FFT-BASED DYNAMIC RANGE COMPRESSION

Leo McCormack and Vesa Välimäki
Acoustics Lab, Dept. of Signal Processing and Acoustics
Aalto University, FI-02150, Espoo, Finland
leo.mccormack, vesa.valimaki@aalto.fi

ABSTRACT

Many of the dynamic range compressor (DRC) designs that are deployed in the marketplace today are constrained to operate in the time-domain; therefore, they offer only temporally dependent control of the amplitude envelope of a signal. Designs that offer an element of frequency dependency, are often restricted to perform specific tasks intended by the developer. Therefore, in order to realise a more flexible DRC implementation, this paper proposes a generalised time-frequency domain design that accommodates both temporally-dependent and frequency-dependent dynamic range control; for which an FFT-based implementation is also presented. Examples given in this paper reveal how the design can be tailored to perform a variety of tasks, using simple parameter manipulation; such as frequency-depended ducking for automatic-mixing purposes and high-resolution multi-band compression.

1. INTRODUCTION

A dynamic range compressor (DRC) is an indispensable tool found in many an audio engineer's toolbox. They are utilised extensively in the fields of music production, automatic-mixing, mastering, broadcasting and hearing aids. From the music production perspective, the main reason for their popularity lies with their ability to shape the amplitude envelope of transient audio signals; so that they may combine more cohesively with other signals present in the mix, or to rectify issues that an existing mix may exhibit. DRCs are also utilised to intentionally reduce the dynamic range of audio signals, as humans tend to associate a louder mix as being superior [1, 2]; which is especially important given stringent broadcasting regulations and the competition between broadcasters.

While the basic design of time-domain DRCs has evolved slowly over the years [3–8], more progress has been made with regard to diverse and creative ways of applying such devices in the field of music production. For example, most commercial implementations of modern DRCs offer the ability to influence the behaviour of the DRC using a different audio signal; a technique commonly referred to as *side-chain compression* [9]. To give a practical example, side-chain compression can be used to reduce the amplitude of a bass guitar signal to coincide with a kick drum transient. This temporally dependent "ducking" in the bass guitar signal may then allow the two instruments to better complement one another. In the production of dance music, this technique is commonly utilised in the more extreme case, by allowing the kick drum transient to conspicuously carve out temporally dependent *holes* in the amplitude of other signals; resulting in a *pumping* effect that pulses in time with the music [10].

Serial and parallel compression techniques have also become widely used, as they allow for even greater control of the amplitude envelope of signals by utilising multiple DRCs with different envelope detector settings [11, 12]. Recent research has also considered the digital modelling of traditional analogue DRC devices [8], as it is often desirable to replicate traditional audio processing techniques, in order to capture their unique distortion characteristics [13].

However, the majority of today's DRC designs are temporally-dependent and frequency-independent (referring to common designs described in [4, 14, 15]), which makes them unsuitable for certain tasks. For example, it might be desirable for the DRC to only derive its gain parameter from a specific frequency range of the input audio, or the user may want to use different DRC parameters for different frequency bands and have them operate independently. Therefore, to accommodate these scenarios, an element of frequency dependency has been adopted by two main sub-categories of DRC design, termed here as the *filtered side-chain compressor* (FSCC) [9, 16] and the *multi-band compressor* (MBC) [6, 17].

In the case of the FSCC, the amount of gain reduction applied to an input signal is influenced by a filtered version of the input signal. One common implementation is the *De-esser* design, which is typically used to reduce sibilance present in vocal recordings [9]. In this instance the gain factor is derived from the frequency range where the sibilance is most prevalent, through the use of a band-pass filter in their side-chain processing. There are also DRC designs that extend this control further and allow much more customisable filtering of the side-chain signal [18], which are suitable for applications where a single band-pass filter is insufficient. However, such designs do not feature in any formal publications. One potential drawback of the FSCC approach is that the calculated gain factor is applied to the whole frequency range of the input signal, which may not be desirable depending on the use case. Therefore, it might be beneficial if FSCC designs were able to apply the calculated gain factor to a user specified frequency range.

For the MBC approach, the input signal is divided into separate sub-bands and offers independent user controls for each of them. This allows for greater control of broadband signals, which therefore makes them especially useful for mastering applications; however, typically only two or three independent sub-bands are controllable by the user. MBCs with a higher number of sub-bands are used more commonly in hearing-aid designs [19–21]. However, due to their specific intended application, and their general lack of envelope detection or side-chain compression support; they are usually unsuitable for music production, mastering or automatic-mixing applications. Designs that are orientated towards these musical applications, such as the Soniformer audio plug-in [22], do not feature in any formal publications.

