Chapter 3. The Solution

Section 3.1. A Hybrid ASK/FSK Approach

Section 3.1.1. Keying on Sinusoids

After considering various approaches, including options based on audio frequency spread spectrum and differential phase shift keying, we knew a technique was required that did not depend on the preservation of exact timing, phase, frequency, or amplitude. Eventually these requirements led us to develop a hybrid scheme similar to amplitude-shift keying (ASK) and frequency-shift keying (FSK) [14]. The digital signature is encoded using 167 sinusoids added to a filtered version of the audio component of the television signal. A patent disclosure has been filed covering the newly developed communications method in applications other than interactive television [15].

The 167 sinusoids consist of five groups that perform three distinct functions. Three groups contain the digital data and the error detection coding bits. Another group contains the sinusoids used in the “control function” for synchronization and for weighting estimates of received data. The final group contains a pattern of sinusoids used to detect and quantify the amount of frequency distortion experienced in the current signal block. These groups and functions will be described individually in more detail shortly. But first a succinct general description of the coding scheme will be presented.

In the hybrid method the presence or absence of sinusoids in specific frequency locations conveys data. Sinusoidal frequencies are chosen to correspond to the bins of a

4096 point FFT with a sampling rate of 16.0 kHz. This allows fast and simple detection and decoding on the receiving end, where all computations must be performed continuously and in real time. With a 16.0 kHz sampling rate, a 4096 sample block of signal has a duration of 256 ms. After a block has been processed, the oldest 205 samples are shifted out and 205 samples of new signal are shifted in on the other end (i.e., the signal blocks are overlapped by 95 percent). The refreshed block is then processed, and the cycle is repeated. All processing of a refreshed block is therefore performed within 12.8125 ms.

Since multiple sinusoids will be packed into a relatively small frequency range, windowing should be applied at the receiver to minimize leakage interference between the sinusoids. If no window (i.e., a rectangular or “boxcar” window) is applied to the data at the receiver, each sinusoid will show up as a sinc function in the frequency domain. Each sinc function will have a main lobe width of almost two FFT bins, a first sidelobe level thirteen dB below the main lobe peak, and sidelobes spaced one FFT bin apart that roll off at a rate of six dB per octave. Table 3.1 below shows the peak sidelobe levels and the rolloff rates for several commonly used windows [16]. In the table *M* is the number of data points available; here *M* would be 4096.

Table 3.1 Window Characteristics

Window Type	Peak Sidelobe Amplitude (Relative, in dB)	Approximate Width of Mainlobe	Rolloff Rate (dB / Octave)
Rectangular	-13	$4\pi / M$	6
Bartlett	-25	$8\pi / (M\text{-}1)$	12
Hanning	-31	$8\pi / (M\text{-}1)$	18
Hamming	-41	$8\pi / (M\text{-}1)$	6
Blackman	-57	$12\pi / (M\text{-}1)$	18

The on or off (binary one or zero) condition of the sinusoid is determined by comparing the magnitude of the FFT at a signal bin with the magnitudes of its neighbor bins some distance away to the left and right. If the magnitude at the signal bin exceeds both its reference neighbors' magnitudes by more than a pre-set threshold, a sinusoid is considered to be present and a digital one is detected. If the condition is not satisfied, the sinusoid is considered to be absent and a digital zero is received. Figure 3.1 illustrates the sinusoid detection process. Note that although the sinusoids are chosen to nominally correspond to the bin locations of the FFT, they will be moved in frequency by the tape wow and flutter. Thus the sample points in the frequency domain will in general not line up perfectly with the main lobe peaks and the zero-crossings of the sincs. Such a shifted case is represented in the figure.

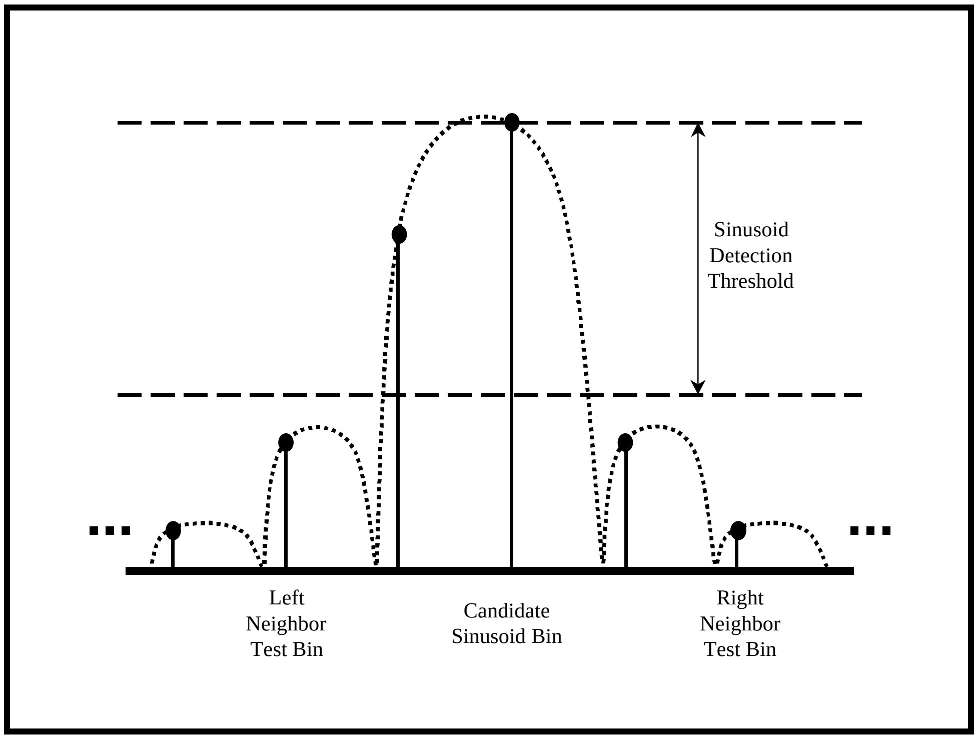

Figure 3.1 Sinusoid Detection Process

The spacing of the sinusoids, the distance to the neighbor reference bins, and the detection threshold must all be set based on the window applied at the receiver. The sinusoids must be spaced far enough apart so that their mainlobes are distinct and are able to decay enough before the next mainlobe begins. Similarly, since the mainlobe widths vary with the window applied, the distance to the reference neighbors should be changed accordingly. If a rectangular window is used in a clean signal environment, the neighbor bins could be chosen as close as a single bin away, since the sinc has its first zero crossing a single FFT bin away from the peak of the main lobe. However, since the sidelobes are relatively high and roll off very slowly for the rectangular window, the sinusoids would have to be spaced far apart to ward off interference between sinusoids. Also, for any window choice, the frequency variation due to wow and flutter must be considered. Although the sinusoids are chosen to nominally fall on the FFT bin frequencies, for any 4096 block of data the wow and flutter will cause a change in frequency for all sinusoids. Once a sinusoid moves from its nominal location, it is unlikely that it will be centered on an FFT bin. Instead, as mentioned above and depicted in Figure 3.1, the FFT bin "samples" can fall anywhere on the main and side lobes. Therefore, the reference neighbors should be spaced at least two bins away from the center signal bin, even in the rectangular window case.

