Proceedings of Urban Intelligence: Machine Learning in Smart City Solutions - CONFSEML 2024
DOI: 10.54254/2755-2721/65/20240515

Voice modulation design implemented through FPGA and MATLAB

Yan Zhang[1,4], Zhipeng Li[2,3,5]

[1]Chengdu University of Technology, No. 218, San Duan, University City Road, Lingang Economic and Technological Development Zones, Yibin City, Sichuan province.
[2]Chengdu University of Technology, No. 1 er Xian Qiao Dong San Lu, Chenghua District, Chengdu City, Sichuan Province
[3]Corresponding author

[4]2648174864@qq.com
[5]2091054@qq.com

Abstract. With the rapid development of network technology, there has been an increased demand for real-time signal processing, particularly in the processing speeds of audio and video signals. This paper designs a program for human voice pitch transformation, applied to FPGA hardware, leveraging FPGA's rapid processing filters to achieve real-time voice pitch transformation. In the FPGA program design process, by studying and applying FRM filter design algorithms, the amount of computational data is reduced, thereby enhancing processing speed. Finally, numerical simulation is implemented based on Matlab to verify the feasibility of the program. This provides an approach for the intelligent real-time processing of audio signals.

Keywords: FPGA, MATLAB, Voice Transformation, Real-Time Processing, Simulation.

1. Introduction

In daily life, there are situations where it is inconvenient to reveal one's true voice, such as in news reports where some interviewees prefer not to disclose their real information. In these cases, it's not only necessary to blur their faces but also to 'blur' their voices, which doesn't mean making the voice sound unclear. Instead, this involves changing the pitch of the voice while ensuring that the voice signal remains unobscured, that is, without altering or losing the information originally carried by the voice signal. The optimal effect would be to transform a male voice into a female voice, and vice versa.

2. Pitch Transformation Principle

The difference in pitch between male and female voices is primarily due to the differing frequencies of their voices [1]. Male voices have lower frequencies, hence a lower pitch; whereas female voices have higher frequencies, leading to a higher pitch. Consequently, male voices sound deeper due to the lower frequency, and female voices sound sharper. Consider the following examples (As shown in Figures 1 and 2 below):

DOI: 10.54254/2755-2721/65/20240515

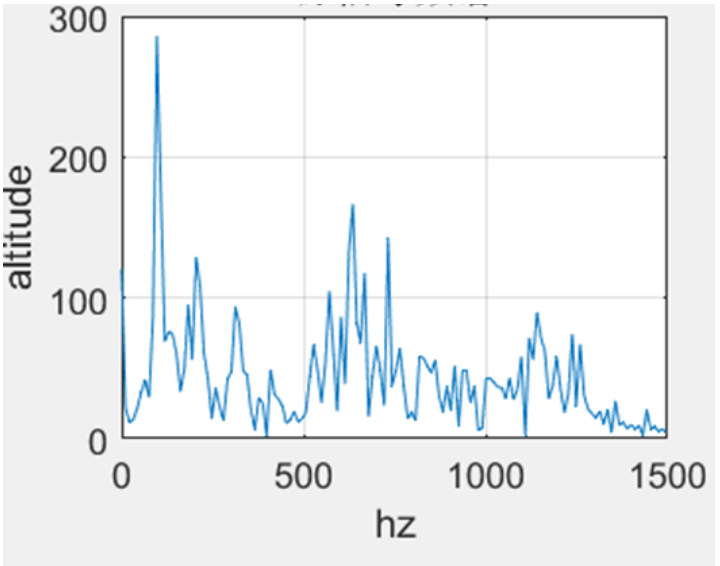

Figure 1. Male Voice Frequency FFT Transformation Spectrum.

