Frequency Shifting Listening Device

Cheng, Nicholas Peng-Hao (1), Wong, Wei Shern (1), Cazzolato, Benjamin (1), Prime, Zebb (1), Hewett, Keith (2) and Burgemeister, Kym (3)

(1) School of Mechanical Engineering, The University of Adelaide, Adelaide, Australia
(2) ARUP, Adelaide, Australia
(3) ARUP, Melbourne, Australia

ABSTRACT

Sound identification is important in vibration and acoustic related engineering fields. The frequency spectrum of sound in some environments falls into the infrasound frequency range below 20 Hz, which is too low for the average human ear. Furthermore, sound above this frequency is still often difficult to identify due to interference and masking by other ambient noise sources. The focus of this paper is on the design and construction of a frequency shifting listening device which is able to shift frequency in real time and amplify low frequency noise into the audible frequency range, typically around 20 Hz to 20 kHz. The design phase of this device includes programming in Matlab/Simulink and rapid prototyping hardware using a Texas Instruments (TI) TMS320C6713 Digital Signal Processor Starter Kit (DSK).

1. INTRODUCTION

The average healthy human ear is typically responsive to noise which is composed of frequencies ranging from 20 Hz to 20 kHz (Bies & Hansen 2009, p. 76). Frequencies below 20 Hz are classified as infrasound, whereas frequencies greater than the upper limit of human hearing, which is above 20 kHz, are classified as ultrasound. By means of frequency shifting, noise in the infrasound region can be altered to be in the audible range of frequencies. Frequency shifting is a widely used application especially in the music industry and is often associated with increasing or decreasing the pitch of a music recording for harmonisation of instruments or vocals within a song (Demers, Grondin & Vakili 2009). An example of this application is a guitar pitch shifter which produces sound at a different pitch to that played on the guitar.

The function of the proposed frequency shifting listening device is to enable users to listen and identify low frequency noise. Identification of low frequency vibration/noise can be helpful for acoustic or vibration related problems; for instance, appropriate measures can be taken to mitigate the effects of annoyance caused by low frequency noise in residential areas if the source of this is identified. This cannot be directly determined by merely recording the noise and playing it back. Hence, engineering companies such as Arup has an interest in acquiring a device which allows them to listen to low frequency noise clearly in real time and help identify the noise source.

2. FREQUENCY SHIFTING LISTENING DEVICE

The concept of the frequency shifting listening device is relatively straightforward. An input audio signal is fed through the device, which consists of a digital signal processor (DSP), and then performs calculations and outputs the resulting signal which could be heard by connecting to a set of speakers or headphones. Besides frequency shifting, the device would have other capabilities including filtering, integrating and differentiating signals.

2.1 Specifications

The proposed frequency shifting listening device has the following features, which includes:

- Selectable low and high pass filters. These are essential to remove masking by background or ambient noise.
- Portability. The frequency shifting listening device is intended to assist engineers on site. Hence the device should be lightweight and convenient to be carried around. It is proposed that device should weigh no more than 2 kg and not exceed a physical dimension of 300 × 200 × 100 mm.
- Power supply with rechargeable batteries. A separate DC jack would be incorporated in the case that the batteries are exhausted and a direct power supply is available.
- I/O jacks. The listening device would have 2 input jacks (for line in and microphone connections) and 2 output jacks (for line out and headset connections). This comprises of 4 channels in the system.
- Selectable gains. The gains of the signal would be changed in discrete steps.
- A single differentiator, double differentiator, single integrator and double integrator. Switching between a double differentiator to double integrator goes from +40 dB/decade to -40 dB/decade of filtering.

3. SIMULINK, CCS AND TI C6713 DSK

A prototype of the proposed frequency shifting listening device was built using a Texas Instruments (TI) TMS320C6713 Digital Signal Processor Starter Kit (DSK). The C6713 DSK was programmed using Simulink and TI's Code Composer Studio version 3.3. Figure 1 shows a flow diagram connecting Simulink with the C6713 DSK. A model is first built in Simulink, run and tested. It can then be exported as C code via Simulink's Embedded Coder. This C program is then compiled, linked, and downloaded into Code Composer Studio and executed. Error messages would be displayed if they are present, however a successful run would be followed by generation of an executable file which would be loaded unto the C6713 DSK.

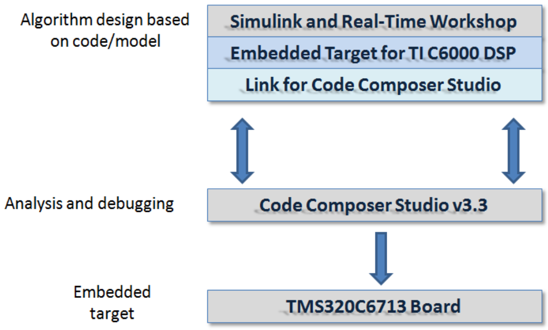

Figure 1. Targeting C6713 DSK using Simulink

The C6713 DSK was used to implement the function of the listening device in real time as it is a relatively low cost standalone development platform (approximately $400) with adequate capability. It features a TMS320C6713 DSP chip operating at 225 MHz and delivering up to 1800 million instructions per second (MIPS) and 1350 million floating-point instructions per second (MFLOPS). The C6713 DSK also features a codec (TLV320AIC23) which essentially functions both as an Analogue-to-Digital (A/D) converter and a Digital-to-Analogue (D/A) converter. In addition, the board has two inputs (mic in and line in) and two outputs (headphone and line out). Figure 2 shows the configuration of the C6713 DSK hardware.