Figure 3.2 below shows how a 32-point FFT with a rectangular data window would sample the spectrum of a complex sinusoid located at a normalized frequency of 0.25. The dotted lines show the underlying sinc function (the true spectrum), and the circles represent the samples that are computed by the FFT. Figure 3.3 shows where the FFT sample points fall if the sinusoid moves slightly, to a normalized frequency of 0.2656. Note that there is no sample point at the peak of the main lobe for the shifted case, and the remaining sample points occur near the peaks of the sidelobes instead of at the zero-crossings of the sinc. In the un-shifted case there is over 250 dB separation between the main lobe peak sample and the other samples at the zero-crossings. (Theoretically there is infinite separation; the values below –250 dB show up here because of finite word-length effects in the computations.) For the case with the shifted sinusoid, the separation has been reduced to between ten and thirty dB.

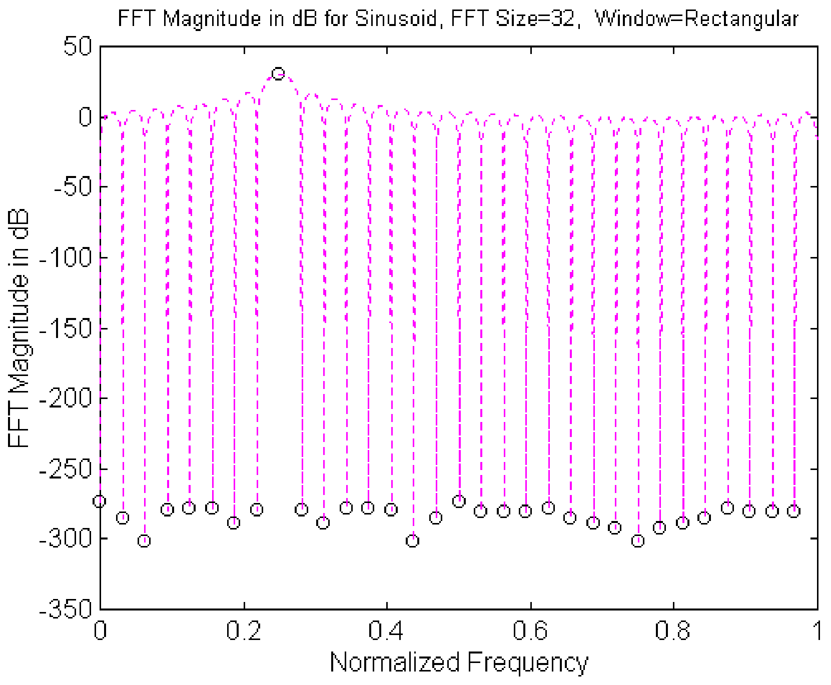

Figure 3.2 FFT of Sinusoid, Rectangular Window

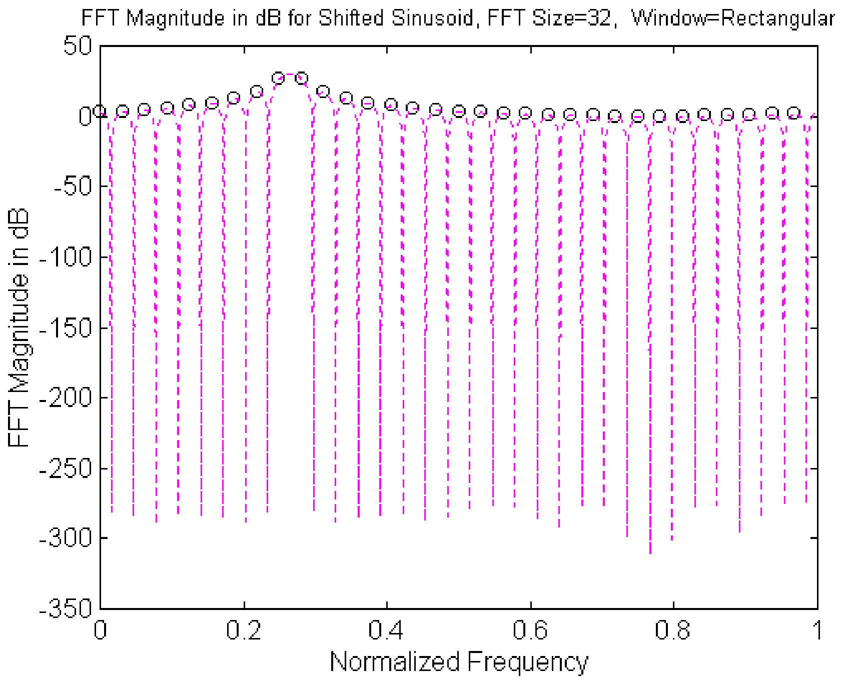

Figure 3.3 FFT of Shifted Sinusoid, Rectangular Window

If a Hanning window is applied to the data before the FFT is computed, the main lobe width will increase by slightly more than a factor of two (see Figure 3.4). The peak sidelobe level will be reduced, and the sidelobe rolloff rate will increase. Figure 3.5 shows the Hanning window result for the shifted sinusoid case. A Blackman window furthers the effects on the main lobe width, the peak sidelobe level, and the sidelobe rolloff rates. Results for the Blackman window applied to the two sinusoids are shown in Figure 3.6 and Figure 3.7. Note that for the windowed cases (as opposed to the rectangular or “no-window” case), the sample points will not all fall on the zero-crossings regardless of where the sinusoid is located. However, since the sidelobes are significantly reduced, the separation in value between the “peak” sample and its neighbors a few bins away is sufficient for our purposes, and will allow proper detection of sinusoids.