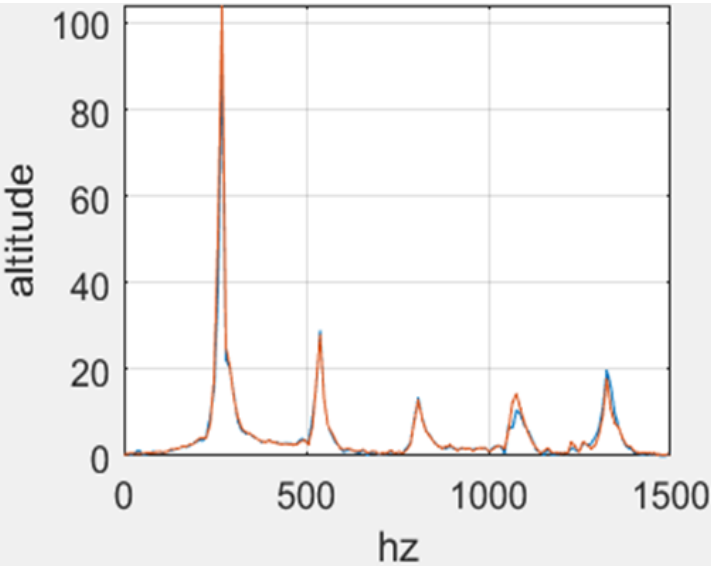

Figure 2. Female Voice Frequency FFT Transformation Spectrum.

Figure 1 shows a spectral analysis of a typical male voice during speech, and Figure 2 shows the same for a typical female voice. It can be observed that the fundamental frequency of the female voice is higher than that of the male. Due to the nature of timbre, to achieve as much voice 'blurring' as possible after transformation, it's necessary to focus on frequencies that most represent male and female voices.

In music theory, the frequency range for tenor male voices is 110~440Hz, with a central frequency point of approximately 275Hz; the standard soprano female frequency range is 246.9~987.4Hz, with a central frequency of 617.35Hz; the standard bass male frequency range is 73.4~293.7Hz, with a central frequency of 183.55Hz; and the standard contralto female frequency range is 164.8~659.2Hz, with a central frequency of 412Hz.

By subtracting the central frequencies of male and female high pitches and then averaging, we get:

$$\Delta f \approx 300\text{hz} \tag{1}$$

DOI: 10.54254/2755-2721/65/20240515

It can be deduced that male and female voices differ by approximately 300Hz in the frequency domain. Therefore, to transform a male voice into a female voice, it is necessary to shift it upwards in the frequency domain by about 300Hz, and the opposite is true for transforming a female voice into a male voice.

3. Spectral Shift Principle

According to the Fourier transform, the spectral density of a signal is represented as

$$F(w) = \int_{-\infty}^{+\infty} f(t) * e^{-jwt} dt \tag{2}$$

which corresponds to specific amplitudes at specific frequencies. If the original signal is multiplied by a complex exponential signal: $s(t)=e^{jw_0 t}$, then the new signal $G(t) = f(t) * s(t) = f(t) * e^{jw_0 t}$ is obtained. Performing the Fourier transform on this new signal:

$$G(w) = \int_{-\infty}^{+\infty} G(t) * e^{-jwt} dt = \int_{-\infty}^{+\infty} f(t) * e^{-j(w-w_0)t} dt \tag{3}$$

The principle of spectral shifting can be derived using the substitution method:

$$F(w - w_0) = G(w) = \int_{-\infty}^{+\infty} f(t) * e^{-j(w-w_0)t} dt \tag{4}$$

The basic principle of spectral shifting is that the original signal is multiplied by a complex exponential signal $e^{jw_0 t}$. According to Euler's formula,

$$e^{jt} = \cos(t) + j * \sin(t) \tag{5}$$

it can be inferred that

$$G(t) = f(t) * \cos(w_0 t) + j * f(t) * \sin(w_0 t) \tag{6}$$

Since imaginary numbers do not exist in practical spectral transformations, the above equation is only mathematically viable. Therefore, taking the real part of Equation 6 yields

$$G_1(t) = f(t) * \cos(w_0 t) \tag{7}$$

Performing the Fourier transform on this signal gives

$$G(t) = f(t) * \cos(w_0 t) + j * f(t) * \sin(w_0 t) \tag{8}$$

Equation 8 indicates that in practical spectral shifts, the amplitude at the target frequency is only half of what it would be in an ideal situation. This means that during the spectral shift, the energy is evenly distributed between two frequency components, and it becomes necessary to filter out the unwanted frequency component to avoid distortion of the voice signal. Afterwards, appropriate amplification is applied to obtain the signal with the transformed pitch.

4. Filter Design

Digital filters are divided into IIR filters (Infinite Impulse Response filters) and FIR filters (Finite Impulse Response filters).