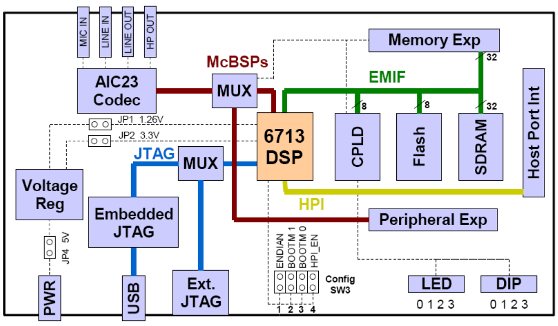

Figure 2. Texas Instruments C6713 DSK hardware (Spectrum Digital 2003)

4. FREQUENCY SHIFTING ALGORITHM

To be able to shift frequencies of a particular signal, a mathematical algorithm must be constructed. The theory behind frequency shifting will be introduced in this section which provides a basis for this algorithm. All of the work covered in this section, which includes derivation of equations and algorithms, was undertaken and documented by Demers, Grondin and Vakili (2009), all of whom successfully completed their project on a real time pitch shifter for an electric guitar. All credit for this algorithm goes to them.

4.1 Theory

The pitch of a sound is associated with the set of frequencies that the sound is composed of. Hence the term "pitch shifting" is often equivalent to "frequency shifting". The difference before and after a sound is pitch shifted becomes obvious when there is a significant increase (or decrease) in its pitch (or frequency). For example, the singing of a man would sound very much akin to the singing of a woman if his pitch is greatly increased. Another example which would completely illustrate the theory behind pitch shifting is by looking at musical instruments and their signature sounds, as seen below.

4.1.1 Music and timbre

The musical range is divided into many octaves, where each octave is made of twelve semitones, also referred to as half steps. Each semitone corresponds to a specific note. A pure note is made of a single sinusoid at the fundamental frequency. It is obvious that the same note produced by different musical instruments do not have the same perceived sound. Otherwise it would be impossible to distinguish a note "C" played using a guitar with a note "C" played by a piano. This is because each note (which is analogous to a signal) is usually composed of a fundamental frequency and a set of harmonics. Harmonics consist of frequencies which are integer number multiples of the fundamental frequency. The organization of harmonics with respect to each other allows one to recognize the musical instrument being played. This identity is called the timbre of an instrument.

4.1.2 Frequency scaling

Each musical note corresponds to a fundamental frequency. Demers, Grondin and Vakili (2009) express the relationship between semitones and fundamental frequency as

$$p = 69 + 12\log\left(\frac{f}{440}\right), \qquad (1)$$

where p is the number of semitones and f is the frequency in Hertz. As mentioned earlier, frequency shifting in its context is equivalent to pitch shifting. From a musical perspective, pitch shifting involves shifting a melody by octaves or semitones. This can be observed in

$$p_{final} = p_{initial} + s, \qquad (2)$$

where p_{final} is the final semitone index, $p_{initial}$ is the initial semitone index and the number of semitones for shifting is s. From a signal perspective, Equation (2) is expressed similarly as a scaling of the fundamental frequency and harmonics by a specific factor,

$$f_{final} = 2^{(s/12)} f_{initial}. \qquad (3)$$

4.2 Approach

The approach taken for pitch shifting is to reproduce a signal at a faster speed. Suppose a song is recorded on tape and played back twice as fast. It is expected that the song played back will have a pitch twice higher than previously recorded; something like a "chipmunk" voice. Figure 3 illustrates the signal played at twice the speed.

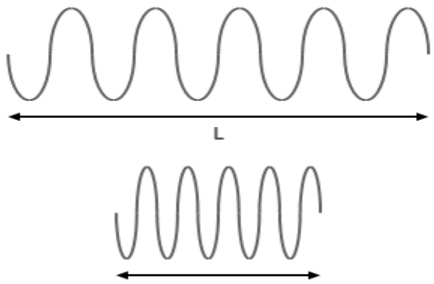

Figure 3. A recording signal played at twice its speed (Demers, Grondin & Vakili 2009)

It can be observed from Figure 3 that the frequency of the signal has increased by a factor of two. From a musical perspective, this means that the pitch of the song has been shifted up by an octave. However, it is important to note that the signal is now half the duration than before, which is undesirable. Nonetheless, the approach above would work if the signal is first doubled in length without affecting the pitch. When the altered signal is played back twice as fast, the frequency will be doubled and its duration would be the same as initially. In most cases, the algorithm used to frequency shift a signal is based on this principle and will be further explained in the following section.

4.3 Algorithm

The algorithm derived in this section is based on the approach described in Section 4.2. Firstly, it is required to change the signal duration without affecting its frequency (or pitch). This can be achieved by using a scaling factor to stretch or compress the spectrum of a signal. Secondly, it is required to frequency shift the signal such that its duration returns to its original length. This is analogous to playing a recording at a faster speed. To illustrate the mentioned approach, suppose it is required to shift the pitch of a signal by one semitone. This would require a scaling factor of $2^{1/12}$, or equivalently 1.0594, which effectively stretches the signal without affecting its pitch. Consequently the signal needs to be replayed 1.0594 faster to successfully shift its pitch by a semitone. The following subsections further explain mathematical algorithms used to realise this approach. The process begins by stretching or compressing a signal by the superposition of frames. Analysis, processing and synthesis of the stretched (or compressed) signal by using a phase vocoder algorithm are necessary to ensure this process is completed smoothly. The signal is then re-sampled to obtain a shift in frequency while returning to its initial duration.

