AC 2010-2233: ADVANCED FUNCTIONS OF JAVA-DSP FOR USE IN ELECTRICAL AND COMPUTER ENGINEERING COURSES

Robert Santucci, Arizona State University
Tushar Gupta, Arizona State University
Mohit Shah, Arizona State University
Andreas Spanias, Arizona State University
Advanced Functions of Java-DSP for Use in Electrical and Computer Engineering Senior Level Courses

Abstract

J-DSP is a java-based object-oriented programming environment developed by Arizona State University as an educational tool for teaching fundamentals and applications of Digital Signal Processing (DSP). This paper presents three new J-DSP modules developed for presenting advanced DSP applications including: feature extraction in music and audio signals, wireless communications, and adaptive echo cancellation.

The feature extraction demonstration enables online simulation of different algorithms that are being used in speech and speaker recognition. Additionally, applications revolving around content-based audio classification and Music Information Retrieval (MIR) are demonstrated. Specific functions that have been developed include modules that are used widely such as pitch detection, tonality, harmonicity, spectral centroid and the Mel-Frequency Cepstral Coefficients (MFCC).

The acoustic echo cancellation tool would help the DSP students to get acquainted with the practical industry applications of the adaptive FIR filters. An acoustic echo canceller (AEC) is used to cancel the unwanted acoustic echoes in telecommunication applications where the loudspeaker and microphone are used in the same vicinity such as teleconferencing. The users of this tool can chose from different signals, e.g. white noise, speech, and music to be played by the loudspeaker to learn the effect of input signals on an acoustic echo canceller. The users can also adjust the length of simulated acoustic impulse response to simulate different acoustic environments, such as car, conference room etc. The number of adaptive filter taps in LMS adaptive filter can also be adjusted. Finally, the residual echo is obtained by subtracting the estimated echo signal from the original echo signal.

The wireless communication tool concentrates on the physical limitations of wireless power amplifiers and how to correct for them using a DSP technique known as predistortion. Wireless power amplifiers typically operate as class AB devices and are subject to a variable amount of clipping depending on input signal amplitude. This variable amount of clipping changes the gain at the fundamental frequency, which must be corrected. It also introduces harmonics that in isolation are easily filtered, but for multi-tone signals are generate harmful mixing products. The gain-based predistorter is described for both memory and memoryless amplifiers.

The simulation software is accompanied by a series of computer experiments and exercises that can be used to provide hands-on training to class participants. This effort is part of a combined research and curriculum program funded by NSF that aims towards exposing students to advanced engineering concepts. The modules developed can be used in signals and systems, communications, DSP, and mixed-signal courses.
Introduction

The progress in artificial intelligence has led to the development of applications such as speaker/speech recognition, web-based applications for the classification of music to come up with better recommendations for the online user, content-based classification of audio signals to represent environmental sounds efficiently amongst many other applications. Algorithms for such applications are developed on a set of basic and advanced features extracted from the given signal. A typical set of features includes tonality, pitch (fundamental frequency), temporal energy, harmonicity, timbre, spectral centroid and deviation and the MFCC. These features play an important role in characterizing the underlying signal and hence stress is laid on extracting as many features as possible that can further relax the constraints on the classification procedure. Feature extraction is now being taught as a part of many undergraduate and graduate DSP courses and is a field open to further research.

The problem of acoustic echoes is very common in telecommunication applications such as teleconferencing and handsfree telephony. An AEC (Acoustic Echo Canceller) is used to maintain the voice quality over a telecommunication link. The operational performance of a conventional AEC depends largely on the adaptive FIR filter. Our focus will be on presenting the AEC as a simple application of the adaptive FIR filter.

The problem of power amplifier linearization is of importance when building practical wireless transceiver circuits. To have these circuits operate with optimal efficiency, DSP techniques can be used to reduce the back off from maximum power output, and thus increase efficiency with which the amplifier converts supply power into radio frequency (RF) power.