The design methods for IIR filters include direct and indirect methods, with the indirect method being the most commonly used. The indirect method involves utilizing the design methods of corresponding analog filters. Since analog filters have an infinite duration impulse response, IIR filters also have an infinite duration impulse response, thus inheriting the good amplitude-frequency characteristics of analog filters [2]. As the design methods for analog filters are well-established, complete with design formulas, tables, and curves for reference, the process of designing IIR filters is relatively simple. The

DOI: 10.54254/2755-2721/65/20240515

advantage of IIR filters is their ability to achieve high selectivity with a lower order, requiring less data processing and storage resources. However, the drawback is that IIR filters, in their design process, only consider amplitude-frequency characteristics. The designed filters generally have some form of non-linear phase characteristics, which are not deterministic. To achieve linear phase, an additional phase correction network is required, increasing the complexity of design and implementation. Also, the recursive structure of IIR filters can lead to the accumulation and amplification of computational errors, thus imposing high demands on the computational precision of IIR filters.

FIR filters can only be designed directly, and their core principle involves approximating a given frequency response through a finite impulse response. When the impulse response of an FIR filter meets the symmetry condition of $\mathrm{h(n)} = \pm \mathrm{h(N-n-1)}$, it possesses strict linear phase characteristics. Common direct design methods include the window function method, frequency sampling method, and optimization design method [2]. FIR filters are widely used in voice signal processing.

To shift a male voice to a female voice requires an upward frequency shift, but also necessitates filtering out high-frequency noise. Therefore, a band-pass filter with cut-off frequencies f_{l1} and f_{h1} is used. For transforming a female voice to a male voice, a low-pass filter with cut-off frequency f is used. Theoretically, f_{h1} should be greater than the highest frequency of the female voice, and f_{l1} should be greater than the highest frequency of the male voice and less than the lowest frequency of the female voice; f_s should be greater than the highest frequency of the male voice and less than the lowest frequency of the female voice. However, due to the overlapping frequency ranges of male and female voices, this can be modified to male tenor's central frequency $\leq f_{l1} \leq$ female contralto's central frequency, and male tenor's central frequency $\leq f_{h1\leq}$ female contralto's central frequency, with the rest remaining unchanged.

The frequency shift of $\Delta f = 300\hbar z$ requires the filter to have a very narrow transition band, placing high demands on the filter design. Traditional methods of designing FIR filters would result in a high-order filter, leading to large data input into the FPGA and reducing its implementation speed. To avoid this, the Frequency Response Masking (FRM) method is used to design FIR filters.

The FRM method is based on the basic principles of the frequency sampling method. Its fundamental idea is to design a prototype wideband filter, then interpolate the prototype filter by a factor of M, compressing the frequency to achieve M sharp passbands with transition bands 1/M of the prototype filter. Then, a "mask" filter is used to filter out the required passband, ultimately obtaining the desired narrow transition band FIR filter. The following is a brief design process for an FRM filter with M=3 (As shown in Figures 3):

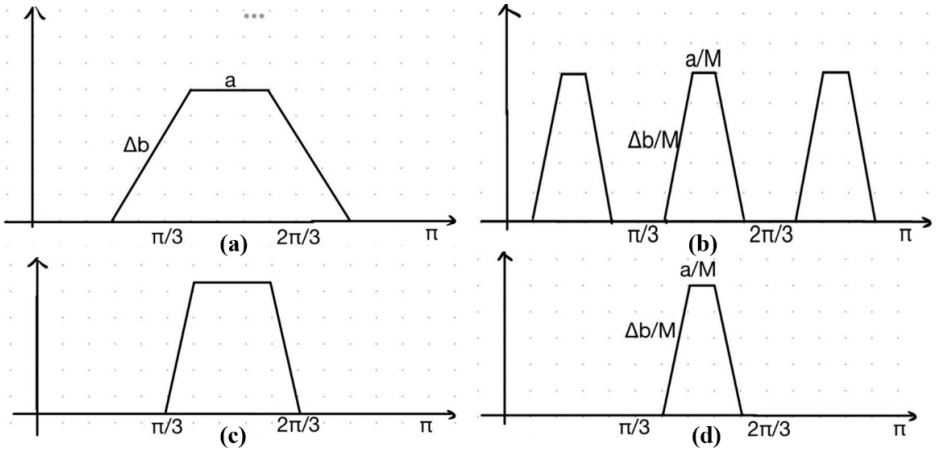

Figure 3. Brief Design Process for an FRM Filter with M=3.