4.3.1 Superposition of frames

In order to stretch or compress a signal by a scaling factor, the signal is first divided into many segments, or 'frames' as shown in Figure 4. The frames taken from the signal are such that they overlap each other by a certain percentage, typically 75 percent. The scaling of the signal is then accomplished by placing these frames at a different distance relative to each other.

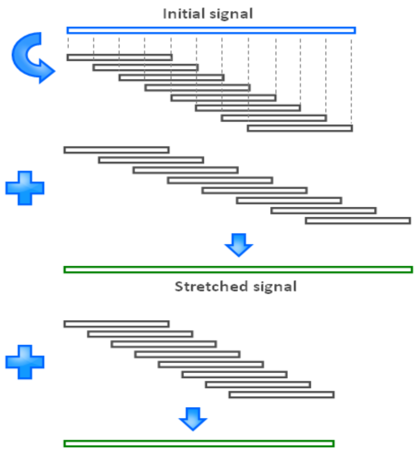

Figure 4. Building a scaled signal by frames (Demers, Grondin & Vakili 2009)

A problem arises once the frames of the signal are placed accordingly. Discontinuities occur along the altered signal which would result in glitches when the signal is played. However, this issue is resolved using a phase vocoder which effectively scales the signal without any discontinuity.

4.3.2 Phase vocoder

The phase vocoder algorithm essentially accomplishes the time scaling (i.e. stretching or compressing) of a signal through the use of Fourier techniques. It is composed of three stages as shown in Figure 5; analysis, processing and synthesis.

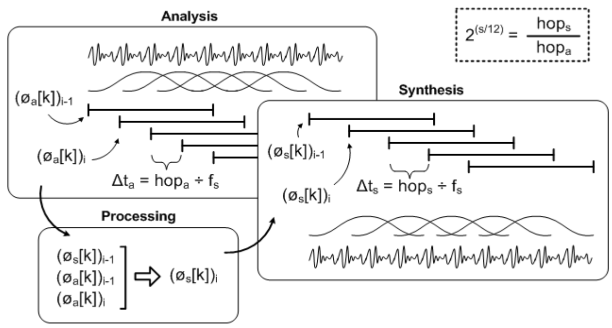

Figure 5. Phase vocoder algorithm (Demers, Grondin & Vakili 2009)

In the analysis stage, frames are constructed using a method known as windowing. A window function is a mathematical function which has zero values outside a certain defined interval. Hence, windowing refers to obtaining a small segment from a signal. Since windowing can cause changes to the frequency spectrum of a signal, a Hanning window of size N is used to reduce this effect. This is justified as the Hanning window contains most of its energy at its centre than its side lobes hence reproducing the signal as faithfully as possible, as shown in Figure 6. The frame which consists of N samples is then transformed from the discrete time domain to the frequency domain by means of the Fast Fourier Transform (FFT) using

$$(X_a[k])_i = \sum_{n=0}^{N-1} x[n + i \times hop_a] w[n]\, e^{-j(2\pi kn)/N} \qquad (4)$$

for which k = 0,1,2,3,...., N-1, where $x[n]$ is the sampled signal, $w[n]$ is the Hanning window function and $X_a[k]$ is the discrete spectrum of the i-th frame. In addition, hop_a refers to the hop size, or number of samples, between two successive windows and is equal to *N/4* for an overlap of 75 percent. This overlap percentage is chosen to increase the resolution of the spectrum.

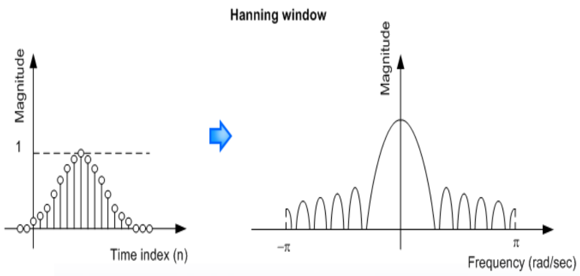

Figure 6. Hanning window (Demers, Grondin & Vakili 2009)

In the processing stage, applying a FFT to the frames results in N frequency bins starting from 0 to $(N-1)/N\ f_s$ with intervals of f_s/N, where f_s is the sampling frequency. The phase information of a signal is important to improve the accuracy of the frequency estimation of each bin. The phase difference between two successive frames is termed the phase shift. Figure 7 depicts an example of two signals with different frequencies. The first signal has no phase difference as its frequency matches into the first bin frequency. However, the second signal, which has a greater frequency than the first, corresponds to a phase greater than zero in the first bin.

