Intelligent Equalization on Multi-tracks Using Yule-Walker for Spectral Matching

Harrison Zafrin
NYU
hzz200@nyu.edu

ABSTRACT

In this paper, intelligent equalization is implemented using a dynamic IIR filter where the coefficients are generated via the Yule-Walker method. While in previous research this technique was shown to be an effective form of intelligent mastering, the goal of this paper is to test these methods in a multi-track setting for future implementation in autonomous mixing systems. In this off-line MATLAB implementation, audio examples are provided to show the result of filtering, and the spectras are compared to objectivley observe the results. To look at the code used to run this experiment, visit https://github.com/bombsandbottles/Intelligent-Equalization.

1. INTRODUCTION

The spectral response of a sound/recording is highly correlated with genre/instrument type with the data supporting the existence of idealized target frequency responses that professional engineers uniformly conform to [1]. With this in mind, it is often the plight of novice audio engineers to achieve these idealized spectral distributions. This is because mixing as a whole is notoriously difficult, requiring years of experience, expert taste, and fine tuned hearing. Therefore, the goal of this research is to eliminate this struggle, and shift the focus back to writing phenomenal music by autonomously employing best practices in sound engineering.

Autonomous mastering has already made it to market in the form of Landr (https://www.landr.com/), an on-line tool which uses techniques not foreign to those discussed in this paper and past research [2]. However while past research was focused on using intelligent equalization as a mastering tool, this paper is focused on testing it on a stem by stem basis. Consider the following example: An engineer has a vocal that is "muddy" in its spectral character, lacking the polish it needs to stand out as a well engineered stem. With the technology presented in this paper, the engineer can choose a preset EQ curve of which his unprocessed vocal can morph based on genre/instrument type, instantly giving his vocal "brilliance", conforming it to industry standards.

The ideal situation just presented is tested in this paper. The following vocal stems were analyzed to form an idealized top-40 female pop vocal: [Britney Spears - Break The Ice, Christina Aguilera - Your Body, Katy Perry - Firework, Katy Perry - ET, Kesha - Die Young]. An unprocessed female vocal stem is then run through a time-varying filter to match the target spectra dictated by this genre/instrument type.

2. ALGORITHM AND IMPLEMENTATION

In this section, the process on how to perform intelligent equalization using dynamic IIR filtering is detailed. In summation, the process can be broken down into the following steps: First, a target/idealized frequency response for which our analyzed audio will be morphed to is acquired via the averaging of multiple magnitude spectra. Once we have our target equalization curve, a transfer function is created on a frame by frame basis through the comparison of our target equalization curve spectra and the incoming audio. Using the Yule-Walker method, the resulting transfer function from the magnitude spectra comparison is converted into a set of IIR filter coefficients. The transfer functions across frames are smoothed to provide a sense of cohesion, and the filter is applied via a difference equation.

2.1 Creating the Target Spectra

To create the target spectra, the method used in [1] is implemented. Using this method, we perform a 4096 point STFT on an audio stem such that it results in a $K \times T$ matrix where K is the frequency bin number and T is the frame number across time. This matrix known as $X(k, \tau)$ is then averaged across τ to give the mean frequency response of the individual audio stem. We opt for a 50% overlap between frames with a hanning window:

$$\bar{X}(k) = \frac{\sum_{\tau} |X(k, \tau)|}{\left(\frac{x_{len}}{w_{len}}\right) + 1} \tag{1}$$

To account for differences in the magnitudes between audio files, normalization is performed on each averaged spectra in the dataset such that the frequency bin sum is 1:

$$\tilde{X}(k) = \frac{\bar{X}(k)}{\sum_k \bar{X}} \tag{2}$$

And then converted into a cumulative distribution function across the bins:

$$X_c(k) = \sum_{i=0}^{k} \tilde{X}(i) \tag{3}$$

After concatenating each cumulative distribution X_c into the matrix $X_c(k, n)$ where N is the number of tracks contributing to the target curve, a mean calculation is performed on each cumulative distribution ($\overline{X_c(k)}$). From here, the overall average spectrum across all tracks can be computed as the difference between adjacent averaged cumulative distributions multiplied by the average magnitude of all tracks in the dataset S:

$$\bar{X}_{AV}(k) = \frac{\sum_k \bar{X}(k)}{S} \left(\overline{X_c(k)} - \overline{X_c(k-1)} \right) \tag{4}$$

The resulting target equalization curve can be seen in figure 1. As recommended by the literature in [2], a 17-point moving average filter is applied to the spectrum at only the frequencies above 200hz. The smoothed version of the spectra is used as the target curve in this experiment.

