Real-Time "Wah" GUI Implementation

Peter Burrows, MS EE '15
Electrical Engineering Department
School of Engineering and Applied Science
Columbia University in the City of New York

***Abstract*—The popular sound generated by the "wah-wah" foot-controlled pedal has inspired the implementation of a real-time digital manual-wah effect. A graphical user interface (GUI) has been constructed such that users can input and play an audio file, control the frequency sweep motion, and listen to the results in real-time.**

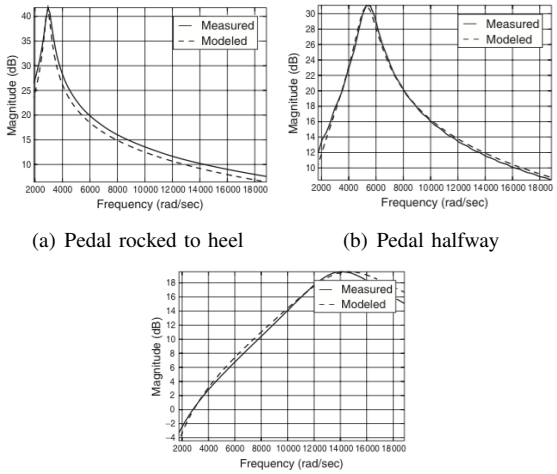
I. Background/Problem Formulation

The sound of the frequency sweep generated by "wah-wah" filter-based devices has appealed to music listeners of all genres. In signal processing, this frequency sweep is defined as an oscillation of the center frequency about a bandpass filter [1]. Audibly, the frequency response of the sweeping motion mimics the sound of a whining human voice, which inspired the Dunlop Manufacturing Company's fabrication of the GCB95 Cry Baby "wah-wah" foot-controlled guitar pedal in 1966. Blues-rock guitar legend, Jimi Hendrix, was, then, introduced to the device and pioneered the movement that carried the unique sound to the mainstream by the end of the 60s.

Currently, the "wah-wah" filter is prevalent in music of all genres. Modifications and parameter embellishments, such as band pass width and peak selection, are implemented differently depending on the genre. In general, however, one can categorize the "wah-wah" effect into two broader classes:

- In **automatic-wah** designs, the center frequency of the bandpass filter oscillates across the band at a constant rate that is set by the user [1]. This preset, linearly oscillating motion prohibits the sweeping flexibility that a manual-wah filter offers; nonetheless, the various implementations of the auto-wah design have proven to be successful for numerous musicians; among these musicians is Peter Frampton, whose implementation can be heard throughout his 1975 release, titled "Show Me the Way."
- In **manual-wah** designs, the user controls the rate at which the center frequency glides across the pass band. "Wah-Wah" foot-controlled pedals are built to mechanically rock from heel to toe, and toe to heel. When positioned at the heel, the lower frequencies are emphasized, and when positioned at the toe, the higher frequencies are emphasized. This property permits users to sweep the center frequency across the signal in nonlinear motions. To gain an understanding of this mechanical property, an image of Dunlop's Cry Baby is provided (Figure 1) along with the frequency response of the Cry Baby at three different mechanical positions (Figure 2).

Fig. 1. Dunlop's GCB95 Crybaby



(a) Pedal rocked to heel (b) Pedal halfway

(c) Pedal rocked to toe

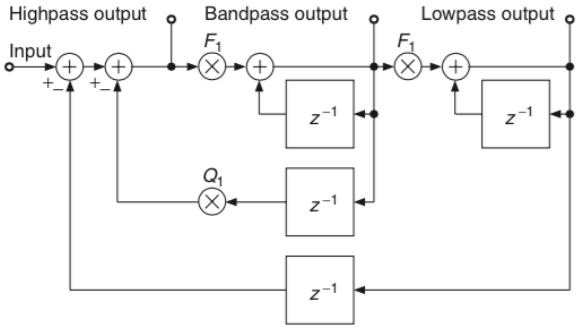
Fig. 2. Measured Frequency Response of the Crybaby [2]

For modern blues rock guitarists, the analog manual-wah foot-controlled pedal dominates the recording and performance space. Currently, Dunlop retails its quality-engineered GCB95 Cry Baby pedal at roughly $115. For guitarists who are not quite ready to perform, record, or invest in a pedal, experimentation and amusement can be achieved through a cost-effective, digital product. The following text presents an implementation of the latter suggestion.