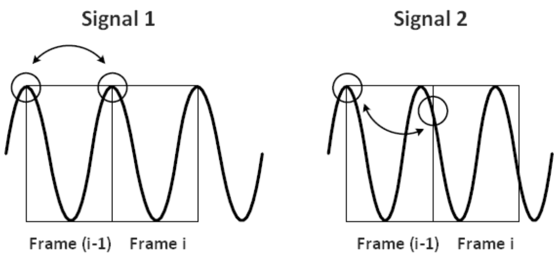

Figure 7. Two signals with different frequencies and phase difference (Demers, Grondin & Vakili 2009)

The phase shift can be utilised to calculate the true frequency associated with a bin. The equation for this is

$$(\omega_{true}[k])_i = \frac{(\Delta\varphi_a[k])_i}{\Delta t_a} \quad (5)$$

where k and i denotes the bin and frame index respectively, $(\Delta\varphi_a[k])$ is the phase shift, and Δt_a is the time interval between two frames which is equal to hop_a/f_s. However, the phase information provided by the FFT is bounded, which implies that the phase shift lies between –π and π. Therefore, to calculate the true frequency within the frame, the frequency deviation from the bin is first calculated, bounded and then summed with the bin frequency. The following equations follow this procedure, where Equation (8) calculates the true frequency by initially obtaining the frequency deviation as given in Equation (6) and then bounded using Equation (7). The variables $(\varphi_a[k])_i$ and $(\varphi_a[k])_{i-1}$ are the phase of the current and previous frame respectively, whereas $\omega_{bin}[k]$, $(\Delta\omega[k])_i$ and $(\Delta\omega_{bounded}[k])_i$ are the bin frequency, frequency deviation and bounded frequency deviation respectively.

$$(\Delta\omega[k])_i = \frac{(\varphi_a[k])_i - (\varphi_a[k])_{i-1}}{\Delta t_a} - \omega_{bin}[k] \quad (6)$$

$$(\Delta\omega_{bounded}[k])_i = mod[((\Delta\omega[k])_i + \pi), 2\pi] - \pi \quad (7)$$

$$(\omega_{true}[k])_i = \omega_{bin}[k] + (\Delta\omega_{bounded}[k])_i \quad (8)$$

Furthermore, the new phase of each bin is obtained by summing the phase shift required to ensure signal continuity. This is equal to the product of the true frequency ω_{true} with the time interval of the synthesis stage Δt_s, defined by hop_s/f_s, as expressed in Equation (9). In addition, Equation (10) shows the amplitude and phase of the new spectrum.

$$(\varphi_s[k])_i = (\varphi_s[k])_{i-1} + \Delta t_s \times (\omega_{true}[k])_i \quad (9)$$

$$|(X_s[k])_i| = |(X_a[k])_i|, \angle(X_s[k])_i = (\varphi_i)_s \quad (10)$$

During the synthesis stage, it is desired to return to the time domain after adjusting the phase in the frequency domain. This is accomplished by applying the Inverse Discrete Fourier Transform on each frame. A Hanning window is then used to smooth the signal and obtain the windowed samples $q_i[n]$, where n progresses from 0 to N-1. Lastly, each frame is superimposed to obtain the desired output $y[n]$ as shown in Equation (12), where L is the number of frames and $u[n]$ is the unit step function.

$$q_i[n] = \left\{\frac{1}{N}\sum_{k=0}^{N-1}(X_s[k])_i e^{-j(\frac{2\pi kn}{N})}\right\} w[n] \quad (11)$$

$$y[n] = \sum_{i=0}^{L-1} q_i[n - i \times hop_s]\{u[n - i \times hop_s] - u[n - i \times hop_s - N]\} \quad (12)$$

4.3.3 Resampling

From the previous calculations, a time scaled signal is obtained without any change to the frequencies. The signal then needs to be re-sampled in order to return to its initial duration, while effectively frequency shifting. For example, shifting frequencies by a factor of 2 would require one sample to be taken out of every two to output the desired result. This process is only true if the scaling factor is an integer. A non-integer scaling factor would require the time scaled signal to be re-sampled by means of interpolation. However, the model currently built for the frequency shifting listening device only focuses at shifting frequencies by a factor of 2, 3 and 4.

4.4 Implementation in Simulink

The frequency shifting algorithm was built as a model and implemented in Simulink. A 1000 Hz sine wave of amplitude 50 was used as an input to test the effectiveness of the model. The output was analysed using a Spectrum Scope block. Figure 8 shows the FFT spectrum of the 1000 Hz tone generated with a peak of 9 dB whereas Figures 9, 10 and 11 show the FFT spectrums for double, triple and quadruple times the original frequency respectively. These were obtained with a sampling frequency of 48 kHz, a frame of 1024 samples, a 1024 point FFT and an initial overlap of 768 samples, which is a 75 percent overlap of each frame.