With regard to today's automatic-mixing approaches [23–25], many of them operate in the frequency or time-frequency domains and are applied off-line. The latest implementations in particular rely heavily on machine learning principles; therefore, they have removed the audio producer entirely from the mixing process and have generally yielded inferior results, when compared to their professionally mastered counterparts [25]. Improvements could be expected, by utilising higher frequency-resolution MBC designs that apply frequency dependent side-chain ducking. This would allow signals to frequency-dependently attenuate the amplitude of other signals, thus creating "space" for them in the mix, while also keeping the human element in the mixing process.

Therefore, the purpose of this paper is to present a generalised time-frequency domain DRC design, which can be utilised as a more customisable FSCC, an expanded MBC, and a potential semi-automatic-mixing method. The paper also provides details of an implementation of the proposed design, which is based on the Fast Fourier Transform (FFT). As demonstrated in the examples presented, the implementation provides promising results, while using a variety of different audio material and simple parameter manipulation.

This paper is organised as follows: Section 2 provides a short introduction on how DRCs generally operate and also describes the typical parameters that are offered to the user; Section 3 gives an example of a time-domain DRC design, which is then expanded to the proposed time-frequency domain design in Section 4; Section 5 then details how the proposed design was implemented, in order to provide the examples that are shown in Section 6; and Section 7 concludes the paper.

2. BACKGROUND

Typically, DRCs operate by duplicating the input signal to obtain two identical signals, henceforth referred to as the *input signal* and the *side-chain signal*. However, it is important to note that the side-chain signal can be an entirely different signal, in the case of the side-chain compression technique. The output signal is then created by attenuating the input signal based on a gain factor that is calculated from processing the side-chain signal. This side-chain processing typically comprises two main components, a *gain computer* and an *envelope detector* [14]. These two components allow many user parameters to be delivered, which typically include:

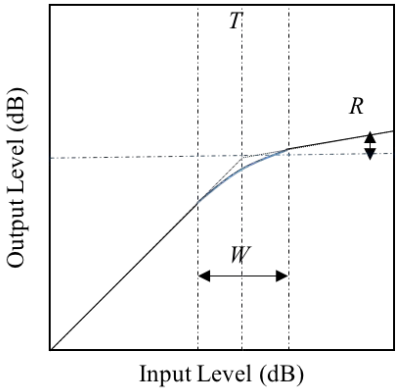


Figure 1: A depiction of the threshold T, ratio R and knee width W parameters. The blue line represents a soft knee, while the black dotted line represents a hard knee.

- *threshold*, above which attenuation of the input signal occurs;
- *ratio*, referring to the input/output ratio in decibels, which influences the gain reduction based on the extent to which the signal has surpassed the threshold;
- *knee width*, which permits a less abrupt attenuation around the threshold by allowing the gain factor to be determined by a gradual change in ratio. This is often termed as a *soft knee*, whereas the abrupt application of attenuation about the threshold is termed as a *hard knee* (see Fig. 1);
- *attack time*, which determines how quickly the envelope detector reaches the target gain reduction, as dictated by the threshold and ratio parameters;
- *release time*, which refers to how quickly the envelope detector returns to unity gain, when the side-chain signal ceases to exceed the threshold.

Additionally, the following parameters can also be offered to the user:

- *look-a-head*, the amount of time the input signal is delayed by, allowing the side-chain processing to react more reliably to acute transients;
- *make-up gain*, which is a gain factor applied after application of the DRC, as it is often desirable to increase the amplitude of the output in order to compensate for the loss in sound energy;
- *side-chain filter parameters*, in order to dictate which frequencies of the side-chain will influence the dynamic range compression of the input signal (used in FSCC designs).

Note that DRCs can either have a feed-forward or feed-back architecture [26]. However, feed-forward designs are generally more preferable, as the feed-back topology prohibits the use of a look-a-head parameter.

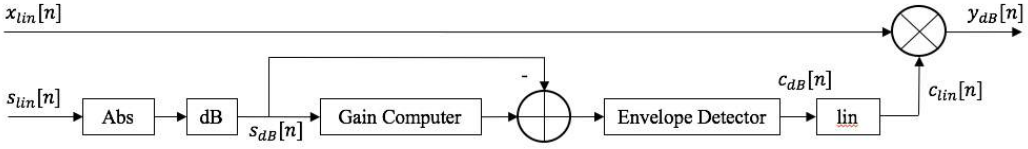


Figure 2: The block diagram of the time-domain feed-forward DRC, derived from [14]. Note that the s_{lin} signal can be identical to the x_{lin} signal, in order to perform typical dynamic range compression behaviour; or alternatively, they can be independent signals, in order to perform the side-chain compression technique.