II. Methodology

The implementation of the real-time manual-wah heavily draws from the ideas presented by [2], [3], and [4]. In particular, [3] explains that the 2nd order state variable filter works best for audio applications because the bandpass width and center frequency variables are set independent of the

system, an ideal situation for a manual-wah implementation, which permits the users change the center frequency in a nonlinear motion. The form of the digital state variable filter, as depicted in Figure 3(a), reduces to the difference equations seen in 3(b).



(a) Form

$$y_l(n) = F_1 y_b(n) + y_l(n-1)$$
$$y_b(n) = F_1 y_h(n) + y_b(n-1)$$
$$y_h(n) = x(n) - y_l(n-1) - Q_1 y_b(n-1),$$

(b) Difference Equations

$$F_1 = 2\sin(\pi f_c/f_S) \qquad Q_1 = 1/Q.$$

(c) Parameters

Fig. 3. Digital State Variable Filter [3]

The drawback of the state variable filter is its instability at higher center frequency parameter values, defined as F_1, and smaller bandpass width values, defined as Q_1 [3]. During the manual-wah implementation, the bandpass width value was the primary concern because the center frequencies were relatively small. Recalling Figure 2, which depicted the frequency response of the Cry Baby at three different mechanical positions, the minimum frequency of the bandpass was calculated as 2000 rad/s (roughly 318 Hz) and the maximum frequency off the bandpass was calculated as 18000 (roughly 2865 Hz). Both of these frequencies were preset in the GUI, which will be discussed in the next section. In terms of the bandpass width, a value of 0.05 was preset; however, depending on the desired sound, the GUI presents the bandpass width as a modifiable attribute.

A. GUI

A simple and functional user interface, as depicted in Figure 4, was constructed through the aid of tutorials and examples provided by [5] and [6]. The GUI controls are described as follows:

- First, the analog foot-controlled component of the Cry Baby is analogous to the bottom slider in the GUI. The slider residing at a location to the absolute left emulates a mechanical manual-wah pedal pressed down at the heel, which emphasizes the lower frequencies of the bandpass filter. Likewise, the slider residing at a location to the absolute right emulates a mechanical manual-wah pedal pressed down at the toe, which emphasized the higher frequencies of the bandpass filter.
- Second, the bandpass width is controlled by the top slider. Values at the bottom of this range begin to yield instability, and values at the top of this range begin to yield bypass.
- Third, the bypass checkbox was included for the user to hear the input audio as a clean signal that does not pass through the manual-wah filter. When the bypass checkbox is selected, the sliders will not have any effect on the input audio signal.

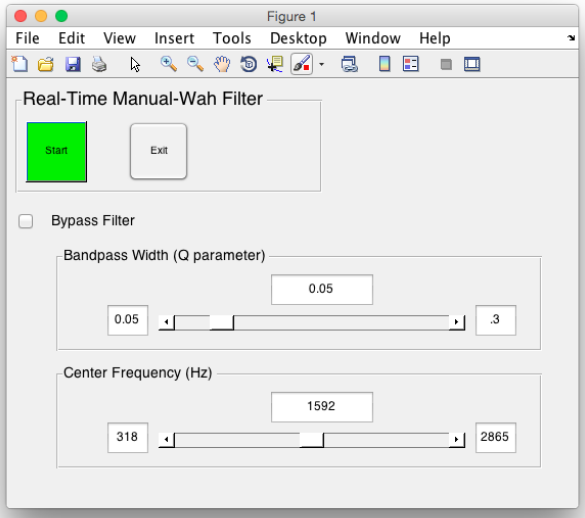


Fig. 4. Real-time wah graphical user interface (GUI) in MATLAB

B. Implementation of Real-Time Processing

In general, the real-time processing involved was carried out using two steps:

- First, the audio file was first split into segments of 512 samples, each of which corresponded to 0.5 seconds in duration. In MATLAB, each segment was mapped to a floating point array data type spanning 512 rows and 2 columns.
- A loop was then established to process each 512-length segment, play each 512-length segment, then increment to the next segment upon the next iteration. The processing component of the loop involved polling the slider values inputted by the user, upon the start of each iteration, and updating the difference equations using the values. This polling process is discussed further in the subsection, *Linearly Interpolating the Slider Values*.
- **Note**: Processing the values in smaller segments lengths would yield better audible results; however, segment sizes less than 512 samples compensated MATLAB's ability to playback the processed results without a significant audible delay.

After all of the segments were processed and played, an output audio file was written to the disc as `outfile.wav` under the working directory.