Figure 3(a) shows the designed prototype filter, whose passband should be M times the main frequency range of the audio signal. Let the passband be a, the transition band be Δb, the impulse response be $h_1(t)$, the z-transform be $H_1(z)$, and the frequency response be $H_1\left(e^{jw}\right)$; after performing

DOI: 10.54254/2755-2721/65/20240515

an M-fold interpolation on Figure 2(a), $H_2(z) = H_1(z^M)$. Transforming the z-transform into a frequency response, $H_2(e^{jw}) = H_2(z)_{z=e^{jw}}$, results in the spectral graph shown in Figure 3(b). From Figure 3(b), it can be seen that the spectrum of the prototype filter is compressed by a factor of M, resulting in M sharp passbands, each with its transition and passbands being 1/M of the original, with a passband of a/M and a transition band of Δb/M. Figure 3(c) is the spectral graph of the mask filter. To filter out the target passband through the mask filter, it is necessary for the passband of the mask filter to be larger than the target passband and ensure that the transition band does not overlap with other sharp passbands; Figure 3(d) shows the sharp passband filtered out by the mask filter.

Since the frequency band is compressed, the transition band of the filter, compared to a filter of the same order, is much steeper, being only 1/M of the original. However, the passband is also compressed accordingly. Given that the main frequency band of everyday human speech is relatively narrow, by setting the passband of the prototype filter to be M times the main frequency band, the resulting sharp passband filter will precisely encompass the main frequency band, thus avoiding distortion of the voice signal after passing through the target filter.

From Figures 3(b) and 3(c), it is evident that the interpolation factor M directly affects the complexity of the target filter. When M is set high, the order of the prototype filter can be lower, but correspondingly, there will be more sharp passbands in the frequency domain, totaling M. This increases the difficulty of accurately filtering out the target passband, meaning the mask filter must be of a high enough order to ensure its transition band does not overlap with other passbands. Conversely, when M is low, the compression factor in the frequency domain is smaller, and the difference between the transition band of the target filter and the prototype filter is reduced. Therefore, a higher order of the prototype filter is required to meet the specifications. Thus, choosing an appropriate value of M is also crucial in filter design [2, 3].

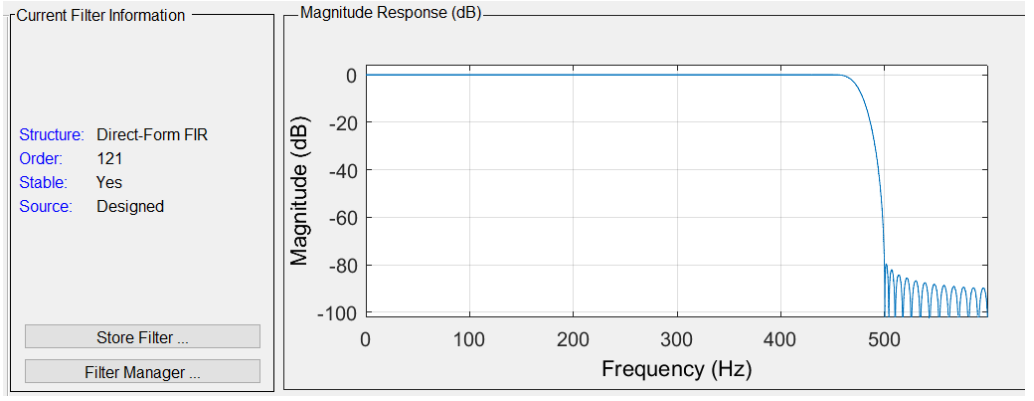

Figure 4. Amplitude-Frequency Characteristics of Directly Designed FIR Filter.