These algorithms rely heavily on mathematical and scientific computing tools. So, it helps the students if they can easily simulate and visualize these features as real-time graphs and data plots. In this paper, we describe a series of simulation tools that we have developed using the J-DSP simulation environment. J-DSP is a GUI-driven on-line simulation environment that enables a student to study and verify through graphics various aspects of these algorithms. The J-DSP functions developed exploit the graphical interface and allow the user to simulate complex algorithms by simply forming block diagrams. The software tools are accompanied by exercises that are accessible by students participating in DSP courses and other related courses. The tools described in this paper represent a significant extension of the J-DSP software tools, presented earlier by Spanias et al.

This paper is organized as follows. First, the various features are explained briefly along with their mathematical description. The simulated results obtained from J-DSP are then given. Finally, a set of on-line laboratory tutorials and exercises is developed to review these concepts.

Feature Extraction

The features can be broadly classified into two categories – global descriptors and instantaneous descriptors. In the former, the feature is calculated for the entire signal as a whole. The latter features are calculated for each frame obtained by segmenting the given signal. In this paper, we
are interested in extracting the instantaneous descriptors for each frame. The features in this category are related to the temporal, spectral shape, harmonic and energy features. A brief description of the features and their extraction algorithms implemented in this paper is given as follows.

**Pitch Detection**

Pitch represents the periodicity inherent in the temporal domain or the perceived fundamental frequency of the underlying signal. Although the actual frequency can be determined accurately it may differ from the pitch due to the presence of harmonics. The biased or unbiased autocorrelation sequence for the given frame is calculated from the signal values as follows.

$$r(m) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} x(n+|m|)x(n)$$

This sequence is then checked for periodicity by detecting the location (lag) and the magnitude of the peaks. Based on a certain threshold, certain peaks qualify while others do not. Using this information, we can then detect the pitch of the frame.

![Figure 1. Block Diagram of Pitch Detection Algorithm](image1)

![Figure 2. Pitch Detection in J-DSP](image2)
**Temporal Energy**
The temporal energy can be computed by averaging the squared signal values over an entire frame. This energy can be used to decide whether a given frame is voiced (contains important information regarding the signal) or unvoiced (noise-like) by applying a fixed threshold.

![Figure 3. Temporal Energy of a frame in J-DSP](image)

**Tonality**
Due to the presence of a large amount of noise in the signal, the original tone of the signal might get masked. Tonality is a measure of the signal’s tone-like or noise-like characteristic. The Spectral Flatness Measure (SFM), defined as the ratio of the geometric mean to the arithmetic mean of the power spectrum, is used to compute the tonality for each frame.

$$P(k) = \text{Re}^2[X(k)] + \text{Im}^2[X(k)]$$  \hspace{1cm} (2)

$$SFM (dB) = 10 \log_{10} \frac{\text{GM} \{P(k)\}}{\text{AM} \{P(k)\}}$$  \hspace{1cm} (3)

$$\text{Tonality} = \min(\frac{SFM_{dB}}{SFM_{dB_{max}}}, 1)$$  \hspace{1cm} (4)

An SFM_{dB_{max}} equal to -60 dB is chosen as the reference to compute tonality. A tonality value close to 1 indicates the presence of strong tonal components while 0 indicates noise-like signal. Tonality plays an important role in perceptual coding of audio signals. It is employed in psychoacoustic models such as the MPEG.

**Spectral Centroid**
Spectral centroid is defined as the mean of the distribution of frequency components for a given frame of the signal. The linear frequency or the Mel-scale can be used as parameters on which the weights (magnitude of FFT components) are applied.
Harmonicity
Harmonicity features comprise of harmonicity ratio and the fundamental frequency. The former represents the degree to which periodicity is present in the signal. The latter is the frequency, the multiple integer of which, best explains the content of the signal spectrum. The fundamental frequency is computed using a likelihood approximation based on Goldstein’s algorithm. Harmonicity should be large for speech, music and certain machine noises and smaller for most other types of environmental audio.