C. Linear Interpolating the Slider Values

In MATLAB, the rate at which the GUI polls the slider values is unpredictible due to the operating system's scheduling delays on which the software is deployed. To counter this unpredictability, the values of the slider were linearly interpolated to purify the audible output. For example, if the last five slider values recorded were the following sequence - 1, 2, 5, 5, 5 - then the linear interpolation algorithm would overwrite the values to produce the sequence defined as follows - 1, 2, 3, 4, 5 - in turn, mitigating the uneven audible output at the expense of the accuracy of the user's input.

III. Results

The manual-wah GUI was verified using Simulink's spectral analyzer module. Figure 5 is a screen shot of the spectral analyzer taken from a demonstration of the wah GUI in which a trial was run to test an oscillating frequency sweep. From the figure, the red, yellow, and orange hues indicate the sweeping path of the center frequency across the bandpass filter. The video demonstration that yields these results can be found at [7].

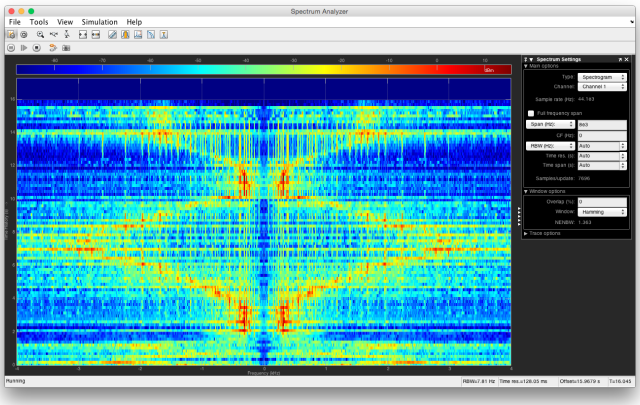


Fig. 5. STFT of audio processed using the GUI

Audibly, the GUI output does resemble the sound of a manual-wah filter and is proven through the frequency measurements recorded by the spectrum analyzer.

IV. Future Works

In the implementation of the real-time manual-wah filter, MATLAB's well-documented presence in conjunction with its ease of use was traded for performance.

A future implementation or optimization would require a software development environment with more computational power than that of MATLAB. Such an environment would offer an audible response that is more sensitive to the user's actual input. In order to accomodate the real-time constraint, the signal processing component would be required to interact more efficiently with memory. On behalf of the slider polling, advanced interpolation techniques would be required to yield an audible output that more accurately models the user's sweeping motion. Ultimately, in choosing the platform for a future implementation, it is critical to realize that performance will be increased at the expense of programming ease and analytics; both of which were available in MATLAB, and both of which were enablers for rapidly prototyping the GUI implementation.

References

[1] P. Dutilleux, M. Holters, S. Disch and U. Zolzer, "Filters and delays," in *DAFX: Digital Audio Effects*, 2nd ed. Chichester, West Sussex, U.K : John Wiley & Sons Ltd, 2011, ch. 2, sec. 2.4.1, pp. 67 - 68.

[2] V. Valimaki, S. Bilbao, J. O. Smith, J. S. Abel, J. Pakarinen and D. Berners, "Virtual analog effects," in *DAFX: Digital Audio Effects*, 2nd ed. Chichester, West Sussex, U.K : John Wiley & Sons Ltd, 2011, ch. 2, sec. 12.2.4, pp. 480 - 482.

[3] P. Dutilleux, M. Holters, S. Disch and U. Zolzer, "Filters and delays," in *DAFX: Digital Audio Effects*, 2nd ed. Chichester, West Sussex, U.K : John Wiley & Sons Ltd, 2011, ch. 2, sec. 2.2, pp. 48 - 51.

[4] D. Marshall, Y. Lai, "Digital audio effects," (2011 Jan. 30). Cardiff University, *Course Webpage Module: CM0268*. pp. 350 - 362. [Online]. Available: `http://www.cs.cf.ac.uk/Dave/CM0268/PDF/10_CM0268_Audio_FX.pdf`

[5] "Creating apps with graphical user interfaces in MATLAB" [Online]. Available: `http://www.mathworks.com/discovery/matlab-gui.html?refresh=true`

[6] M. Fig. (2009, July 27). "41 complete GUI examples" [Online]. Available: `http://www.mathworks.com/matlabcentral/fileexchange/24861-41-complete-gui-examples`

[7] P. Burrows. (2014, Dec. 18). "Real-Time Wah GUI Implementation in MATLAB (Guitar Demo)" [Online]. Available: `https://www.youtube.com/watch?v=WRH_n2Wz49o`