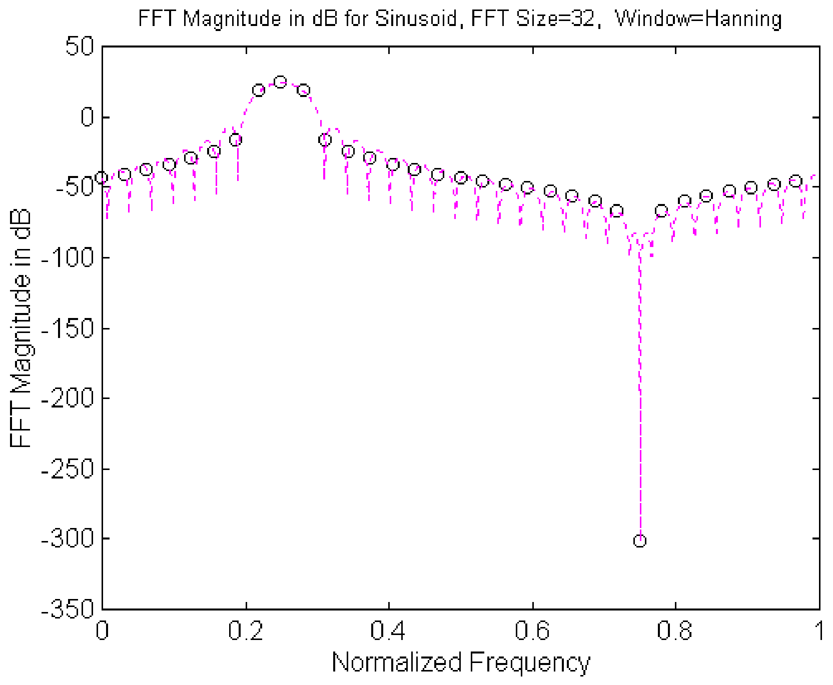

Figure 3.4 FFT of Sinusoid, Hanning Window

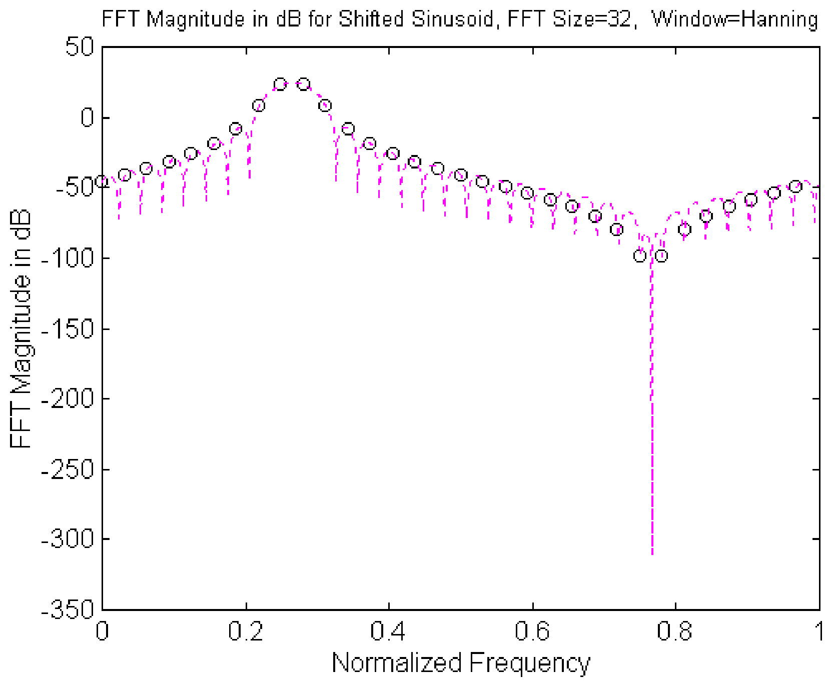

Figure 3.5 FFT of Shifted Sinusoid, Hanning Window

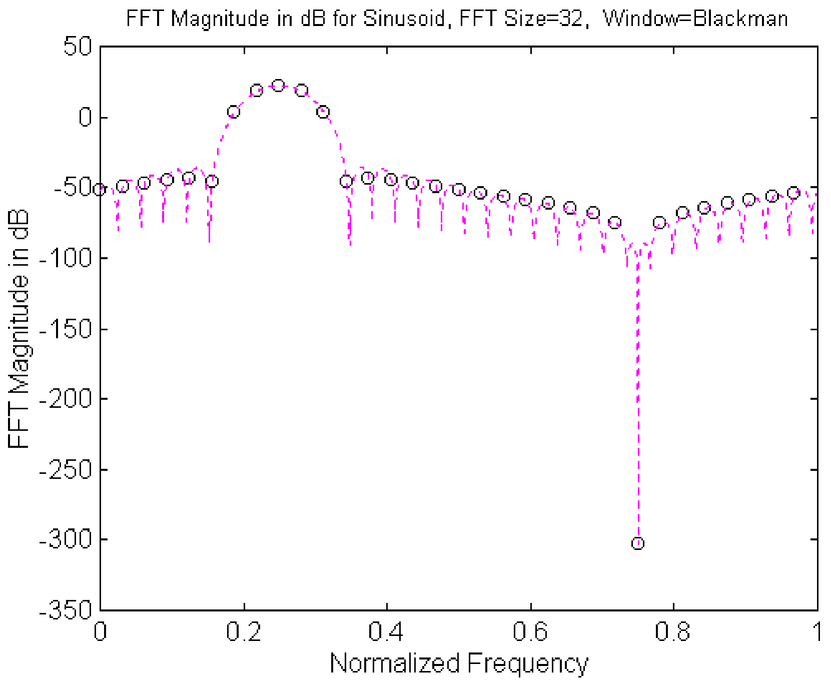

Figure 3.6 FFT of Sinusoid, Blackman Window

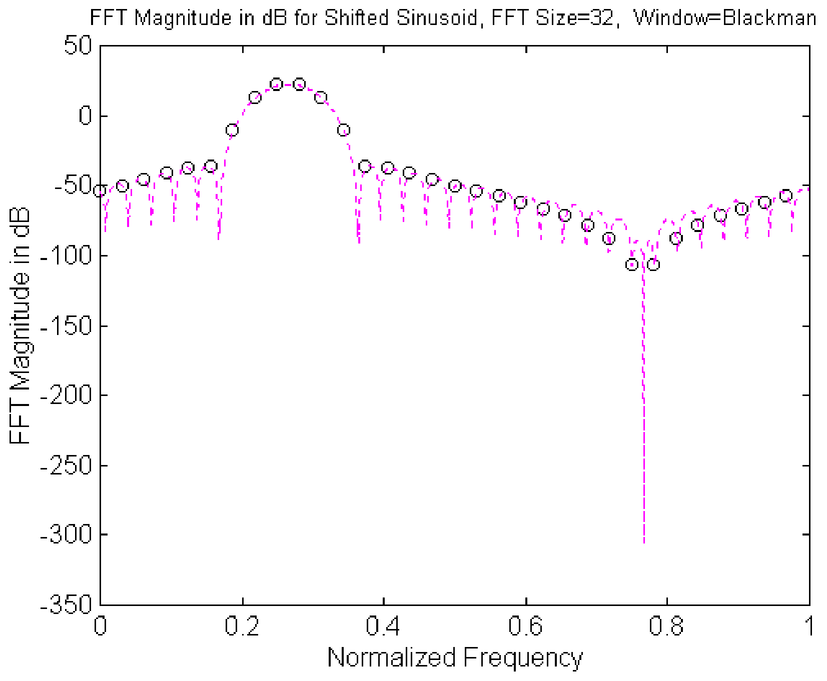

Figure 3.7 FFT of Shifted Sinusoid, Blackman Window

After testing with various window options (see Section 4.2 later in the thesis), the Hanning window was chosen as a compromise between mainlobe width and sidelobe reduction. The reference neighbors were then defined as the bins two away to either side of a signal bin, and the detection threshold was set to four dB. In a clean environment the four dB threshold is easily surpassed, and could be set much higher. Figure 3.8 depicts an actual case of the sinusoidal detection process using the FFT magnitude spectrum of an audio signal containing an embedded code. The asterisks mark the candidate FFT bins, and the circles mark the bins two away on either side of the centers. The bit sequence [1 0 1 0 0 1 1 1] is represented in the example. Note that in this particular example the signal-to-noise-ratio was relatively high, and so the sinusoid peaks are clearly evident. However, after signal degradation by tape wow and flutter and by transmission through various channels, the sinusoidal peak bins are often only about four or five dB above their reference neighbors. (Of course at times they can also be completely attenuated or overwhelmed by interference, and their detection is impossible.) The four dB value for the threshold was obtained after testing in various signal environments. If set lower, noise variance easily satisfies the detection criteria and false detections result. If set much higher, actual sinusoids are unable to satisfy the condition in low SNR environments. The four dB threshold is a compromise between the two extremes that seems to work well under a wide range of signal conditions.