3. A TYPICAL TIME-DOMAIN DRC DESIGN

An example of a feed-forward time-domain DRC design is described in [14] and features a second order gain computer and a smooth peak envelope detector; the block diagram of which is depicted in Fig. 2. Note that it is recommended to place the envelope detector after the gain computer, in order to mitigate certain artefacts described in [27].

This design operates in the log-domain, as this is a more perceptually relevant and effective domain for applying dynamic range compression

$$s_{dB}[n] = 20\log_{10}|s_{lin}|, \tag{1}$$

where s_{dB} and s_{lin} refer to the log-domain (decibels) and linear versions of the side-chain signal respectively and $|.|$ refers to the magnitude of a scalar. Alternatively, the root-mean-square (RMS) values of the input signal may be used in situations where the induced latency poses no issue [15].

The gain computer provides one target output gain per sample, $y_G[n]$, and is calculated as

$$y_G[n] = \begin{cases} s_{dB}[n] & 2(s_{dB}[n]-T) \leq -W \\ s_{dB}[n] + \frac{(\frac{1}{R}-1)(s_{dB}[n]-T+\frac{W}{2})^2}{2W} & 2|(s_{dB}[n]-T)| \leq W \\ T + \frac{(s_{dB}[n]-T)}{R} & 2(s_{dB}[n]-T) > W, \end{cases} \tag{2}$$

where T and W are the threshold and knee width parameters, respectively, expressed in decibels and R is the compression ratio.

The envelope detector is then formulated as [28]

$$v_L[n] = \begin{cases} \alpha_A v_L[n-1] + (1-\alpha_A)b_G[n] & b_G[n] > v_L[n-1] \\ \alpha_R v_L[n-1] + (1-\alpha_R)b_G[n] & b_G[n] \leq v_L[n-1], \end{cases} \tag{3}$$

where v_L refers to the output of the envelope detector in decibels; b_G is the input gain, calculated as the difference between the gain computer estimate and the original side-chain value $b_G = y_G - s_{dB}$; and the attack and release time constants are formulated as $\alpha_A = e^{-1/attack}$ and $\alpha_R = e^{-1/release}$ respectively, where the attack and release times are given as samples per millisecond.

The final gain factor c_{lin} is then obtained by returning to the linear domain

$$c_{lin}[n] = 10^{v_L/20}. \tag{4}$$

An example of this DRC design applied to a drum sample, is shown in Fig. 3. It can be observed that the gain reduction is temporally dependent on the transients of the side-chain signal; where transients with larger magnitude have resulted in a more dramatic gain reduction and the envelope detector has influenced the smoothness of the transitions between unity gain and gain reduction.

Note that this design can also be converted to an FSCC design by applying a filter $F(z)$ to the side-chain signal, such as a band-pass filter in the case of a De-esser.

4. THE PROPOSED TIME-FREQUENCY DOMAIN DRC DESIGN

This section details the proposed DRC algorithm, which assumes that the input and side-chain signals have been

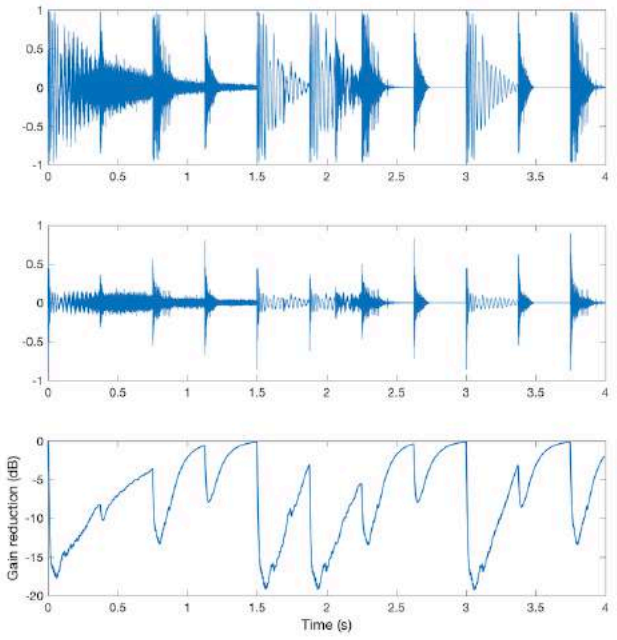


Figure 3: An example of a drum sample (top), after it has been passed through the time-domain DRC design (middle); using $T = -20$ dB, R = 8:1, W = 0 dB, Attack = 10 ms and Release = 80 ms. The gain reduction over time is also depicted (bottom), which shows how the DRC has responded to the transients in the input signal.