Figure 4. Computing the tonality of a given frame

Figure 5. Spectral Centroid

Figure 6. Harmonicity using J-DSP
Mel-Frequency Cepstral Coefficients (MFCC)

The MFCC represents the shape of the spectrum with very few coefficients. The cepstrum is defined as the Fourier transform of the logarithm of the spectrum. The Mel-cepstrum is the cepstrum computed on the Mel-bands instead of the Fourier spectrum. The use of Mel-spectrum allows us to better account for the mid-frequency part of the signal. The MFCC are the coefficients of the Mel-spectrum.

Acoustic Echo Canceller

An acoustic echo canceller is used to cancel the unwanted acoustic echoes in telecommunication applications where the loudspeaker and microphone are used in the same vicinity such as teleconferencing or in Hands Free Kits (HFK’s). The acoustic echo is created by the acoustic coupling between the receive path and the transmit path of telecommunication systems as shown in figure 8.

In figure 8, the far end user speaks through the microphone and the speech signal \( x(n) \) is transmitted to the near end. At the near end, the loudspeaker plays the speech signal \( x(n) \). Let us assume that the user at the near end is not speaking. The speech waves from the loudspeaker get reflected from the floor, walls and surrounding objects and picked up by the near end.
microphone as shown in figure 8. A delayed and attenuated version of speech signal $x(n)$ is collected at the near end microphone. This phenomenon could be interpreted heuristically as if the speech signal $x(n)$ passes through a FIR system with an impulse response $h(n)$ commonly referred as acoustic impulse response (AIR). So, we obtain signal $d(n)$ at the microphone, that is basically the convolution of speech signal $x(n)$ with the $h(n)$. The room impulse response of a sedan car is shown in figure 9. The signal $d(n)$ is then sent back to the far end loudspeaker and the user gets to listen to the filtered version of his/her own speech which could be very annoying and forces them to stop speaking until the echo fades away. In HFK’s, the loudspeaker and the microphone are mounted very closely which causes the played signal to leak via their shell to the microphone.

An acoustic echo canceller (AEC) is used in telephony devices to reduce the acoustic feedback and improve the communication quality. They are placed in parallel to the propagation path of the echo signal as shown in figure 10. The AEC is implemented as a linear adaptive FIR filter. The main function of an AEC is to estimate the acoustic impulse response $h(n)$ from the loudspeaker to the microphone including the reflection paths. A digital replica of the echo signal $y(n)$ is estimated by filtering the received far end signal with estimated AIR $w(n)$ which is then subtracted from the observed microphone signal. The AEC is usually realized on a digital signal processor (DSP) which implies digital-to-analog conversion of the received far-end signal and analog-to-digital conversion of the microphone signal. Since the echo path is unknown and, moreover, can change during the operation time, the linear filter has to be realized adaptively.

![Estimated room impulse response](image)

**Figure 9.** Typical acoustic impulse response of a small car.

In figure 10, let $x(n)$ be the received far end input speech signal (from the far end speaker) which is fed to the near end loudspeaker and $d(n)$ is the signal picked up by the microphone which in this case is the desired input signal (the echo signal). The echo signal $d(n)$ is given by

$$d(n) = x(n) * h(n)$$

(5)

The adaptive filter $w(n)$ is used to model the transfer function of the room in which the loudspeaker and microphone are present in order to generate a replica of the echo, $y(n)$.
The near end noise and speech are assumed to be zero for simplicity.