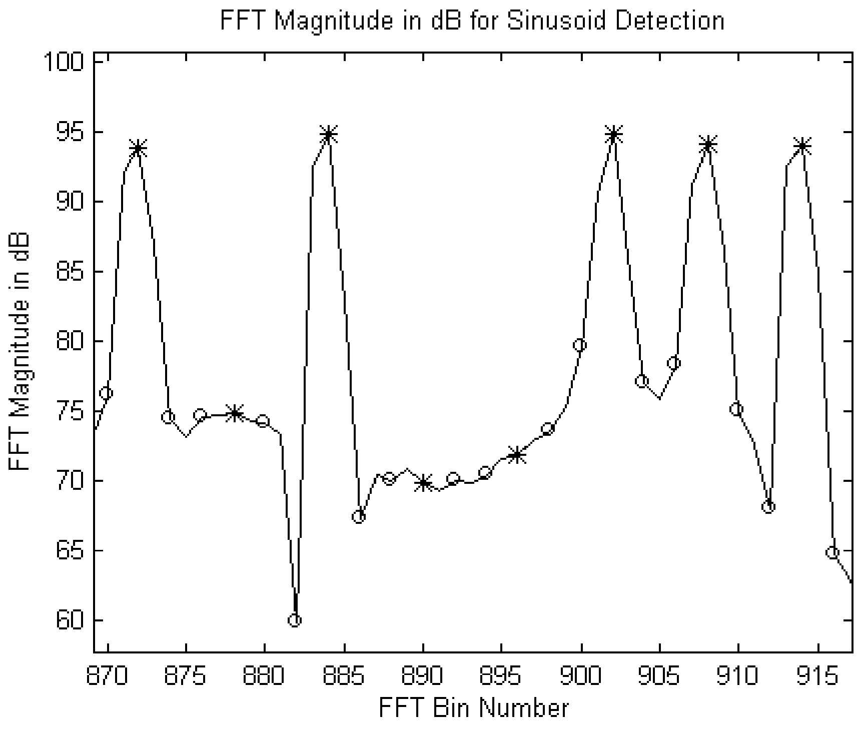

Figure 3.8 Sinusoid Detection by FFT Magnitude Results

Section 3.1.2. Control Function

As mentioned in the previous sub-section, one group of sinusoids is used in the "control function" for synchronization and weighting of data estimates. Figure 3.9 below demonstrates self-synchronization with the control function. As the FFTs are performed on refreshed blocks of signal, the 21 control sinusoid locations are examined. If sinusoids are detected at fourteen or more of these locations (a two-thirds majority), valid data is assumed to be present in that FFT block, and the data decoding process is initiated by polling the data sinusoid locations. In the figure the asterisks indicate when a two-thirds majority of control sinusoids is present, and hence when the digital data is available.

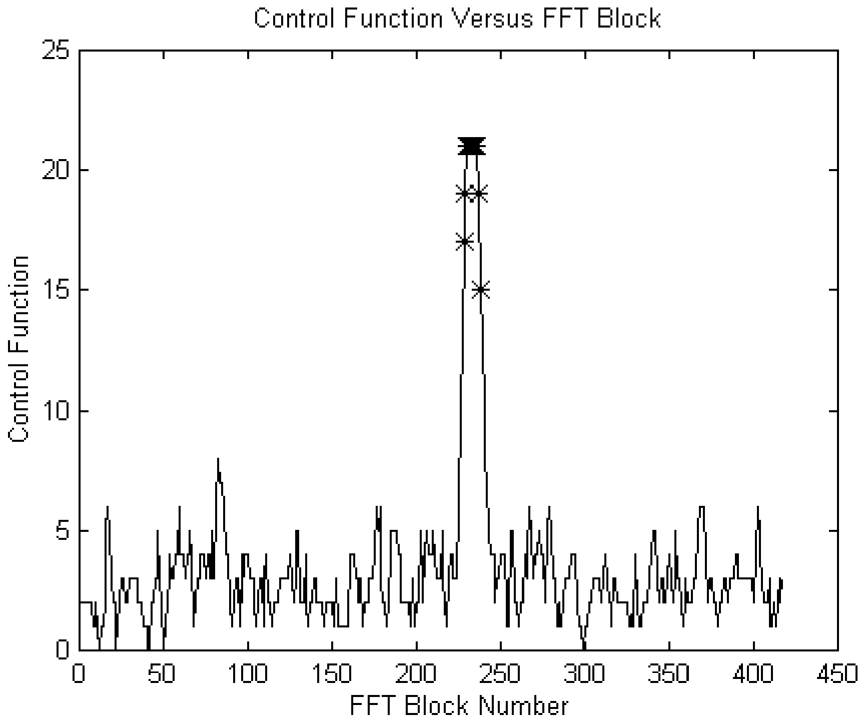

Figure 3.9 Control Function for Synchronization

The control sinusoids are uniformly interspersed with the data sinusoids throughout the entire four kHz band being used (from 2.4 to 6.4 kHz). Thus the control sinusoids not only serve a synchronization purpose, they also provide an indication of the quality of the received data in that particular signal block. Since a triplication code is used for the data (to be discussed in Section 3.1.3), a two-thirds majority requirement for the control function is commensurate. Furthermore, when the data is being tabulated over successive FFT blocks, the results of each block are weighted according to the value of the control function in that block. For example, we have more confidence in the data when the control function is 21/21 versus when it is 14/21. The data associated with such blocks should be weighted accordingly. Thus digital ones are represented by positive control function values and digital zeros by negative control function values.

Section 3.1.3. Data Subbands and Triplication Codes

Since the audio transmission channel is noisy and corruptive, the error rate for individual sinusoidal detection is quite large. Tape media not only distort the signal with wow and flutter, but also add noise. The acoustic paths from the television speakers to the AudioLink microphone introduce multipath interference as well as additional noise. Extraneous room noises such as human voices or music can also compete with the desired audio signal upon reception at the AudioLink. When these multiple corruptions are combined in the received audio signal, the resulting signal to noise ratio is often quite low, and therefore the detection rate for a given sinusoid is likewise often low.

Most communication systems employ coding schemes to remove inefficient redundancy in a signal source, so that fewer bits and less bandwidth are necessary for transmission. However, *efficient* redundancy is usually inserted so that transmission can be successful despite some percentage of individual bit errors [17], [13]. If the bit error rate is relatively small for a channel, one of many efficient error correction codes can be employed to correct bit errors at the receiver. However, as the rate of individual bit errors rises and/or if they occur in large bursts, more redundancy is necessary, and many of the commonly used schemes cannot effectively handle the situation. Such is the case in our application. The individual bit error rate is often on the order of 33 percent for a given signal block. (Note that if it rises much above 33 percent, the data block is discarded due to the control function, which would presumably have a similar reception statistic.) Furthermore, bit errors commonly occur in bursts since many signal corruptions are frequency dependent and destroy a succession of data bits.