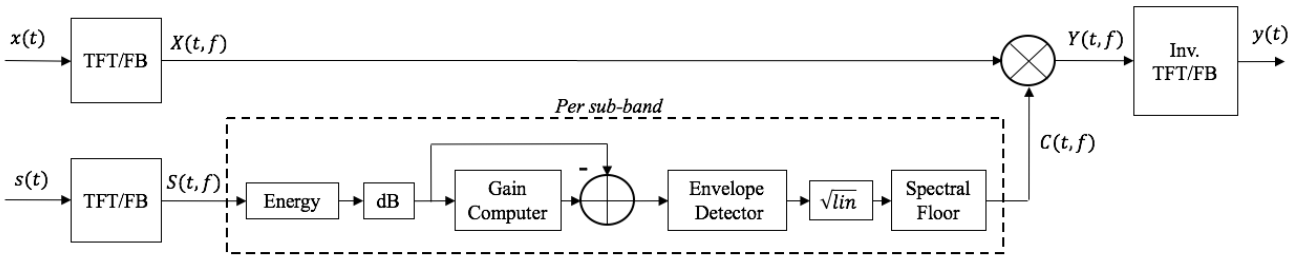


Figure 4: Block diagram of the proposed time-frequency domain DRC design, where TFT/FB denotes a time-frequency transform or perfect reconstruction filterbank. Note, however, that if $x(t)$ and $s(t)$ are the same signal, then the TFT/FB should be applied only once.

transformed into the time-frequency domain, via a Short-Time Fourier Transform (STFT) or a perfect reconstruction filterbank. Therefore, time-domain frames of N input samples $\tilde{x} = [x_0, ..., x_{N-1}]^T$ are now represented as their time-frequency domain counterparts $X(t, f)$; where t refers to the down-sampled time index and $f \in [0, ..., L-1]$ to the frequency indices given by a hop size L.

The proposed time-frequency domain architecture is shown in Fig. 4 and shares many similarities with its time-domain counterpart (see Fig. 2).

However, due to the change in domain, it is now the *energy* of each time-frequency index that is used as the input for the gain computer

$$X_G(t, f) = 10 \log_{10} |S(t, f)|^2, \tag{5}$$

where X_G and S refer to the side-chain energy and the side-chain signal respectively.

For the gain computer, the frequency-dependent energy values are compared against their respective threshold parameters, so that the dynamic range compression may be applied in a frequency-dependent manner (note that the time and frequency indices (t, f) have been omitted for compactness)

$$Y_G = \begin{cases} X_G & 2(X_G - T) \leq -W \\ X_G + \frac{(\frac{1}{R}-1)(X_G - T + \frac{W}{2})^2}{2W} & 2|(X_G - T)| \leq W \\ T + \frac{(X_G - T)}{R} & 2(X_G - T) > W, \end{cases} \tag{6}$$

where Y_G is the new target gain, expressed in decibels; T and W are the threshold and knee width parameters respectively, which are also expressed in decibels and R is the compression ratio.

The envelope detector also works in a similar manner to that of the time-domain implementation in (3)

$$V_G = \begin{cases} \alpha_A V_G z^{-1} + (1 - \alpha_A) B_G & B_G > V_G z^{-1} \\ \alpha_R V_G z^{-1} + (1 - \alpha_R) B_G & B_G \leq V_G z^{-1}, \end{cases} \tag{7}$$

where V_G refers to the output of the envelope detector in decibels and B_G refers to the input gain, calculated as the difference between the gain computer estimate and the energy of the side-chain $B_G = Y_G - X_G$. However, the attack and release time constants must now take into account the change of domain, so are formulated as: $\alpha_A = e^{-1/(attack/L)}$ and $\alpha_R = e^{-1/(release/L)}$ respectively, where the attack and release times are given as time-domain samples per millisecond.

Due to the nature of spectral processing, a spectral floor parameter λ is now recommended to mitigate audible artefacts that can occur when a frequency band is too harshly attenuated. Therefore, the final temporally-dependent and frequency-dependent gain factors $C(t, f)$ are given as

$$C(t, f) = \max \left(\lambda, \sqrt{10^{\frac{V_G(t,f)}{20}}} \right), \tag{8}$$

which should be multiplied element-wise with the input signal, in order to yield the dynamically compressed audio signal, which can be auditioned after performing an appropriate inverse time-frequency transform.

5. AN FFT-BASED IMPLEMENTATION OF THE PROPOSED DESIGN

In order to investigate the behaviour and performance of the proposed design, the algorithms presented were implemented using MatLab and then realised as a Virtual Studio Technology (VST) audio plug-in [1]. The graphical user interface (GUI) was designed using the open source JUCE framework and is depicted in Fig. 5.