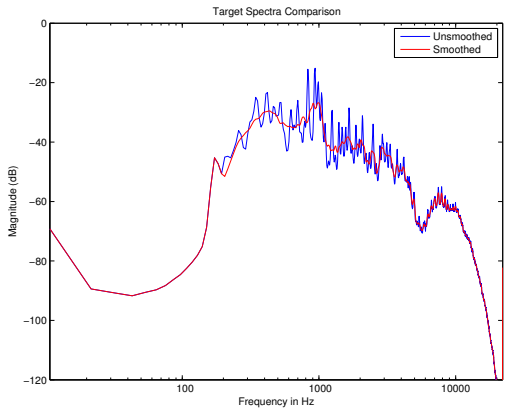

Figure 1. Target equalization curve for a top-40 female pop vocal.

2.2 Hysteresis Noise Gate and Loudness Measurement

In an effort to save computational cost and avoid undesirable transfer functions generated from silence, a hysteresis noise gate in conjunction with the ITU-R BS.1770 loudness measurement is implemented to detect active frames. In this loudness calculation, the signal is preemptively filtered to emulate the frequency response of the human ear before a subsequent modified RMS measurement. The filter coefficients for a 48khz sampling rate are outlined in the original publication [3], while a formula to calculate the coefficients for different sampling rates is given here [4]. These filters can be easily implemented in MATLAB by supplying the proper filter coefficients to the `filt()` function.

While the ITU-R BS.1770 standard was devised with mastered programmes in mind, autonomous mixing systems have been using it to quantify fader position coefficients. Therefore, post K-filtering, the loudness for a given track m is calculated by the following equation:

$$L_m[n] = 0.691 \cdot 10log_{10} \cdot \left(\frac{1}{N} \sum_{n=0}^{N-1} xg_m^2[n] \right) \tag{5}$$

Here we see a mean-squaring operation is performed across an N long frame of audio where N represents 3 seconds in samples. The resulting value is converted to dB and the corresponding LU unit is loaded into the loudness vector L_m. In this experiment, the window size of our audio corresponds to the window size of our subsequent spectral analysis (4096 point). Therefore, in order to simulate the smoothness of a 3 second window, an exponential moving average filter is applied to our resulting L_m vector as in [4]. The following exponential moving average filter is used in this experiment with an α value of 0.9 based on empirical experimentation:

$$y[n] = (\alpha \cdot y(n-1)) + (1-\alpha) \cdot x(n) \tag{6}$$

Finally, a hysteresis noise gate is applied with the thresholds of -25 and -30 LUFS to determine whether or not a frame is active. Active frames will be analyzed to generate a transfer function for filtering, while inactive frames will take on the transfer function of the previous active frame to provide continuity in the changing of filter states.

2.3 Spectral Analysis

The to-be equalized audio is analyzed in the same fashion as the target EQ curve. Therefore, a 4096 point FFT is performed with a hanning window and 50% overlap between frames. Since the resulting magnitude spectrum will appear in the denominator of the IIR design, problems may arise if values are too small. Therefore, magnitudes below 0.0001 are either set to 0 or converted to 0.0001 based on empirical experiments.

2.4 Obtain Desired Magnitude Response

To obtain the desired magnitude response per frame, the following calculation is performed:

$$|H_d(\omega)| = \frac{|T(\omega)|}{|X(\omega)|}, \quad \omega \in (0, \pi) \tag{7}$$

Here our target magnitude spectrum $|T(\omega)|$ is divided by our active frame $|X(\omega)|$ at the following frequencies: 0, 20, 25, 31.5, 40, 50, 63, 80, 100, 125, 160, 200, 250, 315, 400, 500, 630, 800, 1000, 1250, 1600, 2000, 2500, 3150, 4000, 5000, 6300, 8000, 10000, 12220, 16000, 22050. The resulting division allows us to compress the amount of information into 33 points yielding us our desired transfer function per frame known as $|H_d(\omega)|$. $|H_d(\omega)|$ is subsequently normalized for each frame such it lies between 0 and 1 to prevent overshooting.

2.5 Filter Curve Smoothing

Since our transfer function $|H_d(\omega)|$ is dependent on the incoming active frame, large changes in the frequency response of adjacent frames can lead to unpleasant sonic results. Therefore, to provide a sense of cohesion between changing filter states, an exponential moving average filter is applied to $|H_d(\omega)|$ with an α value of 0.9 based on empirical experiments:

$$H'_d(\omega_n) = \alpha \cdot H'_d(\omega_{n-1}) + (1-\alpha) \cdot H_d(\omega_n) \quad (8)$$

2.6 Obtaining IIR Coefficients Using the Yule-Walker Method

The Yule-Walker method [5] is used to perform a least-squares fitting to the desired frequency response $|H_d(\omega)|$ for the active frame. Due the complexity of the math behind the Yule-Walker method, the MATLAB function `yulewalk()` was used to generate a 16th order recursive IIR filter with coefficients B and A that best approximate the target frequency response:

$$\frac{B(z)}{A(z)} = \frac{b(0) + b(1) \cdot z^{-1} + \cdots + b(n) \cdot z^{-n}}{1 + a(1) \cdot z^{-1} + \cdots + a(n) \cdot z^{-n}} \quad (9)$$

2.7 Applying The Filter

In this MATLAB implementation, the `filt()` function cannot be used as it does not have a memory of previous x and y values for each frame. Therefore, a manual filtering is implemented via the following difference equation:

$$y[n] = b(1) \cdot x[n] + b(2) \cdot x[n-1] + \cdots + b(17) \cdot x[n-16] - a(2) \cdot y[n-1] - \cdots - a(17) \cdot y[n-16]$$

To accomplish this, two buffers are created such that the previous 16 values of $x[n-1]$ to $x[n-16]$ and $y[n-1]$ to $y[n-16]$ can be stored and updated across frames.

