Analog Compressor Modeling

Frank White, Timothy Dick

University of Rochester ECE272 Audio Digital Signal Processing

1. Abstract

A compressor is an automatic gain control effect that scales the output level of a signal based on the input level. If the input level is above a certain threshold, the compressor reduces the gain of the signal based on how high the input signal is above the threshold.

The 1176LN is a hardware compressor originally produced in the late sixties. The circuitry has been tweaked several times to decrease noise and improve efficiency. There are modern reproductions currently being produced by Universal Audio. The compressor is an entirely solid state piece of gear that is sought after for its clear tone and clean compression.

We aimed to characterize aspects of the 1176 and to build a functioning digital model in Simulink that behaves closely to the real world compressor.

2. Introduction

2.1. Basic Operation

Compressors take an input signal and change its gain based on its amplitude. A compressor has a threshold, which is the amplitude level above which the unit will reduce the input signal gain. To determine the amount of gain reduction, the compressor compares the input amplitude against a static table, which dictates the output level based on the input level. The gain factor is then smoothed based on the attack and release times set on the unit. The gain factor is then applied to the input signal which is here the actual gain reduction occurs.

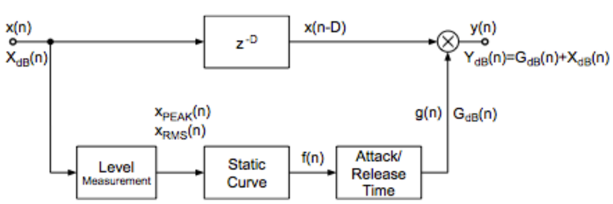


Figure 1. Basic Compressor Block Diagram

2.2. Level Measurement

The compressor must smooth the input signal with a level follower to be able to measure the signal input amplitude. The compressor can not simply measure the amplitude of the input signal because it would detect zero dB input every time the signal switched from positive voltage to negative voltage. This error can be avoided by using an envelope follower. An envelope follower can be implemented with a low pass filter in either the analog or digital domain. In either domain the output of the level measurement stage gives a smooth RMS amplitude value of the input signal.

2.3. Static Table

The static table is where the desired output amplitude based on input amplitude is stored. The table in its simplest form is a piecewise linear function as seen below.

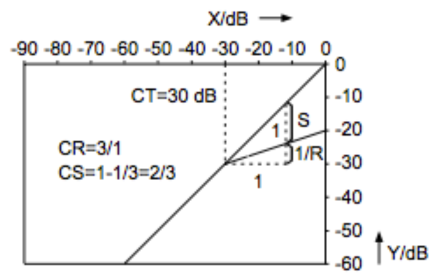


Figure 2: Static Table

The compressor threshold is the break point of the piecewise linear function, which in this case is CT=-30dB. The compressor ratio is the reciprocal of the slope of the line above the threshold. With the example ratio of 3:1 that means for every 3dB of increase above the threshold, the output sees 1dB of increase. On many compressors, the transition on the piecewise linear function is smoothed with a knee. The knee softens the break in the middle of the static table and prevents artifacts from the compressor kicking in and cutting out if the input signal is hovering right around the threshold.

2.4. Gain Factor Smoothing

To achieve variable attack and release times, the output of the static table must be smoothed. This prevents artifacts from instantaneous gain scaling. Like the input level measurement, this can be achieved by low pass filtering the gain factor. The current gain factor must be compared with previous gain factors to determine whether the compressor is in an attack or release phase. This acts as a small hysteresis curve given by:

$$g(n) = (1-k) * g(n-1) + k * f(n)$$

with $k=AT$ or $k=RT$, which correspond to the attack time and release time factors. These factors are found using:

$$k = 1 - e^{-2.2T/t}$$

where T is the sampling period and t is the attack or release time in seconds. This stage smoothly connects different scaling values over the prescribed attack or release times.