In an (n,k),t block code, a code word n bits long is used to transmit k data bits (n>k). The number of check digits in a block code is $m = n - k$, and the number of bit errors it can correct is given by t. Different (n,k) block codes will be capable of correcting different numbers of bit errors (t), in general. But the error correcting capability of all block codes (linear and non-linear) is limited by the Hamming bound [18]:

$$2^{\mathrm{m}} = 2^{n-k} \geq \sum_{j=0}^{t} \binom{n}{j}$$

(3.1)

where $\binom{n}{j}$ is the binomial coefficient

$$\binom{n}{j} = \frac{n!}{j!(n-j)!}$$

(3.2)

Note that linear codes, by definition, are those that form vector subspaces over finite Galois fields [19].

Because of the frequency range available for our transmission, a maximum of 141 data sinusoids can be accommodated. The IVDS system requires 35 signature bits plus twelve error detecting bits (to be described shortly), for a total of 47 data bits. Therefore, in our application n = 141, k = 47, and $m = n - k = 94$. Using a computer to solve for the maximum number of bit errors that can be corrected, we obtain t_{max} = 25 bits. It should be noted that the Hamming bound is a necessary but not a sufficient condition, and it may not be possible to construct codes with certain parameters even if the bound is satisfied. Given the high error rate of our channel, we can often expect more than 25 bit errors out of the 141 available bits. This bound shows that no block code can perfectly handle our situation.

Since many signal corruptions encountered in our application are frequency dependent and destroy a succession of data bits, a triplication code has a chance of succeeding where other block codes would fail. The 141 sinusoids are divided into three groups - one in the lower frequency region, one in the upper frequency region, and one in the middle. The 47 data bits are then coded into each of the three regions. Since signal degradation most often occurs within a given frequency band for our channel, it is likely that other bands will be preserved well enough for successful transmission. In an extreme case an entire group of 47 bits can be destroyed and the transmission can still be successful due to the triplication code. An example would be where human voice interference corrupts a large part of the lower frequency region. As long as the corresponding replicated bits in the middle and high frequency regions are received correctly, the triplication code will yield the correct result by assigning a bit value based on a two-thirds majority.

Although block codes cannot handle the large number of errors we would encounter if they occurred randomly, several coding techniques have been devised for the burst error cases. One non-binary possibility is Reed-Solomon (RS) codes [19]. RS codes, however, would be too taxing on the IVDS hardware in terms of memory and computational requirements, and so they could not be used. However, BCH codes [19] (named for their inventors, Bose, Chaudhuri, and Hocquenghem) could be used in groups, with the bits interleaved in the frequency domain [19]. BCH codes are considered among the most powerful linear codes because they possess the best combination of high rate and high error correction for a given block length. One (15,5) BCH code can correct $t = 3$ or fewer bit errors [9]. If ten blocks of these codes were used in our application (fifty data bits enclosed in 150 code bits), and the bits were interleaved in the frequency domain, an error burst up to thirty bits long could be corrected.

Figure 3.10 below provides a representation of the probability of successful transmission with the code options discussed. The Hamming bound (the dashed line) shows that for a (141,47) code, the maximum error correcting capability is 25 bits, after which any block code will fail. In practice the cutoff is likely to be far to the left of where it is shown in the figure, since it is probably not possible to create a code that achieves the performance at the bound. If ten groups of (15,5) t=3 BCH codes were interleaved in the frequency domain as mentioned above, the performance would be similar to that shown by the dotted line. Since any group by definition (t=3) can correct up to three bit errors, successful transmission is guaranteed up to that point. However, if four errors are present, there is a small but non-zero probability that they will all occur in the same group, and thus the probability of success drops slightly below unity at four bit errors. At 31 errors at least one of the ten groups will have at least four errors, and the code will fail. Therefore the probability of success is zero if more than thirty errors are present. The triplication code is guaranteed to be able to correct only a single error. At two errors there is a small but non-zero probability that they will occur for the same data bit, and the transmission will fail. However, success is still possible until 48 bits are in error, since at that point at least one data bit will be lost. As the figure shows, the triplication code provides an extended range of possible success at the expense of a higher rate of failure for smaller rates of bit errors. It should be noted that the figure is not to scale, and the exact curves depend strongly on the statistics of the bit errors and on the bursts in which they occur. These statistics are difficult if not impossible to compute for our application. Since the AudioLink often operates in the region where many (20-50) burst errors are present, the triplication code is a reasonable choice.

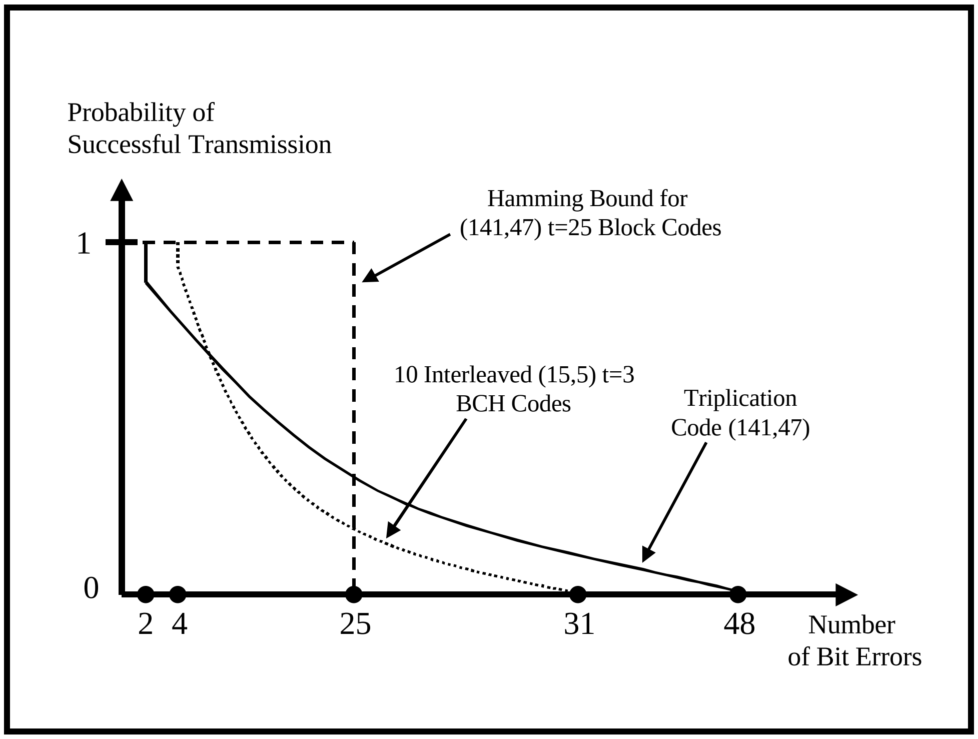

Figure 3.10 Code Comparisons

Because of deadlines imposed by the project sponsors, the coding method had to be chosen and implemented in a very short time period. The digital signal processor (DSP) used in the receiver unit was already heavily burdened, both in memory and in computational requirements, so complicated or demanding coding methods were not feasible. Therefore, given the large bit error rate on our channel, the constraints imposed on hardware implementation, and the lack of time available to investigate alternative strategies, we chose the simple triplication code.