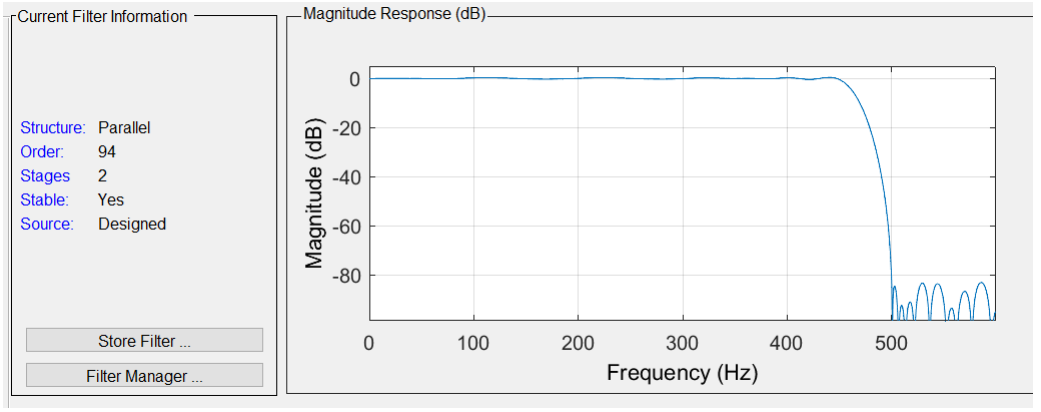

Figure 5. Amplitude-Frequency Characteristics of FRM Filter.

DOI: 10.54254/2755-2721/65/20240515

The above two figures, Figure 4 and Figure 5, illustrate that under the same transition band conditions, the FRM filter has a lower order compared to the FIR filter designed by the direct method.

4.1. FPGA Program Design

The entire program process is divided into three steps: transferring data into the FPGA development board, processing the data on the FPGA development board, and sending the processed data back to the computer.

Data transmission can be achieved through serial ports, WiFi, Bluetooth, etc. For simplicity, this project uses serial port transmission for short-duration audio files.

4.2. Serial Port Program Flowchart

As shown in Figure 6:

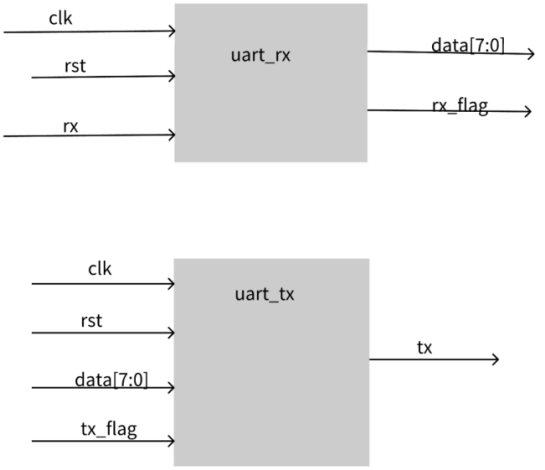

Figure 6. Serial Port Program Flowchart.

The serial port module is divided into receiving and sending parts. Both receiving and sending are completed under the reference of a clock signal. The receiving part inputs serial data rx and outputs 8-bit parallel data data along with a reception completion flag rx_flag. The sending part inputs 8-bit parallel data data and a conversion flag bit tx_flag (determining the starting bit of data conversion). clk and rst are inputs for the clock and reset signals, respectively.

4.3. Filter Design Flowchart

As shown in Figure 7:

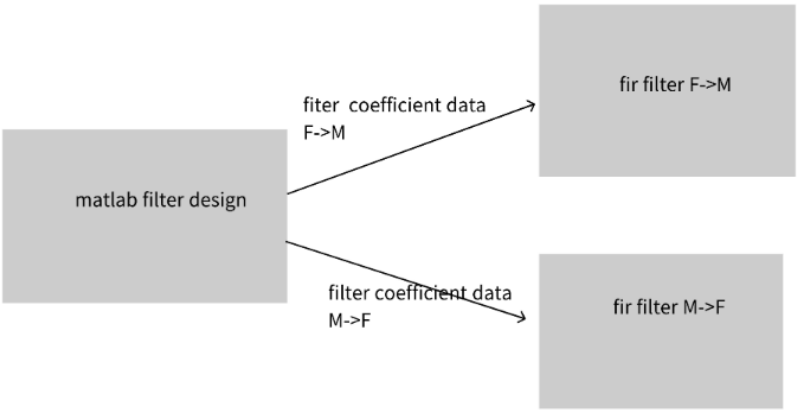

Figure 7. Filter Program Flowchart.