3. Testing

To test the 1176, a Stanford Research SR1 Audio analyzer was used to feed the unit signal and measure the output.

3.1. Threshold & Knee Width

Since the 1176 compressor does not have a variable threshold, its actual threshold had to be measured. To measure the lower limit of the knee, the compressor was fed a sine wave with a small amplitude that was then increased until the unit started compressing. The compressor was then fed a sine wave with amplitude well above the threshold that was then decreased until the unit stopped compressing.

Ratio	Lower Limit (dBm)	Upper Limit (dBm)
4:1	-17.5	-15
8:1	-12.8	-10.8
12:1	-9.6	-11
20:1	-8.4	-7.6

It is clear that the compressor threshold changes with the ratio. The lower ratios have lower threshold values than the higher ones. These set thresholds are utilized by changing the input gain of the compressor to change the input amplitude before it reaches the compression stage.

3.2. Attack and Release Times

The attack and release times for the 1176 are set based on the attack and release knobs on the unit. However, the actual attack and release times do not correspond to the markings on the unit. The actual 1176 attack and release times were found in the compressor manual.

3.3. Frequency Dependent Compression

The compressor was fed frequency swept sine waves at varying amplitudes to see how the output amplitude depended on frequency.

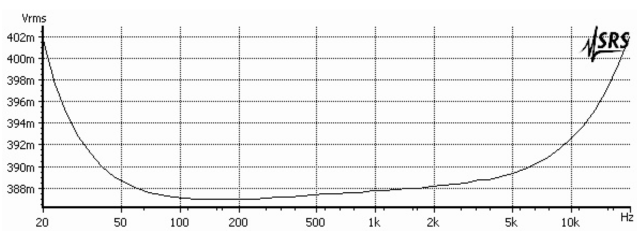


Figure 3: Compressed Frequency Response

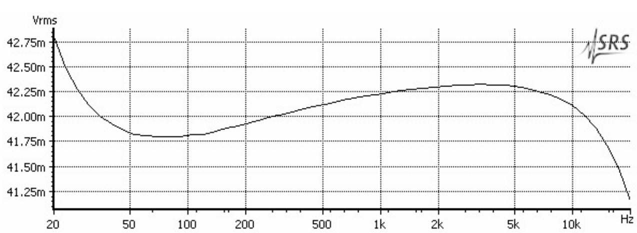


Figure 4: Uncompressed Frequency Response

These figures show that frequency response of the output changes depending on whether the unit is compressing the signal or not. Extreme high and low frequencies were boosted when the input amplitude was above the compressor threshold. The unit had a very flat response in the middle of the audio band.

4. Simulink Implementation

The model of this compressor was built in Simulink. The overall block diagram can be seen below:

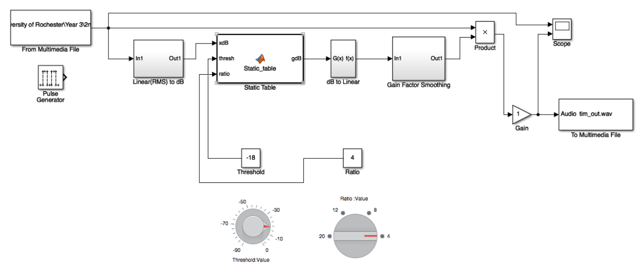


Figure 5: Compressor Block Diagram

4.1. Input Level Measurement

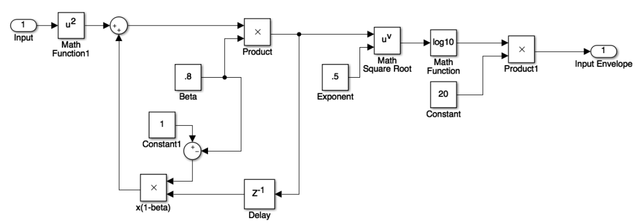


Figure 6: Input Level Measurement Block

An auto regressive formula was used to calculate the input RMS amplitude which is given by:

$$x_{RMS}{}^2[n] = (1-\beta)x_{RMS}{}^2[n-1] + \beta * x[n]$$

where β is a weighting parameter between zero and one. This first difference equation smooths the input signal since it is a digital low pass filter. The output of the auto-recursive formula is then converted to dB by:

$$envelope = 20 * \log_{10}(x_{RMS}[n])$$

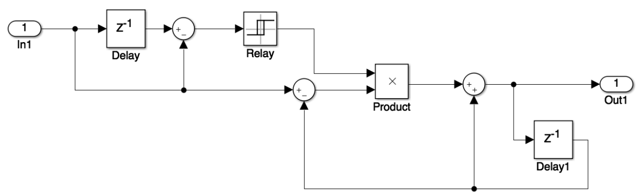
4.2. Static Table

To implement the static table, a custom Simulink block was created using Matlab code. The block takes in the threshold and ratio as well as the envelope and outputs the desired gain reduction factor based on the following lines of Matlab code:

```
if xdB > thresh %if input gain is above threshold
     gdB = (thresh + (1/ratio)*(xdB-thresh)) - xdB;
     %calculate desired output gain
else  %if input is below thresh, output
    gdB = 0;   %zero gain reduction
```

4.3. Gain Factor Smoothing

To implement the gain factor smoothing section of the compressor, similar to above, a custom Simulink block was created. It includes a feedforward first difference section, along with a relay that switches between the chosen attack time and release time coefficients that ultimately gets multiplied into another variation of a feedback first difference. This Simulink block diagram:



This block implements the difference equation given above in the Gain Factor smoothing section, which has a corresponding transfer function of:

$$H(z) = \frac{k}{1-(1-k)z^{-1}}$$

The setback of this implementation in Simulink, is that the AT and RT parameters are only accessible from within the relay block. The other tricky part was setting the switching point for the relay. The best results came from the switch on point being as close to 0 as possible, without actually being 0. The same with the switch off point, but its negative. Matlab includes a number called 'eps' which corresponds to floating point accuracy, or 2^{-52}. To help visualize and confirm that the variable attack and release times work, a square pulse was fed into the model to see what the resulting waveform would look like, and as expected, we had a smooth onset as well as release. The result can be seen here:

5. Conclusion

We found that it was very difficult to model many of the characteristics of the UA 1176LN compressor. The gear has many non-linearities that make modeling very challenging. Our model only approximates the most basic operation of the compressor and therefore is not a very accurate digital representation. We have successfully implemented a working dynamic range compressor in Simulink that has variable threshold, compressor ratio as well as variable attack and release times.

6. Future Work

There are many aspects of the 1176 that are still yet to be implemented in this model. First, we found that the 1176LN that we tested operated based on a feed back architecture rather than the feed forward architecture that we implemented in our model. The input and output gain stages will also be important to implement to match the hardware controls.

The most difficult aspect of the 1176 to model is the changing frequency response. There are several different ways to implement this characteristic. One option could be to implement a variant on multi-band compression that tailors the frequency response based on compression level. Another option could be to implement several filters to shape the tone of the output signal to better match the response of the hardware.

References

[1]U. Zolzer, *Digital audio signal processing*. Chichester, U.K.: Wiley, 2008.
[2]D. Giannoulis, M. MASSBERG and J. REISS, "Digital Dynamic Range Compressor Design— A Tutorial and Analysis", *Journal of the Audio Engineering Society*, vol. 60, no. 6, 2016.
[3]J. Reiss, "Under the Hood of a Dynamic Range Compressor", Queen Mary, University of London Center for Digital Music.
[4]U. Zö lzer and X. Amatriain, *DAFX*. Chichester, England: Wiley, 2002.
[5]B. Putnam Jr., *Model 1176LN Solid-State Limiting Amplifier Manual*, 1st ed. Evanston, IL: Universal Ausdio, 2009.