold.

Problem 4. [**10 Points**] **Mystery Compressor.**

For this problem, you will be asked to analyze the mystery plug-in `DRC`, which is contained in the file `files.zip` (the "mystery compressor" folder contains versions of the plugin for different OSs, so you should only copy the one you need). You may create any test signals you need to analyze the plug-in, and may use MATLAB to study the output of the plug-in. Answer the following questions and explain how you came to your conclusions. *Hint: Unlike many audio processing devices, the mystery plug-in DRC will support the entire spectral band from DC to the Nyquist limit.*

4(a). [**2 Points**] Determine the threshold of compression and the compression ratio.

4(b). [**2 Points**] Write pseudo-code for the static gain functions Φ_F and Φ_B that would be needed to implement this compressor using feedforward or feedback topologies.

4(c). [**2 Points**] Determine whether the plug-in uses RMS or peak detection.

4(d). [**2 Points**] Determine the approximate attack and release times for this plug-in.

4(e). [**2 Points**] Write pseudo-code for the signal estimator that would be necessary to implement this plugin for the feed-forward case and the feedback case.