The alias-free STFT filterbank in [29], which uses analysis and synthesis windows that are optimised to reduce temporal aliasing, was selected as the time-frequency transform for the implementation. A hop size of 128 samples and an FFT length of 1024 samples were selected, with a further sub-division of the lowest four bands to increase low-frequency resolution. This additional filtering is similar to the hybrid filtering that is performed in the filterbanks used for spatial audio coding [30]; therefore, there are 133 frequency bands in total, with centre frequencies that roughly correspond to the resolution of human hearing. A sample rate of 48 kHz and a frame size of 512 samples were hard coded into the plug-in.

[1] Various pre-rendered sound examples and Mac OSX/Windows versions of the VST audio plug-in, can be found on the companion web-page: http://research.spa.aalto.fi/publications/papers/smc17-fft-drc/

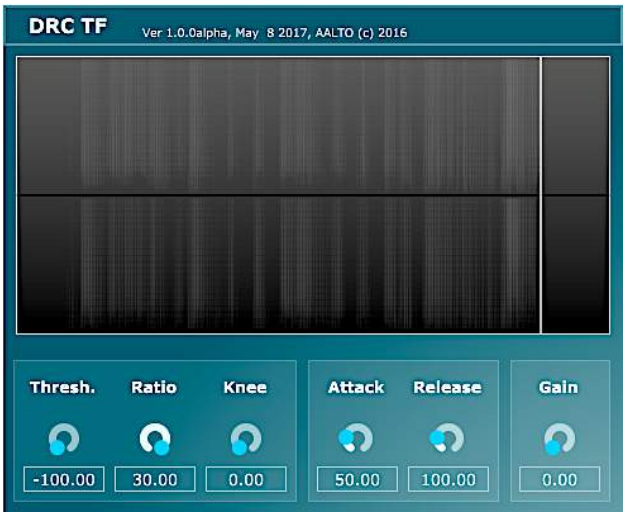


Figure 5: An image of the GUI for the VST implementation of the proposed algorithm. Note that the same parameters are applied for each sub-band; although frequency-dependent parameter selection is supported internally.

A total of 8 seconds of historic frequency-dependent gain factors $C(t, f)$, utilised in the processing section of the VST, are stored and displayed on the GUI to provide the user with visual feed-back of the extent of gain reduction.

Some of the drawbacks of the proposed design are: the numerous control parameters; increased latency and computational complexity. Therefore, in order to reduce the number of user parameters on the GUI, the same threshold, ratio, knee-width, attack time and release time values are used in the side-chain processing for all of the 133 sub-bands. However, this implementation should be viewed as a demonstration that the proposed algorithm can operate in real time; whereas the MatLab implementation has left all of the user parameters open to manipulation, in order to provide the examples shown in Section 6. With regard to the induced latency, due to the manner in which Digital Audio Workstations (DAWs) operate, delay-compensation can be introduced to make this drawback less of an issue.

6. EXAMPLE APPLICATIONS

The purpose of this section is to provide some perspective on the wide range of tasks that the proposed design can be applied to, via simple manipulation of the calculated gain factors. For example, a standard time-domain DRC design can be modelled by taking the mean of the frequency-dependent gain factors and applying this average to all of the sub-bands equally, which would enable an outcome similar to the example given in Section 3. However, as this results in increased latency and computational complexity, this is not recommended for this particular use case. However, a more appropriate use of this concept is to take the mean of the gain factors for a certain frequency range of the side-chain signal in order to to mimic a customisable FSCC. Now, unlike its time-domain counterpart, the proposed system is capable of applying this average gain factor to a specified frequency-range of the input signal. For example, high frequencies can now be attenuated during periods of high energy in the low frequencies.

The proposed design can also be used simply as an MBC, but with a higher frequency-resolution when compared to most MBCs employed in today's marketplace. The result after passing a drum sample though the proposed design is shown in Fig. 6b Since the individual components of the drum kit (snare, kick drum and hi-hat) are less likely to coincide simultaneously in time, or overlap substantially with regard to their frequency content, the music producer is able to use one instance of the proposed DRC and still dictate different parameters for each component. The alternative would be three independent instances of a time-domain DRC for each drum kit component.