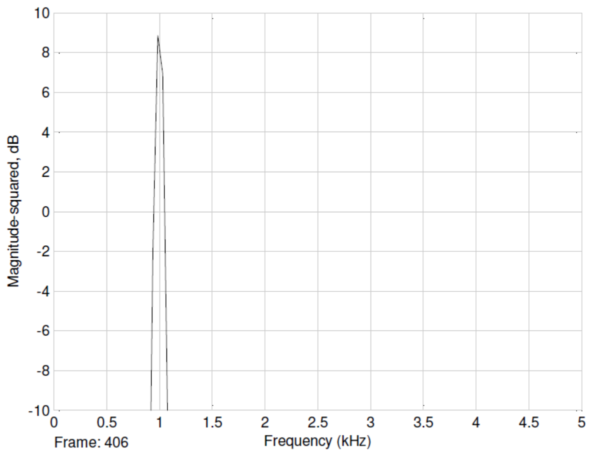

Figure 8. FFT spectrum of generated 1000 Hz tone

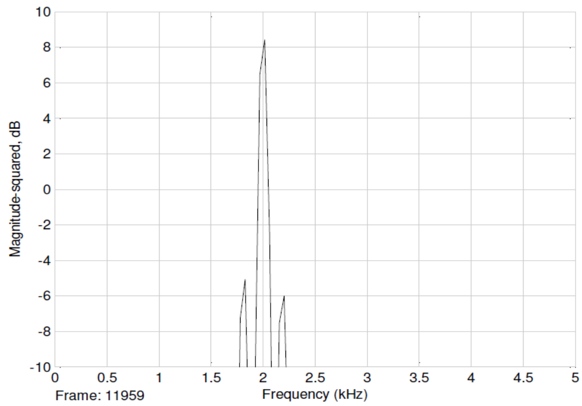

Figure 9. FFT spectrum after frequency shifting by a factor of 2

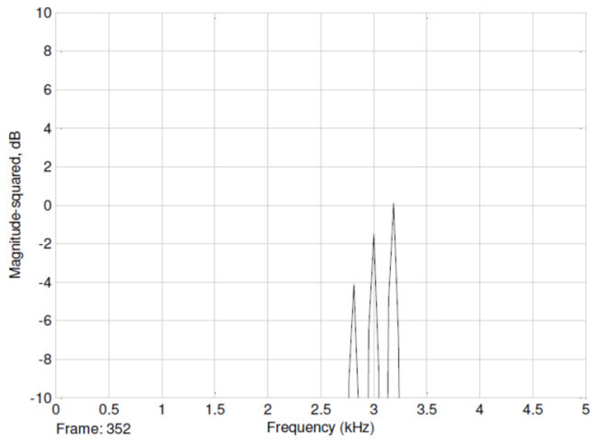

Figure 10. FFT spectrum after frequency shifting by a factor of 3

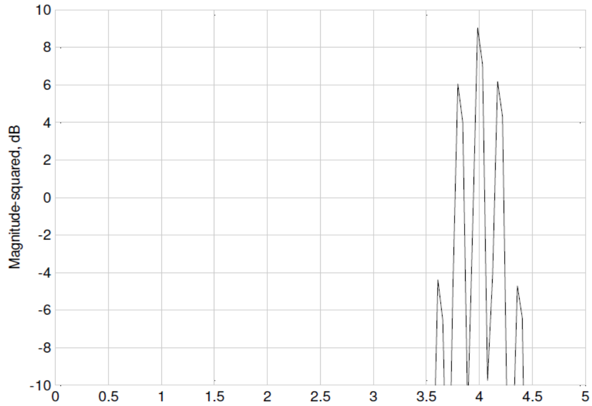

Figure 11. FFT spectrum after frequency shifting by a factor of 4

It was expected that the FFT spectrums would show a spike at 2000 Hz, 3000 Hz and 4000 Hz respectively. The figures show that these were present with several side bands and reduction in peak magnitude. The difference in frequency, Δf, of the side bands comes from the sample frequency divided by the number of samples between each frame, or rather expressed as

$$\Delta f = \frac{f_s}{hop_a} \qquad (13)$$

where f_s is the sampling rate and hop_a is the number of samples between two successive frames. One possible method to reduce this problem would be to increase the overlap between frames. This would increase the ability to estimate the true frequency by making Δf of the sidebands larger but with smaller magnitude. Figures 12, 13 and 14 show the effects of increasing overlap percentage between successive frames. These FFT spectrums were obtained using the same sampling frequency of 48 kHz, a frame of 1024 samples and a 1024 point FFT, but with an increased overlap of 93.75 percent. This gives an initial overlap of 960 samples and a hop_a size of 64 samples. From the figures, it can be seen that a clear spike at 2000 Hz, 3000 Hz and 4000 Hz is obtained. This also implies that the sidebands have negligible magnitude which does not exceed more than -10 dB, which is the lowest limit of the FFT spectrums shown.

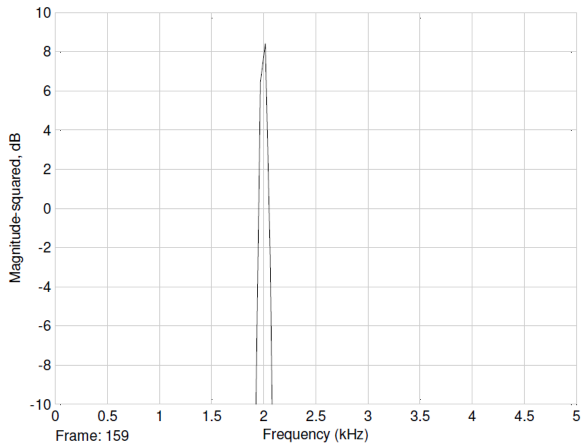

Figure 12. FFT spectrum after frequency shifting by a factor of 2 with increased overlap