Section 3.1.4. Bit Voting in Time

Once the detection process is initiated by the control function, valid data received in the current and subsequent FFT blocks are tabulated. The end of data transmission is detected by the level of the control function dropping below 14/21 and remaining low for a specified period of time (say one second). (Because of implementation issues, this specification was later changed. The AudioLink now listens for approximately 500 ms after the control function first turns on, and the control function is not used for determining the end of the data transmission. In later versions of the AudioLink, it may be desirable to return to the original method of triggering on the descent of the control function.) When it is determined in this manner that the data transmission is complete, final decisions are made regarding the individual data bits. Since the bit votes from each FFT block (weighted by the control function from each block) have been summed over time, a final decision can be made regarding each bit's status by a simple threshold test. If a bit's value is positive it is considered to be a digital one, and if it is negative it is considered to be a zero. The triplication code is then decoded by a two-thirds majority vote among the three frequency subbands. Finally the cyclic-redundancy-check (CRC) is confirmed to verify error-free reception (to be discussed in Section 3.1.5), and the 35 bit digital signature results.

Table 3.2 below demonstrates the data decoding process (without the CRC) through an example. Suppose we desire to transmit a digital signature of two bits [1 0], and on the decoding end the control function is detected as shown in the table. When the control function is below fourteen no data is present. When the control function is fourteen or larger the data bit locations are analyzed to see if sinusoids are present. When a sinusoid is present, the value of the control function is added to the corresponding data bit location. Likewise, the lack of a sinusoid represents a digital zero and the value of the control function is subtracted from the corresponding data bit location. Once the control function drops below fourteen and stays there, the data collection process terminates. Any bit locations containing positive values are considered to be digital ones, and any locations containing negative values are zeros.

Table 3.2 Example of the Data Decoding Process

FFT Block	Control Function	Bit 1-1	Bit 2-1	Bit 1-2	Bit 2-2	Bit 1-3	Bit 2-3
1	4	0	0	0	0	0	0
2	3	0	0	0	0	0	0
3	14	14	-14	14	-14	-14	-14
4	18	18	-18	18	-18	-18	-18
5	21	21	-21	21	-21	21	-21
6	20	20	-20	20	-20	20	-20
7	16	-16	-16	16	-16	-16	-16
8	13	0	0	0	0	0	0
9	7	0	0	0	0	0	0
10	2	0	0	0	0	0	0
Final Values		57	-89	89	-89	-7	-89
Bit Decisions		1	0	1	0	0	0

Note that Bit 1-1 contains an error during FFT block seven. However, the correct value was received often and strongly enough in other FFT blocks to produce the correct bit decision at the end. Bit 1-3, however, has been corrupted several times (in FFT blocks 3, 4, and 7). Multipath interference can cause a null in the frequency domain resulting in such a repeating bit error. In this case the bit decision is incorrectly made as a zero. However, the proper digital signature will still be extracted due to the redundancy of the triplication code. The values for Bit 1 are [1 1 0] yielding a 1. Similarly the values for Bit 2 are [0 0 0] yielding a 0.

Section 3.1.5. Cyclic Redundancy Check

Since many sinusoids can be attenuated due to transmission losses, multipath effects, and noise interference, redundancy and error correction techniques are necessary. The triplication code provides some redundancy and error correction capability, as discussed in Section 3.1.3. However, it remains imperative to be able to verify the integrity of the digital signature after the triplication code is undone. For this error *detection* purpose, a cyclic redundancy check (CRC) [20] of twelve bits is appended to the 35 bit digital signature, resulting in 47 data bits total. A twelve bit CRC provides an error coverage of $1-2^{-12} = 0.999756 = 99.9756$ percent. It can detect: a) all single error bursts of twelve bits or less; b) 99.95 percent of all error bursts of length thirteen; and c) 99.22 percent of all error bursts of length greater than thirteen [19].

Cyclic redundancy checks can be computed based on generator polynomials using shift registers, feedback, and exclusive-or gates [21]. The data stream is fed in, and the states of the shift registers after clocking the data stream through provide the CRC bits to append to the data stream. On the receiving end, the data stream and CRC bits are sent through the identical shift register. After all bits are clocked through, the registers will be at zero state if no errors were detected. One or more of the registers will contain a one if there were errors in the data stream. The CRC-12 we used was derived from the only twelfth-order generator polynomial listed [19]:

$$g_{12}(x) = x^{12} + x^{11} + x^3 + x^2 + x + 1 = \left(x^{11} + x^2 + 1\right)\left(x + 1\right) \tag{3.3}$$

There are two options for use of the CRC at the AudioLink receiver. The CRC is always checked at the output of the triplication decoder. If no errors are detected, the AudioLink can proceed to act as appropriate for the information received. If errors are detected, however, the AudioLink can proceed along one of two paths. One option is to ignore the corrupted message, take no further action, and begin listening again for subsequent messages. The alternative option is to attempt to salvage the message from each of the three data bands independently. For instance, suppose the signal was degraded severely in two of the three bands, but in one band the transmission was successful. The triplication decoder will produce an erroneous result, and the CRC will fail. However, if the data in each of the three bands are considered independently, the CRC will verify the correct digital signature in the clean band. This "second chance" approach increases the success rate for reception, but at the expense of a corresponding increase in the false detection rate. Investigations revealed that the largest problem for false detection was by far the all zero case, where an entire band was wiped out and the estimate of the received data was all zeros. The CRC bits for the all zero data stream are also all zeros, and so the CRC check will falsely indicate valid data in this case. Thus eliminating from consideration the all zero digital signature significantly reduced the false detection rate.

Section 3.1.6. Frequency Locking Mechanism

Since the playback and recording speeds of videotape machines are not perfectly constant, the time scale of the audio signal can expand and contract locally. As described in Section 2.3, the slow variation in the pitch of a sound reproduction resulting from the variations in the speed of recording or reproducing equipment is called "wow." Similarly a rapid variation is called "flutter." These phenomena have the effect of shifting all sinusoids in the frequency domain. A frequency locking mechanism was developed to detect, quantify, and compensate for the resulting spectral shifting. This locking mechanism consists of five non-uniformly spaced sinusoids placed at the upper end of the frequency region, and the amount of spectral shift is determined with a frequency domain matched filter. Positioning the locking sinusoids at the higher frequencies provides a more accurate estimate of frequency shift, since the amount of shift a sinusoid experiences is directly proportional to its frequency. Figure 3.11 shows the frequency domain matched filter. This matched filter is applied to the FFT bins in the region of the locking sinusoids. When the matched filter overlaps with the locations of the five sinusoids in the FFT a large correlation results, and thus the amount of frequency shift is revealed. The output of the matched filter during zero frequency shift is shown in Figure 3.12.