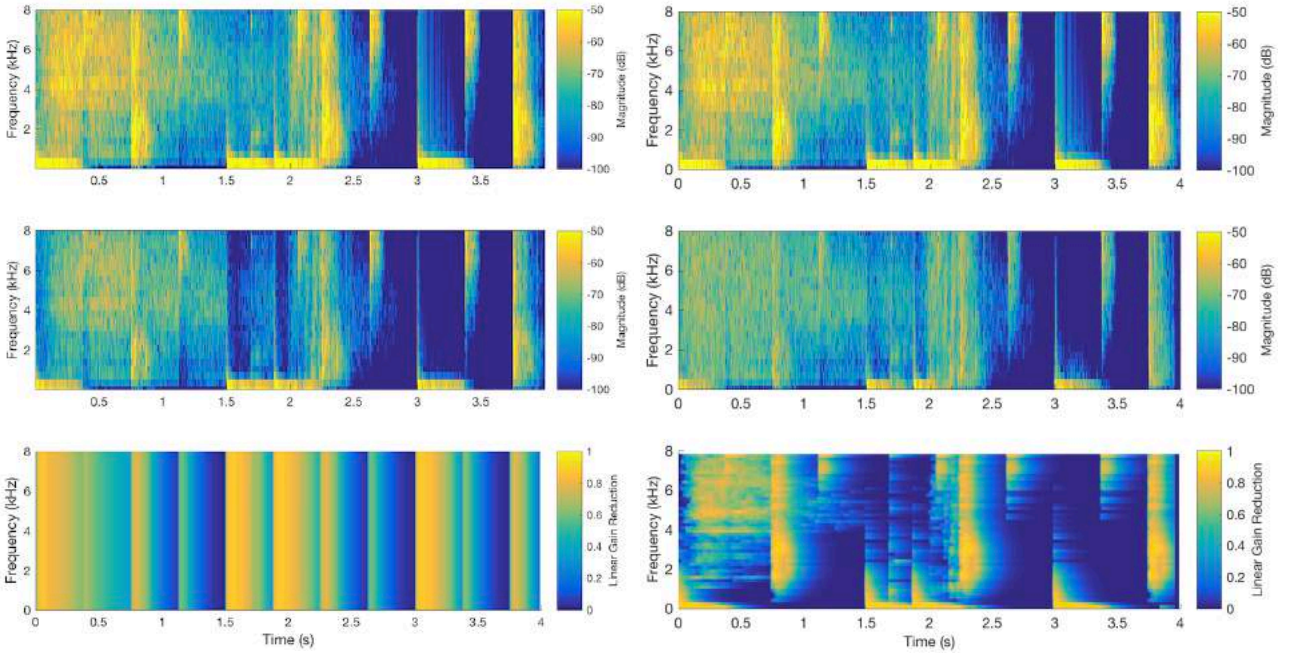
Another potential use for the proposed design is for automatic-mixing purposes. The example shown in Fig. 7 demonstrates how a popular music track, which can be approximated as uniformly distributed white noise, can be subjected to both frequency and temporally dependent side-chain compression, in order to accommodate an additional drum loop sample. One alternative method for attaining this result would be through the use of automation and digital audio filters, which would generally be considered a much more laborious process for accomplishing the same task.

In broadcasting, dynamic range compression is utilised heavily for both limiting and side-chain compression. The former is to ensure that the sound levels do not exceed the limits set by the broadcast regulators and by using the proposed design, only the frequency bands that exceed the limit will be attenuated. The latter is used to decrease the level of audio signals if speech signals exceed a particular threshold; an example utilising the proposed design is shown in Fig. 8.

Please note, however, that the user parameters selected for the presented examples (including Fig. 5) have been purposely configured to more harshly attenuate the signals, in order to make the changes in the spectrogram magnitudes easier to identify. More conservative gain reduction should be used during listening. Likewise, the frequency range shown on the spectrograms has been truncated to 8 kHz (to avoid depicting *dead-space*), despite the processing using a sample rate of 48 kHz.

7. CONCLUSION

This paper has outlined a design for a time-frequency domain DRC, which applies dynamic range compression with both temporally and frequency-dependent characteristics; allowing for increased flexibility when processing audio signals, owing to the greater degree of control that the system provides to the sound engineer. To the best of the authors' knowledge, this approach to dynamic range compression has historically featured mainly in algorithms orientated towards hearing aids, with regard to formal publications. Therefore, some applications have been presented, which highlight the potential usefulness of such a design in the fields of music production, mastering and broadcasting. These include frequency-dependent side-chain ducking for automatic-mixing purposes, and high



(a) $T = -20$ dB, R = 8:1, W = 0 dB, Att/Rel = 10/80 ms.

(b) $T = -50$ dB, R = 8:1, W = 0 dB, Att/Rel = 10/80 ms.

Figure 6: Spectrograms of the drum kit sample (top left, top right) and the output audio, when utilising the time-domain (middle left) and the proposed (middle right) DRC designs; also shown are the frequency-dependent gain factors for the time-domain (bottom left) and the proposed (bottom right) DRC designs.