DOI: 10.54254/2755-2721/65/20240515

First, use MATLAB to generate filter coefficients, write the firfilter module, and import the filter coefficients generated by MATLAB. In the above voice analysis, two bandpass filters were used, and it's necessary to instantiate both filters. Since the only difference between the two filters is the cut-off frequency, their internal structures are identical, only differing in the imported filter coefficients. Due to the previously mentioned symmetry property of FIR filters, a symmetric structure is adopted for the filter design [4]. The implementation method involves first summing the contents of registers with identical coefficients and then multiplying the sum by the coefficient, as shown in Figure 8:

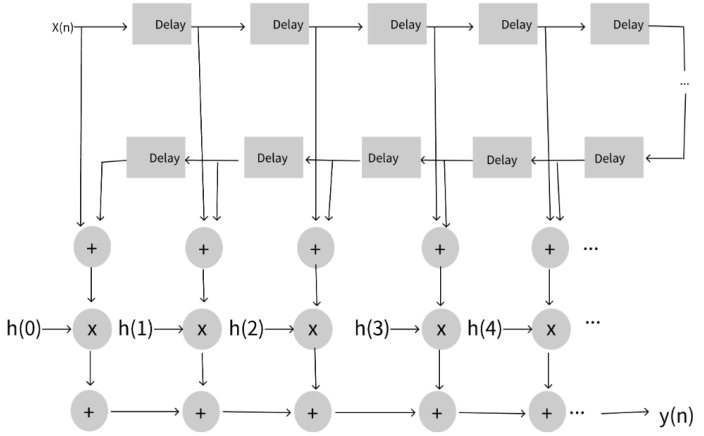

Figure 8. Filter Program Flowchart.

The symmetric structure implementation of the filter can reduce the number of multipliers by half, reducing multiplication operations and increasing speed.

4.4. Overall Program Design

As shown in Figure 9:

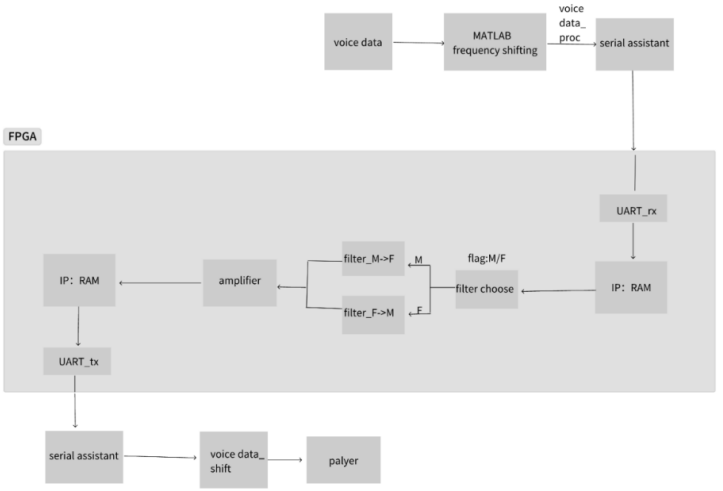

Figure 9. Overall Program Flowchart.

The voice signal is imported into MATLAB in WAV file format, and the audioread function is used to read the file at a sampling rate of 8k. A 4096-point FFT transformation is performed on the data to obtain a spectrum graph and record the waveform. The signal is then frequency shifted by multiplying it with a fixed-frequency cosine signal for mixing. All of the above are processed on the PC side using MATLAB software. The processed signal is sent to the FPGA via a serial port assistant, with a flag added during data transmission for subsequent recognition by the FPGA program. The FPGA receives

DOI: 10.54254/2755-2721/65/20240515

the data through the serial port and uses the IP core: RAM, to temporarily store the received data in RAM for subsequent processing. The received data is evaluated to decide which filter to use. The filter processes the data, and the subsequent amplifier, essentially a multiplier, restores the original amplitude of the signal. After amplification, the data is re-stored in RAM and sent back to the PC via a serial port for playback. MATLAB software is then used to perform FFT analysis on the returned data and observe the spectrum graph.

4.5. MATLAB Simulation to Verify the Feasibility of the Principle

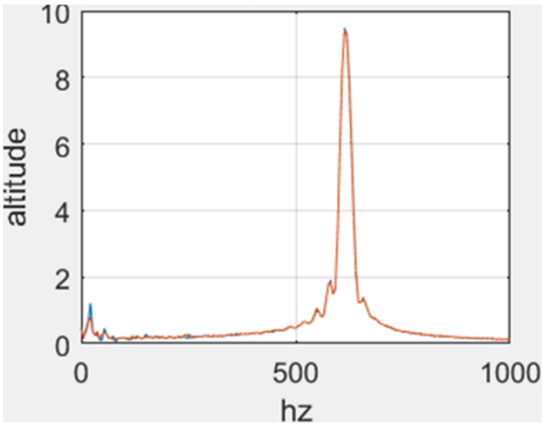

Figure 10. Spectrum of Unprocessed Audio.