The estimated echo is then subtracted from the desired input signal \( d(n) \) yielding the estimation error signal which is given by,

\[
e(n) = d(n) - y(n)
\]

The aim is to cancel the echo signal \( d(n) \) and to minimize the error signal \( e(n) \) as much as possible. There are several adaptive algorithms which could be used to estimate the echo. Most commonly Normalized Least Mean Square (NLMS) is used for the adaptation of weight vector in the FIR filter.

\[
w_{n+1} = w_n + \mu \frac{e(n)x_n}{x_n^T x_n + r}
\]

where \( \mu \) is the step size, and \( r \) is a small constant called the regularization constant.
Amplifier Linearization and Predistortion

Overview
Mobile communication systems are subject to two severe physical limitations: power and bandwidth. First these devices are not typically connected to the power grid; therefore power consumption is critical to battery life and device usability. The second limitation is that bandwidth available over the air is both limited and expensive; therefore both a signal’s amplitude and phase must be modulated to achieve high data rate within a given bandwidth. Amplitude modulation can have a significant impact on the amplifiers used to drive strong signals into cellular antennas. To attain maximum efficiency, these amplifiers must be operated near their maximum power output. However, near maximum amplitude the complex gain provided by the amplifier changes. This divergence away from an “ideal” amplifier results in a distortion of the output constellation from the amplifier’s presented input constellation.

Figure 12: Typical Predistortion System
Pre-distortion is a method of canceling amplifier distortion by warping the signal to be transmitted to invert the amplifier distortion. A typical transmitter system utilizing predistortion is shown in Figure 12.

One of the mechanisms in which gain is reduced at high drive strengths is relevant for class AB amplifiers, commonly used in wireless systems. In a class AB amplifier, the conduction angle is between 180 and 360, meaning the waveform is clipped for a fraction of a cycle less than 180 degrees. This clipping usually occurs around a fixed level. As a result, the conduction angle is a function of the input signal magnitude. This variable clipping causes two effects: amplifier gain at the fundamental frequency decreases with increasing input magnitude and the production of harmonics. A power series is frequently fit to model the non-linear transfer function through the amplifier due to nonlinearity using least-squares techniques. This fitting yields amplifier gain $G$ as a power series with weighting parameters $a_k$, where $K$ is the maximum order of nonlinearity.\(^7\)

$$G(v(n)) = \sum_{k=1}^{K} a_k |v(n)|^{k-1} \quad (11)$$

When a multi-tone signal is fed into the amplifier, mixing products are produced as the harmonics are multiplied. Two sets of these mixing products produce problems. First, mixing products that produce tones around the signal frequency are problematic. These mixing products result near the signal frequencies and are thus difficult to filter out and result in spectral spreading. Secondly, mixing products are produced near DC. These are discussed later.

Figure 13: Illustration of Variable Fundamental Gain and Harmonic Levels

This class AB clipping mechanism, the generation of harmonics, and the variation of fundamental gain can be demonstrated in JDSP utilizing the following test bench. In the following example, the input frequency is considered to be at the RF data rate. It can be seen that reducing the amount of power into the power amplifier (PA), also known as backing off the power, reduces the level of the harmonic content.
Memoryless Power Amplifiers

The gain-based predistorter\(^8\) compensates for variable amplifier gain around the fundamental. This topology exploits two assumptions. First, it assumes compression occurs primarily as a result of input power. Second, it assumes adjacent input power levels have highly correlated power amplifier complex gains. Consequently, modem outputs are assigned LUT bins based on their power level. Each bin has a correction factor which multiplies the modem output, bringing it back to nominal gain. In essence, this approach compensates for the amplifier going into compression by driving it even harder to get the desired output as gain reduces in the compression region. For this to work, the amplifier must be monotonic, always having increasing output power for an increase in input power.

The correction factors for each LUT bin can be calculated adaptively utilizing the complex least-mean-squares (LMS) algorithm\(^9\). To do this, a receiver unit must take an input coupled just before the antenna that indicates what symbol is actually being transmitted by the amplifier. In a half-duplex communication system, often the already present receiver can be utilized for this purpose. This receiver measures the value actually transmitted, and feeds that as the actual output transmitted to the adaptation algorithm to be compared with the desired output transmitted. Note that for this to work requires a well-calibrated receiver where the quadrature demodulator produces little offset and the timing of the received signal can be synchronized a
delayed version of the transmitted signal. The algorithm operates on the complex baseband signal.