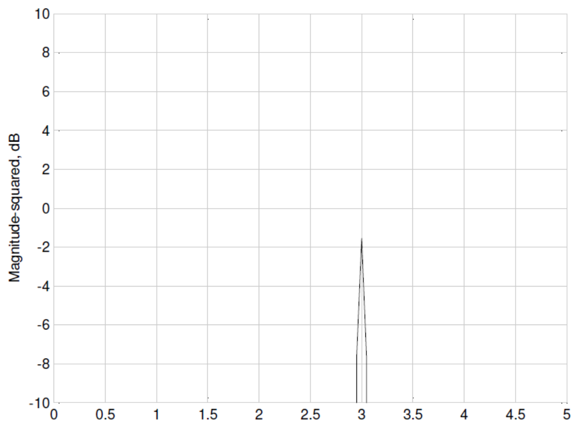

Figure 13. FFT spectrum after frequency shifting by a factor of 3 with increased overlap

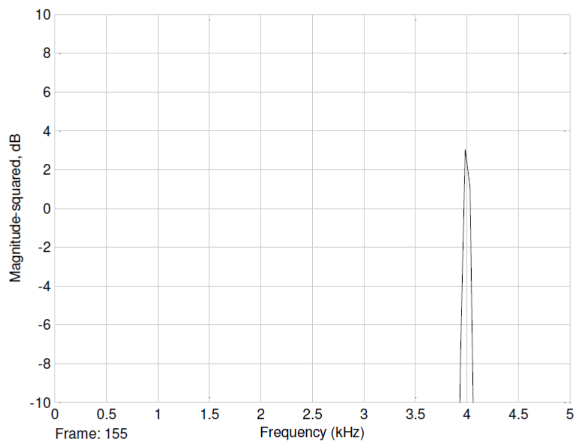

Figure 14. FFT spectrum after frequency shifting by a factor of 4 with increased overlap

5. FILTERS

The main objective of filters is to allow certain frequencies of a signal to pass through a system while attenuating others, thus giving emphasis or de-emphasis to sections of a signal based on its frequency content. Filters are commonly used to ameliorate the quality of a signal, extract information from sections of a signal or separate signals into a number of components to, for example, efficiently use an available communication channel (Ifeachor & Jervis 2002, p. 318). In conjunction with these objectives, Smith (1997) assigns two areas where filters are mainly used; signal restoration and signal separation. For the purpose of the frequency shifting listening device, filters will be used for signal restoration, specifically to improve the sound quality of an audio signal by removing or reducing unwanted noise contained within it.

5.1 Selection of an appropriate filter

Since the form of signal operated in the device is discretized, it is only appropriate to use digital filters. The digital filters available for implementation are the Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters. In this project IIR filters were chosen in preference to FIR filters. The main reason for this is because IIR filters generally have lower orders than FIR filters with similar performance characteristics (Hussain, Sadik & O'Shea 2011, p. 108). With lower orders, IIR filters tend to require less computational effort due to fewer delay elements, digital multipliers and adders for hardware implementation and fewer calculations in software implementations. This is very important as the frequency shifting listening device requires short computational time for real time implementation.

Unlike FIR filters, IIR digital filters are recursive; meaning that present outputs are not only dependent on present and past inputs, but past outputs as well. In some sense an IIR filter is a feedback system and is characterised by the following recursive equation,

$$y(n) = \sum_{k=0}^{N} b_k x(n-k) - \sum_{k=1}^{M} a_k y(n-k) \qquad (14)$$

where b_k and a_k are the coefficients of the filter, and $x(n)$ and $y(n)$ are the input and output to the filter respectively. The transfer function which defines the IIR filter is simply the z-domain transform of the equation above, giving

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k z^{-k}} \qquad (15)$$

where b_k and a_k are the coefficients of the filter and z^{-k} represents a k-th time delay element. The important part of the IIR filter design process is finding suitable values for the coefficients b_k and a_k such that the filter meets certain specifications. The method used to calculate these coefficients is the bilinear z-transform. Firstly, for a given set of filter specifications, a transfer function is derived from classical prototype low pass analogue filters (normalised with critical frequency at 1 rad/s) such as Butterworth, Chebyshev or Elliptic filters. By frequency transformation, this prototype low pass filter transfer function is then de-normalised and converted to the desired type of filter (e.g. low pass, band pass, band stop) which satisfies the filter specifications. This analogue transfer function is then mapped to the discrete time domain by means of bilinear transform and filter coefficients are calculated. Essentially, IIR filters simulate the desired characteristic behaviour of classical analogue filters. Among the four common analogue filters, the Elliptic filter is chosen as the main design of the IIR filter for two reasons. Firstly, an Elliptic filter yields the lowest filter order for a given set of specifications, hence reducing computational time and effort for real time implementation. Secondly, an Elliptic filter has the best roll off in its transition band and therefore efficiently attenuates unwanted frequencies. Although it may exhibit ripples in the pass band and stop band as well as poor phase response, these disadvantages do not significantly affect the sound quality of the noise output from the filter.

5.2 Implementation of filters

Six selectable low and high pass filters were implemented in the C6713 DSK. These consist of three low pass filters with cut-off frequencies of 250 Hz, 500 Hz and 1000 Hz, as well as three high pass filters with cut-off frequencies of 1 Hz, 2 Hz and 4 Hz. These cut-off frequencies were chosen based on octave band frequencies. More filters could be employed; however this would have an impact on memory storage when implemented on the C6713 DSK. In addition, these filters were implemented as 2nd order Elliptic filters in second order sections (direct form biquad), as shown in Figure 15.