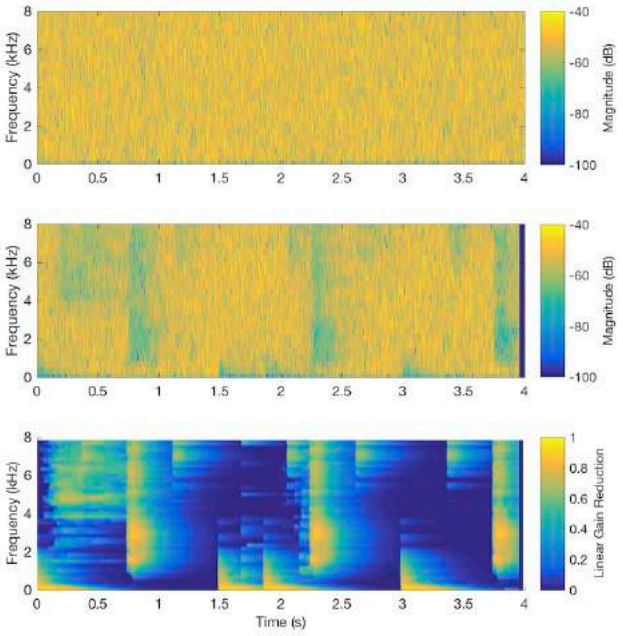


Figure 7: Spectrograms of white noise (top) and after frequency-dependent ducking using a drum kit sample (middle); also shown are the frequency-dependent gain factors (bottom). $T = -50$ dB, R = 8:1, W = 0 dB, Att/Rel = 10/150 ms.

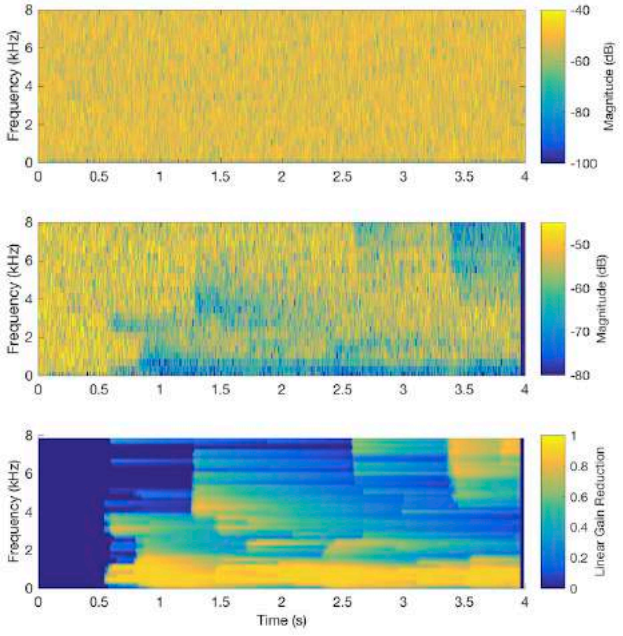


Figure 8: Spectrograms of white noise (top) and after frequency-dependent ducking using a speech sample (middle); also shown are the frequency-dependent gain factors (bottom). $T = -80$ dB, R = 20:1, W = 0 dB, Att/Rel = 10/800 ms.

resolution multi-band compression.

Regarding the presented FFT-based implementation, future work could investigate how best to design a GUI that accommodates the additional parameters, or study how one might automate them; such as in [31].

8. REFERENCES

[1] E. Vickers, "The loudness war: Background, speculation, and recommendations," in *Audio Engineering Society Convention 129*, San Francisco, CA, USA, Nov. 2010.

[2] M. Wendl and H. Lee, "The effect of dynamic range compression on perceived loudness for octave bands of pink noise in relation to crest factor," in *Audio Engineering Society Convention 138*, Warsaw, Poland, May 2015.

[3] B. Blesser, "Audio dynamic range compression for minimum perceived distortion," *IEEE Transactions on Audio and Electroacoustics*, vol. 17, no. 1, pp. 22–32, Mar. 1969.

[4] E. F. Stikvoort, "Digital dynamic range compressor for audio," *Journal of the Audio Engineering Society*, vol. 34, no. 1/2, pp. 3–9, Jan./Feb. 1986.

[5] A. Schneider and J. Hanson, "An adaptive dynamic range controller for digital audio," in *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria, BC, Canada, May 1991, pp. 339–342.

[6] J. C. Schmidt and J. C. Rutledge, "Multichannel dynamic range compression for music signals," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-96)*, vol. 2, Atlanta, GA, USA, May 1996, pp. 1013–1016.

[7] U. Zölzer, *Digital Audio Signal Processing*. John Wiley & Sons, 2008.

[8] P. Raffensperger, "Toward a wave digital filter model of the Fairchild 670 limiter," in *Proc. 15th Int. Conf. Digital Audio Effects (DAFx-12)*, York, UK, Sept. 2012, pp. 195–202.

[9] A. J. Oliveira, "A feedforward side-chain limiter/compressor/de-esser with improved flexibility," *Journal of the Audio Engineering Society*, vol. 37, no. 4, pp. 226–240, Apr. 1989.