3. RESULTS

As can be seen in figures 2 and 3, objective results from the experiment show that the algorithm in its current form is not functioning correctly. It is possible this can be attributed to the simplified implementation of section 2.6. In the original implementation of this approach on mixed 2-tracks [2], the frequencies discussed are centered frequencies, implying a summation of energy across multiple bins to represent perceptual human hearing. Unfortunately due to the limitations of the `yulewalk()` function, this was not able to be correctly implemented in time.

Also simplified is the implementation of filter-curve smoothing in section 2.5 of which an empirically attained value of 0.9 was used for α to achieve the most sonically pleasing result. In this case, vagueness in the literature prevented proper implementation. For the sake of observing what is possible with a working algorithm, please see [2].

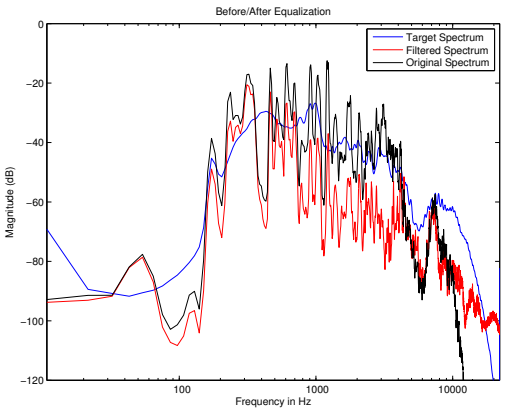

Figure 2. Equalization results for a female pop vocal.

4. CONCLUSION

While a target equalization curve based on genre/instrument type was extracted successfully, its use within a time-varying filter as intelligent equalization was not implemented correctly. Despite the objective results, subjective listening of the output audio examples warrants future research and further adjustment to the code. The processed female vocal stem exhibits an enhanced spectral character in the upper frequencies, giving it a bright and shimmery pop vocal sound. Future research can focus on the idea of time-varying filter states based on amplitude. For example, quieter played parts exhibit a different spectral envelope than those which are played with intensity. Measurements in magnitude within the upper frequencies as well as an amplitude gating mechanism can serve to capture different spectral envelopes for different performance intensities. This would add a sense of dynamics to the time-varying filter instead of keeping its spectral response constant through the duration of a song. It is also necessary to train the target equalization curve with more data. In this experiment, only 5 stems were used as reference, where as previous research utilizes a sample size of 600. Despite these drawbacks, the author believes that the data and methods presented here are convincing enough to pursue further, and looks forward to future developments in the field.

5. REFERENCES

[1] P. D. Pestana, Z. Ma, J. D. Reiss, A. Barbosa, and D. A. Black, "Spectral characteristics of popular commercial recordings 1950-2010," in *Audio Engineering Society Convention 135*. Audio Engineering Society, 2013.

[2] Z. Ma, J. D. Reiss, and D. A. Black, "Implementation of an intelligent equalization tool using yule-walker for music mixing and mastering," in *Audio Engineering Society Convention 134*. Audio Engineering Society, 2013.

[3] I. Rec, "Bs. 1770 algorithms to measure audio programme loudness and true-peak audio level," 2006.

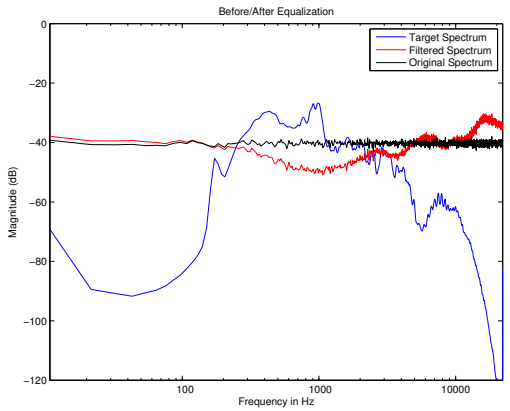

Figure 3. Equalization results on a white noise test.

[4] S. Mansbridge, S. Finn, and J. D. Reiss, "Implementation and evaluation of autonomous multi-track fader control," in *Audio Engineering Society Convention 132*. Audio Engineering Society, 2012.

[5] B. Friedlander and B. Porat, "The modified yule-walker method of arma spectral estimation," *Aerospace and Electronic Systems, IEEE Transactions on*, no. 2, pp. 158–173, 1984.