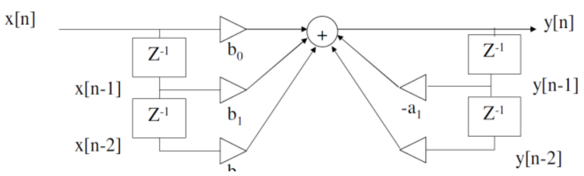

Figure 15. Direct form biquad filter structure (*Digital Filters* 2005)

In Simulink, the design and implementation of these filters were simplified using the Digital Filter Design block available. By selecting the block, a Graphical User Interface (GUI) appears which allows user defined filter specifications and displays the frequency response of the designed filter. Figures 16 and 17 show an example of this.

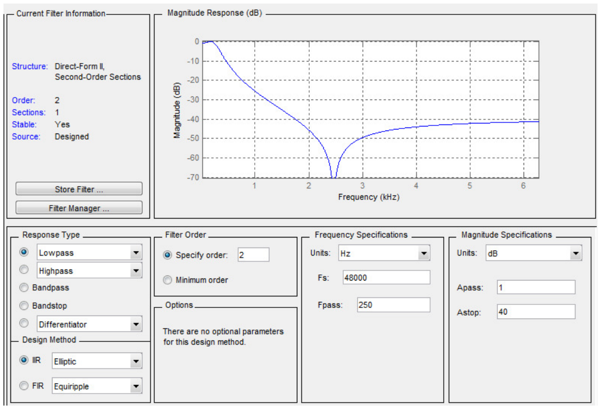

Figure 16. Digital Filter Design block GUI

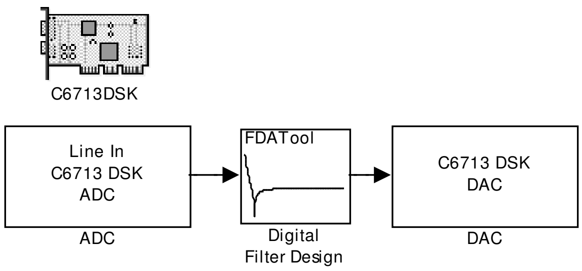

Figure 17. Implementation of a low pass filter in Simulink

6. INTEGRATORS AND DIFFERENTIATORS

The purpose of including integrators and differentiators in the frequency shifting listening device is to change the frequency weighting of signals. It was chosen for the device to have five options on this; a single integrator, double integrator, single differentiator, double differentiator and passing the signal straight through. Since a pole and zero would change the magnitude of a signal by -20 dB/decade and +20 dB/decade respectively, switching between a double integrator to double differentiator goes from -40 dB/decade to +40 dB/decade of filtering. An integrator and differentiator is generally defined by their transfer functions as shown in Equations (16) and (17) respectively.

$$H(s) = \frac{1}{s} \tag{16}$$

$$H(s) = s \tag{17}$$

6.1 Implementation of integrators and differentiators

The single integrator and differentiator are implemented as first order low pass and high pass filters respectively. In the Laplace domain, the transfer function of a first order low pass filter with a cut-off frequency of f_c is

$$H(s) = \frac{\omega_c}{s+\omega_c} = \frac{1}{1+s/\omega_c} \tag{18}$$

where ω_c is $2\pi f_c$. Similarly, the transfer function in the Laplace domain of a first order high pass filter is

$$H(s) = \frac{s}{s+\omega_c} = \frac{s/\omega_c}{1+s/\omega_c} \tag{19}$$

where ω_c is $2\pi f_c$ and f_c is the cut-off frequency. The cut-off frequencies for the integrator and differentiator were taken to be 1 Hz and 200 Hz respectively. The double integrator and differentiator would simply be the square of the transfer functions defined above. Before implementing the integrators and differentiators on the C6713 DSK, a Matlab script was written to generate a discrete transfer function in the z-domain by specifying a cut-off frequency and sampling time. The transformation of the transfer function from the Laplace domain to discrete domain was accomplished by using the Matlab command *c2d*. By default, the *c2d* command computes the discrete transfer function by using zero order hold on inputs. Once the transfer function in the z-domain is obtained, the coefficient values are input to the Simulink Discrete Transfer Function block and used in the Simulink model. An example of this is shown in Figure 18.

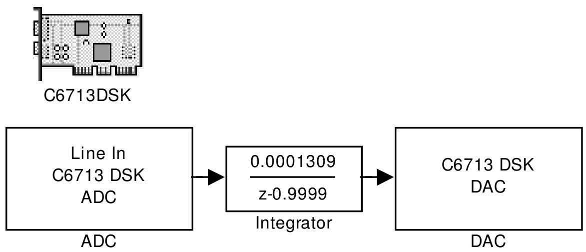

Figure 18. Implementation of a single integrator in Simulink

7. HARDWARE

To meet the design specifications of a portable device, the DSK will be contained within a plastic instrument enclosure which includes a LCD and a portable power supply. This hardware will then be integrated together to allow an interactive functional control of the device.

7.1 Liquid Crystal Display (LCD)

A 20 by 4 alphanumeric LCD will be interfaced with the DSK via a two layer prototyping daughter card which is connected to J4, memory expansion connector. The DSP chip interfaces with the LCD using 32-bit External Memory InferFace (EMIF) signals, controlled by an Altera EMP3128TC100-10 Complex Programmable Logic Device (CPLD) device. The CPLD device is programmed via J8, a dedicated JTAG interface on the DSK, with source files written in industry standard VDHL (hardware design language).