\[ e(n) = v_{\text{Des}}(n) - v_{\text{Act}}(n) = g_o v_{\text{in}}(n) - G(b(n)v_{\text{in}}(n)) \] (12)

By constraining the input based on a narrow input power range, the non-linear term of \( G(\cdot) \) can be made smaller such that \( G(\cdot) \) becomes a roughly linear function of the input signal. This allows replacing non-linear function \( G(\cdot) \) with a scalar value \( G \). Dividing the actual and desired output signals by \( g_o \), the nominal PA gain, makes it such that the expected gain of each path is 1. Taking the expectation because of randomness in \( G \) due to assumption error, variation, and noise, the complex LMS algorithm will minimize the expected error power. The iterative equation used is:

\[ b(n + 1) = b(n) + 2\mu e(n)v_{\text{in}}^*(n) \] (13)

where \( \mu \) is a convergence factor and \( b \) tracks \((g_o/G)\) within each bin. Smaller values of \( \mu \) take longer to converge, but give can give more accurate settling in the presence of randomness. Larger values of \( \mu \) will give higher steady-state error power if in the presence of good \( G \) assumptions. However, when \( \mu \) is too large the algorithm will overcompensate and go unstable. By choosing more look-up-table bins the assumptions in making \( G \) linear hold better, but now more points are needed to make the algorithm converge.

An example figure of an implementation topology is shown in figure 16. A test bench in JDSP, shown in figure 17, allows for experimenting with gain-based predistorters of different sizes and adaptation speeds.
Power Amplifiers with Memory Effects

Low-frequency mixing products near DC can prove problematic for real amplifiers. The resulting low-frequency envelopes can be responsible for memory effects, or the dependency of amplifier outputs on previous inputs along with current inputs\(^\text{10}\). Slowly-varying low-frequency signals can be at high amplitudes and low-amplitudes for longer durations than the amplifier packages thermal time constant, and can thus change the temperature and thus the electrical characteristics of the amplifier. Secondly as signals get wideband, the analog low-pass filters used to maintain a constant DC bias on the amplifier must be more broadband and thus require better performance that can be difficult to achieve at several MHz. The result is variation in the amplifier output from an ideal amplifier.

The gain-based LUT presented previously has been extended to compensate for power amplifier with memory effects by Jardin\(^\text{11}\). In this algorithm, the PA is modeled as a power series function with finite non-linear memory, \(M\). To simplify things, an assumption is made that all power-series components are a function of only one lag. The resulting power amplifier output is:

\[
G(v(n)) = \sum_{m=0}^{M} \sum_{k=1}^{K} a_{k,m} v(n-m) |v(n-m)|^{k-1} \frac{v(n)}{v(n)}
\]  

(14)
Here a similar input assumption is made, except now the $G(\cdot)$ is assumed to be a linear finite-impulse-response filter for a given bin of input ranges. Notice this assumption is weaker as the previous input signals to the power amplifier are not necessarily of consistent amplitudes. Here the LMS can again be applied in its vector format utilizing the same topology as before, although other topologies may yield better performance. The filter coefficients for each lag at time $n$, $b(n)$, can be found using an extension of the technique used for the gain coefficient. Here $\mathbf{v}_{in}(n)$ a vector of the most recent input points.

$$b(n + 1) = b(n) + 2\mu e(n)\mathbf{v}_{in}^*(n)$$

(15)

**Tutorials and Exercises**

**A. Voiced/Unvoiced Decision Making**

Often speech or audio signals are corrupted by noise and involve brief periods of silence. Hence, such frames will have very low energy compared to the other frames where the speaker is speaking or music is being played. Differentiating between such frames helps us to determine the level of information contained in a given frame. This information is binary coded as voiced (binary 1) and unvoiced (binary 0). A threshold, typically in the range of $10^{-3}$, is chosen to demarcate between the two aforementioned classes.