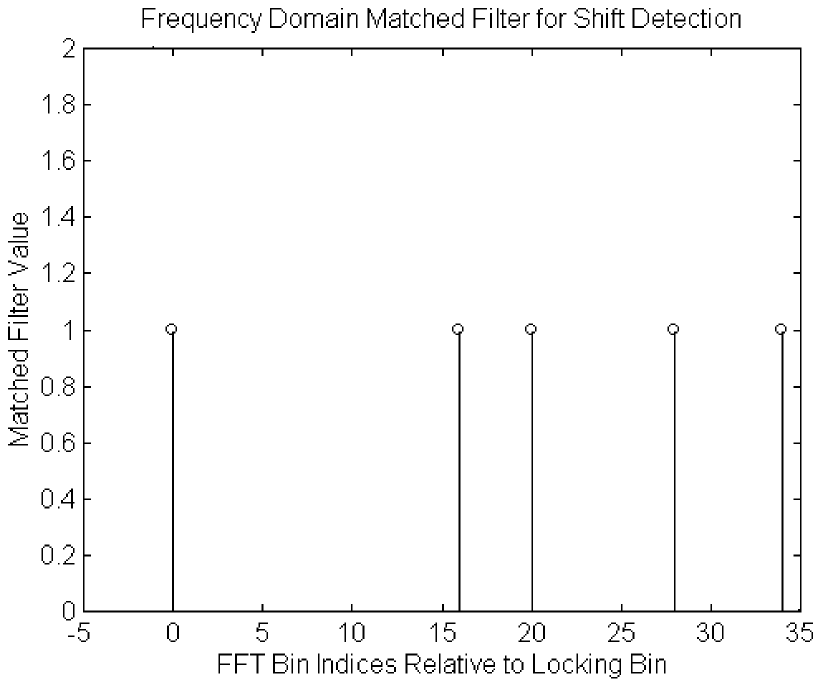

Figure 3.11 Frequency Domain Matched Filter for Frequency Shift Detection

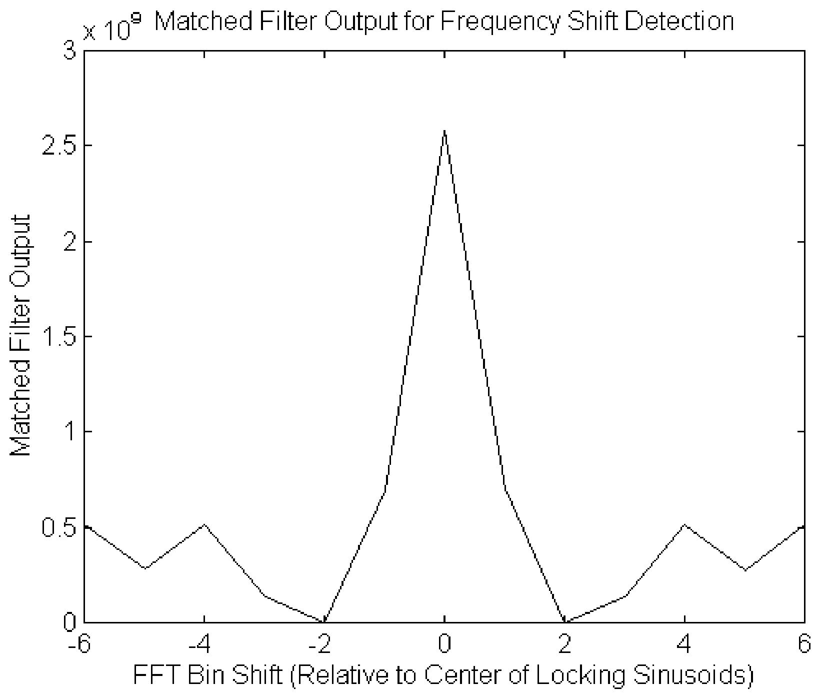

Figure 3.12 Matched Filter Output for Frequency Shift Detection

The spacing of the locking sinusoids was chosen to maximize the correlation during lock and to minimize the correlation for all other shifts. Since the frequency axis is distorted by scaling during wow and flutter, not only are the sinusoids displaced, but the spacing between the sinusoids expands and contracts as well. However, we limit the amount of relative distortion for which we attempt compensation. For those limited distortions, the change in the spacing between the sinusoids is relatively small (less than about 0.22 bins out of 35 bins). Furthermore the main lobes produced by the Hanning window are relatively wide and flat. Thus a sufficiently large correlation remains when the sinusoids and the matched filter overlap even when a shift has occurred.

The maximum ratio between the peak lock correlation to the peak non-lock correlation occurs when the sinusoids are located exactly on the nominal FFT bins during zero frequency shift. This maximum ratio is approximately 2.5 / 0.5 = 5 as shown in Figure 3.12 above. The worst case occurs when the frequency shift is near the maximum considered for compensation and the sinusoids are located between FFT bins. Simulations have revealed that the degradation in the ratio of peak lock correlation to peak non-lock correlation is about ten percent for the worst case scenario, and therefore the frequency shift is still easy to detect and quantify. Once the amount of shift is determined from the peak of the matched filter output, the new data and control bin locations can be read from lookup tables computed a priori. The data extraction then proceeds as discussed earlier.

The detection process is therefore multi-step. First the nominal (zero-shift) control sinusoid locations are examined to determine if the data sinusoid locations should be polled. Next the matched filter is applied and the output used to reveal how much frequency shift, if any, has occurred in the current signal block. If a possible frequency shift is indicated, new control bin locations corresponding to the amount of shift are found in a lookup table. These new control bins are examined to see if an improvement has occurred in the control function relative to the one based on the nominal locations. If neither the nominal nor the shifted locations produce a control function above the 2/3 threshold, no data is present in the block and we proceed to analyze the next signal block. If, however, one or both of the control bin sets produce a control function above the 2/3 threshold, valid data is assumed to be present in the signal block. The amount of frequency shift is determined by whichever of the two control bin location sets, the nominal or the shifted, produces the largest control function value. If the nominal control bin locations produced the largest result, the nominal data bin locations are used to extract the data bits. If the shifted control set produced the larger control function value, then new data bin locations are loaded from a lookup table and are used to extract the data.

Section 3.2. Code Insertion with Psychoacoustic Masking

Section 3.2.1. Masking Overview

To make the inserted digital signature as inaudible as possible psychoacoustic masking properties were employed. Psychoacoustic masking describes the insensitivity of the human auditory system to certain sounds when other, louder, sounds are present. In the presence of a given audio signal, the human ear’s ability to perceive other sounds is changed, both in time and in frequency. A louder sound (the masker) is said to mask the presence of a softer sound (the maskee). If the masker and maskee occur at the same time but contain different frequency components, the effect is called frequency masking. If the maskee occurs slightly before or slightly after the masker, the effects are called pre- and post-masking, respectively [22]. Combinations of temporal and frequency masking are also possible. Figure 3.13 below demonstrates the masking threshold due to the presence of a strong signal. As long as the smaller signals remain below the threshold, they will be imperceptible to human observers. Note that the masking effect decays more rapidly for frequencies below the masker than for frequencies above it. The temporal effect is similar, and masking persists longer in duration after the masker than before it.