The CPLD device controls the daughter card using two of its memory mapped registers. DC_REG is used to monitor and control the daughter card interface where as MISC_REG controls the auxiliary signals that are brought out to the daughter card connectors.

7.2 POWER SUPPLY

The DSK requires a 5V power supply to allow it to function. From this supply voltage, +1.26V is utilised for the DSP core while +3.3V is used for input and ouput buffers and the other chips on the DSK. In addtion to this, the LCD requires +5V to function. To meet this functional requirement, the device will be powered by eight AA batteries connected in series to provide a total supply voltage of 12 volts. Before energizing the board, the power supply is regulated to a constant supply of 5V using an IC LM7805. The circuit design of the voltage regulator is as shown in Figure 19. In addition, a voltage level indicator using ten LEDs is incorporated to show the remaining voltage supply available. The voltage level indicator circuitry is based on an IC LM3914 and is shown in Figure 20. A DC power jack is used to complete the circuit from the regulated voltage supply to the DSK. Upon depleting voltage supplied by the batteries, the DSK can be connected to a main supply via J5 connector. The J5 connector is a female plug with an outer diameter of 5.5 mm and an inner diameter of 2.5 mm. Upon mating the female barrel plug, the DSK will be powered by the mains and the battery supply cut off. Memory connector provides power and ground to daughter card so that daughter card can be powered directly from DSK.

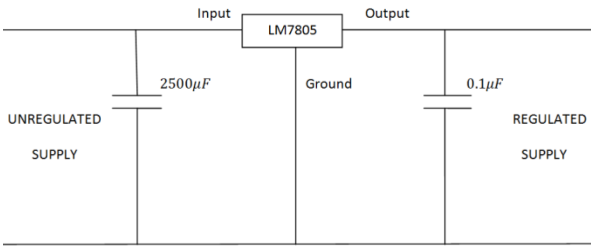

Figure 19. Voltage regulator circuit

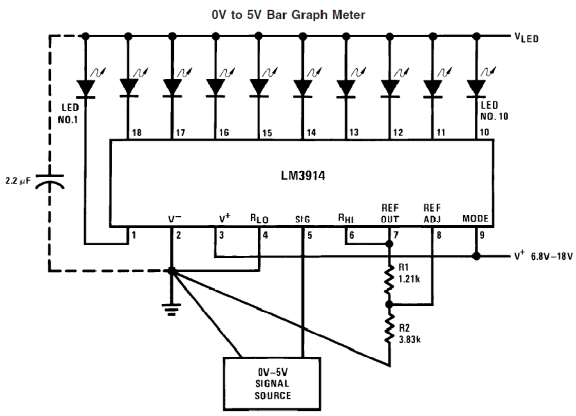

Figure 20. Voltage level indicator circuit (National Semiconductor 2003)

8. DISCUSSION

The complete model was successfully implemented in Simulink. However, several problems emerged when the model was executed on the C6713 DSK. One of the major problems was that no sound was heard on the headphones when the frequency shift subsystem model was included. At the time of writing, the authors are still investigating this cause and are trying to successfully implement the whole model on the C6713 DSK. However, a C code written by Demers, Grondin and Vakili (2009) is available which is able to frequency shift an input signal while preserving its quality well up to 4 semitones. The authors hope to resolve the problem of implementing the complete Simulink model on the C6713 DSK in the near future.

9. CONCLUSION

The available algorithm proposed by Demers, Grondin and Vakili (2009) is effective and has been successful in pitch shifting without any compromise in signal quality for up to a factor of 4. Using this approach, a device which is able to shift frequencies of a signal to its audible range was shown to be possible. A Simulink model was built to simulate the functions of the envisaged frequency shifting listening device. However, the model was unable to run on the C6713 DSK due to unknown problems. Work is still continuing to successfully execute the model on the digital signal processor. Future work which can be done to improve this device would be the integration of a system which is able to correlate the signal with a noise source, similar to a bird call identifier.

REFERENCES

Bies, D & Hansen, C 2009, *Engineering Noise Control: Theory and Practice*, 4th edn, Spon Press, Abingdon, Oxfordshire.

Demers, L, Grondin, F, & Vakili, A 2009, 'Real-time pitch shifter for an electric guitar', School of Electrical, Computer & Software Engineering, McGill University, Montreal, Quebec, Canada.

Digital Filters 2005, Department of Computer Science and Engineering, University of Washington, viewed 8 May 2012, <http://www.cs.washington.edu/education/courses/cse467/05wi/pdfs/lectures/12-DigitalFilters.pdf>.

Hussain, Z, Sadik, A &O'Shea, P 2011, *Digital Signal Processing: An Introduction with Matlab and Applications*, Springer, Berlin.

Ifeachor, E, & Jervis, B 2002, *Digital Signal Processing: A Practical Approach*, 2nd edn, Pearson Education Limited, Harlow, Essex.

National Semiconductor 2003, *LM3914 – LM3914 Dot/Bar Display Driver*, viewed 5 August 2012, <http://www.ti.com/lit/ds/symlink/lm3914.pdf>

Smith, S 1997, *The Scientist and Engineer's Guide to Signal Processing*, California Technical Publishing, San Diego, California.

Spectrum Digital 2003, *TMS320C6713 DSK – Technical Reference,* viewed 2 May 2012, <http://c6000.spectrumdigital.com/dsk6713/V1/docs/dsk6713_TechRef.pdf>.