[10] Music Radar, "How to create classic pumping sidechain compression," http://www.musicradar.com/tuition/tech/how-to-create-classic-pumping-sidechain-compression, 2015, [Online; accessed 01-Feb-2017].

[11] Sound on Sound, "Parallel compression," http://www.soundonsound.com/techniques/parallel-compression, 2013, [Online; accessed 27-Jan-2017].

[12] MusicTech, "Pro Tools tutorial: Cutting edge production techniques—Serial compression serial compression," http://www.musictech.net/2015/05/pro-tools-serial-compression/, May 2015, [Online; accessed 28-Jan-2017].

[13] A. Moore, R. Till, and J. Wakefield, "An investigation into the sonic signature of three classic dynamic range compressors," in *Audio Engineering Society Convention 140*, Paris, France, May 2016.

[14] D. Giannoulis, M. Massberg, and J. D. Reiss, "Digital dynamic range compressor design: Tutorial and analysis," *Journal of the Audio Engineering Society*, vol. 60, no. 6, pp. 399–408, June 2012.

[15] G. W. McNally, "Dynamic range control of digital audio signals," *Journal of the Audio Engineering Society*, vol. 32, no. 5, pp. 316–327, May 1984.

[16] R. J. Cassidy, "Dynamic range compression of audio signals consistent with recent time-varying loudness models," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, Montreal, QC, Canada, May 2004, pp. 213–216.

[17] S. H. Nielsen and T. Lund, "Level control in digital mastering," in *Audio Engineering Society Convention 107*, New York, USA, Sept. 1999.

[18] Fab Filter, "Manual: Pro MB," http://www.fabfilter.com/help/ffpromb-manual.pdf, 2016, [Online; accessed 31-Jan-2017].

[19] D. K. Bustamante and L. D. Braida, "Multiband compression limiting for hearing-impaired listeners," *Journal of Rehabilitation Research and Development*, vol. 24, no. 4, pp. 149–160, 1987.

[20] B. Kollmeier, J. Peissig, and V. Hohmann, "Real-time multiband dynamic compression and noise reduction for binaural hearing aids," *Journal of Rehabilitation Research and Development*, vol. 30, no. 1, pp. 82–94, 1993.

[21] E. Lindemann, "The continuous frequency dynamic range compressor," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA, Oct. 1997, pp. 1–4.

[22] Voxengo, "Soniformer audio plugin," http://www.voxengo.com/product/soniformer/, 2016, [Online; accessed 31-Jan-2017].

[23] E. Perez-Gonzalez and J. Reiss, "Automatic equalization of multichannel audio using cross-adaptive methods," in *Audio Engineering Society Convention 127*, New York, NY, USA, Oct. 2009.

[24] J. D. Reiss, "Intelligent systems for mixing multichannel audio," in *Proc. IEEE 17th Conf. Digital Signal Processing (DSP)*, Corfu, Greece, July 2011, pp. 1–6.

[25] S. I. Mimilakis, K. Drossos, T. Virtanen, and G. Schuller, "Deep neural networks for dynamic range compression in mastering applications," in *Audio Engineering Society Convention 140*, Paris, France, May 2016.

[26] J. S. Abel and D. P. Berners, "On peak-detecting and RMS feedback and feedforward compressors," in *Audio Engineering Society Convention 115*, New York, NY, USA, Oct. 2003.

[27] S. Wei and W. Xu, "FPGA implementation of gain calculation using a polynomial expression for audio signal level dynamic compression," *Acoustical Science and Technology*, vol. 29, no. 6, pp. 372–377, 2008.

[28] P. Dutilleux, K. Dempwolf, M. Holters, and U. Zölzer, *Nonlinear processing, DAFX: Digital Audio Effects, Second Edition*. Chichester, UK: John Wiley and Sons, Ltd, 2011.

[29] J. Vilkamo, "Alias-free short-time Fourier transform–A robust time-frequency transform for audio processing," https://github.com/jvilkamo/afSTFT, 2015, [Online; accessed 05-Feb-2017].

[30] J. Breebaart, S. Disch, C. Faller, J. Herre, G. Hotho, K. Kjörling, F. Myburg, M. Neusinger, W. Oomen, H. Purnhagen *et al.*, "MPEG spatial audio coding/MPEG surround: Overview and current status," in *Audio Engineering Society Convention 119*, New York, NY, USA, Oct. 2005.

[31] D. Giannoulis, M. Massberg, and J. D. Reiss, "Parameter automation in a dynamic range compressor," *Journal of the Audio Engineering Society*, vol. 61, no. 10, pp. 716–726, Oct. 2013.