The input signal is a high-pitched segment from a female singer's song, sampled at 8k and transformed using a 4096-point FFT. The spectrum graph shows that the frequency of the unprocessed audio signal is around 650Hz, clearly indicating a soprano range. Playing the audio allows one to directly experience the distinctive sharpness of the female voice (as shown in Figure 10).

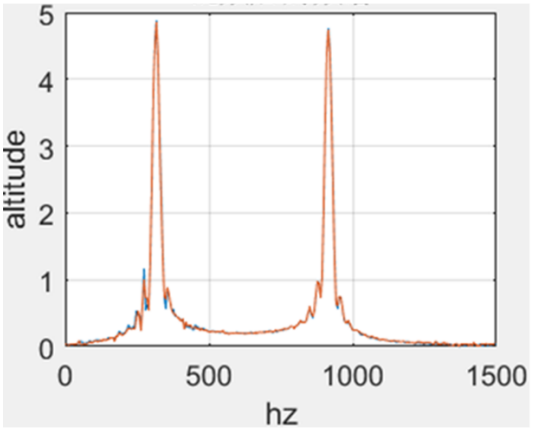

Figure 11. Spectrum After Mixing.

After mixing with a 300Hz sine signal and replaying the audio, both deep and high-pitched sounds are audible. This is due to the presence of both low and high-frequency components after mixing. The spectrum graph reveals two main frequency components of the audio signal—900Hz in the high-frequency part and 350Hz in the low-frequency part, with the low-frequency component being the target frequency. The amplitudes of these two components are approximately half of the original signal's amplitude (as shown in Figure 11).

DOI: 10.54254/2755-2721/65/20240515

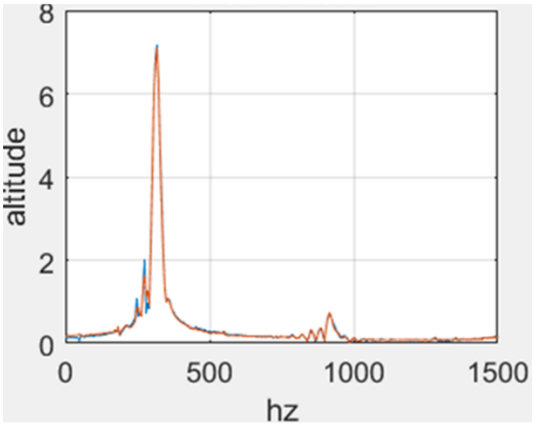

Figure 12. Spectrum After Filtering.

After filtering with an FRM filter, the amplitude of the voice signal is relatively small and needs subsequent amplification. After amplification and audio playback, only the deep voice part remains, closely resembling the pitch of a male voice. Compared to the spectrum after mixing, the 900Hz high-frequency part is significantly suppressed, while the low-frequency part is preserved (as shown in Figure 12).

5. Conclusion

This paper presents a design for pitch transformation of human voice signals based on FPGA and MATLAB. Utilizing FPGA's rapid computation capabilities and MATLAB's signal processing framework, it achieves real-time voice transformation. During the design process, the FRM algorithm and symmetric structure are employed for the filter, which reduces the filter order and the number of multipliers, thereby decreasing computational load and enhancing processing speed. However, there are two unresolved issues in the design: 1) Due to the limited transmission speed of the serial port, for the sake of real-time processing, only short-duration audio files can be transmitted, or longer audio signals must be transmitted in segments, significantly limiting the application scope. 2) Although changing the frequency alters the pitch, it does not change the timbre. The voice does not undergo a complete gender transformation, which in some cases fails to achieve the intended obfuscation. Moving forward, the research will focus on finding more efficient real-time transmission methods and developing better audio processing algorithms to enhance voice transformation effects.

References

[1] S. Xinmei, "Research on MATLAB in Sound Signal Recognition," *Electron. World, 2021*, vol.09, no. 36–37, 2021.
[2] C. Yang, "Research on Design Methods of FRM Filters," 2020.
[3] Stephen, "Design of digital filters in multi carrier communication based on frequency mask method," 2006.
[4] F. B. Qin Zhongyuan, Zhou Lihong, "A method suitable for designing narrow transition band FIR filters," *Electron. Eng.*, vol.11, pp.28–30, 2006.