- Using the Long Signal Generator block and load it with any kind of audio or speech signal.
- Connect it to an Energy block.
- Connect this block to the Plot block to observe the temporal energy values as each frame is evaluated.
- Choose a proper threshold. Classify each frame as voiced/unvoiced on the basis of the principle explained above.

**B. Transient Index**

The transient index is useful in detecting and segmenting sounds whose spectral characteristics exhibit consistent fluctuations between adjacent frames, e.g., crumpling newspaper. This index can be computed from the MFCCs obtained over a collection of short number of frames. The steps to compute the transient index are outlined as follows:

- Load any audio or speech signal using the Long Signal Generator.
- Connect this block to the MFCC block.
- Using the equation given in (16), compute the transient index.

$$\kappa_i = \sum_{k=t-(N+2)}^{t} ||MFCC_k - MFCC_{k-1}||_2$$

(16)

Where, $t$ is the current frame, $MFCC_k$ is the MFCC of $k^{th}$ frame and $N$ is the short number of frames over which the transient index is computed.

**C. Tonality**

This feature is widely used in psychoacoustic models employed in the MPEG series of standards. Tonality helps characterize the tonal or noise-like nature of the signal quantitatively. In this exercise, we study the effect of noise on the tonality values of a given frame.
- Use two Long Signal Generators. Load a speech or audio file in one, while a white or colored noise in the other.
- Connect these two blocks to an adder.
- Connect the adder output to the Tonality block.
- Finally, connect this to the Plot block to observe the tonality values of each frame.
- Now, in the Signal Generator block containing the noise signal, vary its gain and observe the changes in the displayed tonality values.
- You should be able to observe that increasing the noise gain destroys the tone-like characteristic of the signal. This is marked by the decrease in tonality values from 1 to being closer to 0 now.

D. Brightness
The brightness of any audio/speech signal is characterized by its spectral centroid. The spectral centroid, being the barometer of the spectrum, is an indicator of the frequency with higher probability mass, which in turn determines the level of brightness of sound.
- Load the Long Signal Generator first with a Male Speaker and then with a Female speaker.
- Connect this block to the Spectral Centroid block.
- Connect the output to a Plot block.
- Observe the spectral centroid values of each frame for the two different cases mentioned. Since brightness indicates the amount of high frequency components present in the spectrum, you should be able to observe that the spectral centroid is able to objectively depict these qualities for both the signals.

E. Acoustic Echo Cancellation
- Load the Long Signal Generator first with a Male Speaker.
- Connect this block to the Filter blocks and LMS block as shown in figure11.
- Connect the output to a Plot block.
- Observe the final estimated LMS filter coefficients and also observe the frequency response of the estimated room impulse response.

F. Power Amplifier Linearization
- Connect the block diagrams as shown in Figure 17.
- Click on the long signal generator, select the “OFDM BPSK 64-carrier 4xOSR” data source. Leave the window open as the play button will be used to transmit subsequent frames.
- Click on the Plot block, select Magnitude^2, dB, and set y-axis to a manual range -60 to 20.
- Click on the PALinearized Block. Set mu=0.1, LUT size=32, Power Back Off = 1.0dB.
- Examine and compare the frequency domain output of the PA itself without predistortion (top plot) and the PA when Predistortion is utilized (bottom plot).
- Transmit more frames by hitting the frame advance button “>” in the long signal generator, observe the correction get better with each additional frame of data.
- Rewind to frame 1 using the long signal generator by hitting the “|<” button.
- Repeat for mu = 0.2, 1.0, 5.0, 30.0 and examine the results
- Experiment with new combinations of mu, power back off, and LUT size.
Conclusions

This paper presented J-DSP simulation modules and exercises to teach feature extraction techniques to undergraduate and graduate students. The exercises designed based on the J-DSP simulation tool, may be used by instructors in a class setting to demonstrate key signal processing concepts associated with these algorithms. Prior work on J-DSP and its assessment has been reported in 12-14.

Acknowledgements

This work has been funded in part by NSF CCLI award number 0817596.

Bibliography