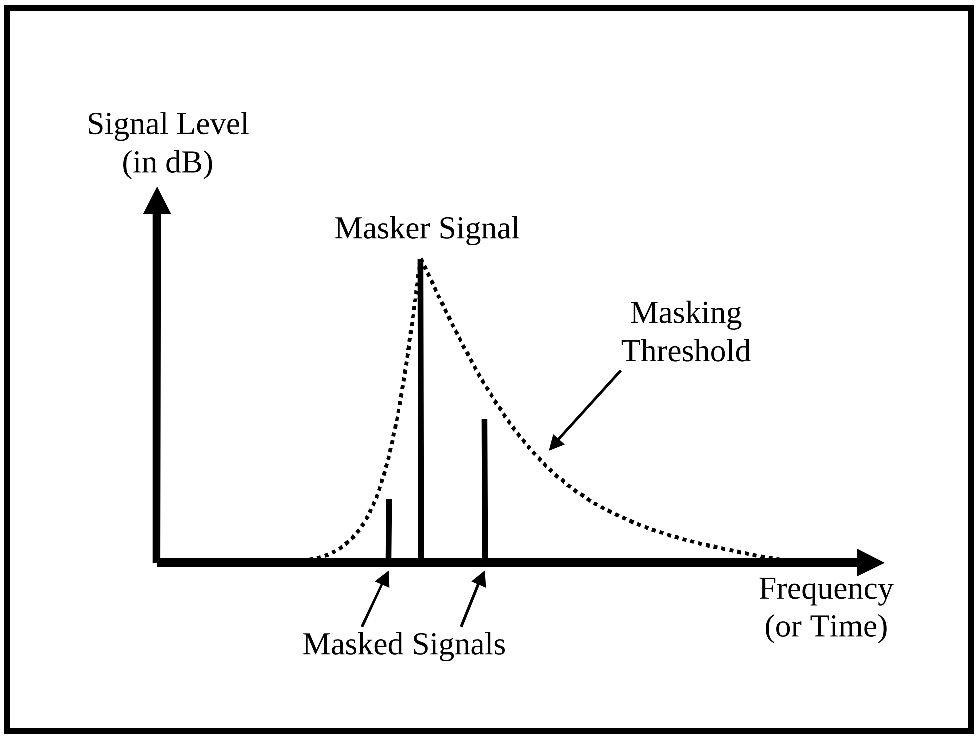

Figure 3.13 Psychoacoustic Masking

Psychoacoustic masking effects have been characterized empirically by subjective listening experiments and theoretically by studying the physiological structures of the human auditory system [35], [37]. For many years researchers have exploited masking properties in the application of audio coding [34], and several modern audio coding standards are founded on such principles [38], [39]. Although sufficient details have been presented here for the immediate application, the reader is encouraged to explore other sources for more information on this interesting topic [35] - [39].

Section 3.2.2. Code Insertion Using Psychoacoustic Masking

When inserting a digital signature into the audio of a television commercial, psychoacoustic masking effects are employed as much as possible. The signature is of short time duration and has low amplitude relative to the local audio. (In order to guarantee that at least one signal block at the receiver is completely contained within the duration of the code, the signature must have a duration of one block plus the number of refresh samples jumped between consecutive blocks. This corresponds to 4096 plus 205 samples, at a sampling rate of 16.0 kHz, or 268.8125 ms. In practice the signature duration is usually set to about 333 ms (1/3 second). This duration ensures that at least five blocks will be contained within the signature at the receiver.) Furthermore, the sinusoidal frequencies were chosen to be in the range from 2.4 to 6.4 kHz, where human sensitivity declines compared to its peak, which occurs around 1 kHz. This frequency range also allows the signature to be placed where strong low frequency content is present in the audio signal, to help mask the weaker high frequency sinusoids. Since humans are much more sensitive to the lower frequencies, this masking can be quite effective. The original audio content above 6.4 kHz, if preserved after filtering, also helps in masking the sinusoids. In addition, using frequencies above 2.4 kHz provides some resistance to human voice interference at the receiver. Note that the original audio content preserved for masking purposes is often wideband versus tonal in nature, and the sinusoids by definition are tones. Research has shown that broad-band noise is more effective at masking tones than tones are at masking noise [35], [36].

Although the audio component of a television signal is not bandlimited to 6.4 kHz, and frequencies above this could have been used to take advantage of further reduction in human sensitivity, the sampling rate of the decoder had to be kept as low as possible because of computational requirements. In addition, some equipment - such as older VCRs - produced attenuation at higher frequencies during laboratory tests. Such lowpass responses could attenuate the sinusoids and prevent successful reception if they were moved to higher frequencies.

Once the target location within the audio signal has been chosen, a zero-phase bandstop filter is used to locally remove any frequency content between 2.4 and 6.4 kHz. A 400th order finite impulse response (FIR) filter was designed by the window method [16] for this purpose. Exactly zero-phase distortion is obtained by applying the filter causally and non-causally. Such zero-phase distortion is important in preserving the quality of the audio signal, especially at the endpoints of the modified block. Figure 3.14 below shows the frequency response of the bandstop filter, and Figure 3.15 shows its impulse response. Note that the actual attenuation is double that shown in the figure, since the filter is applied twice - once in each time direction.

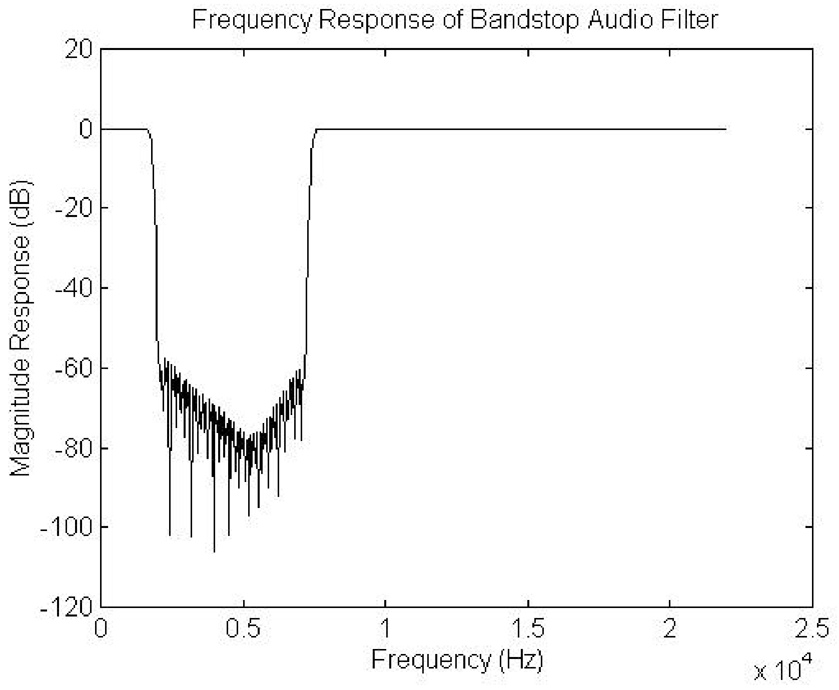

Figure 3.14 Frequency Response of Bandstop Audio Filter

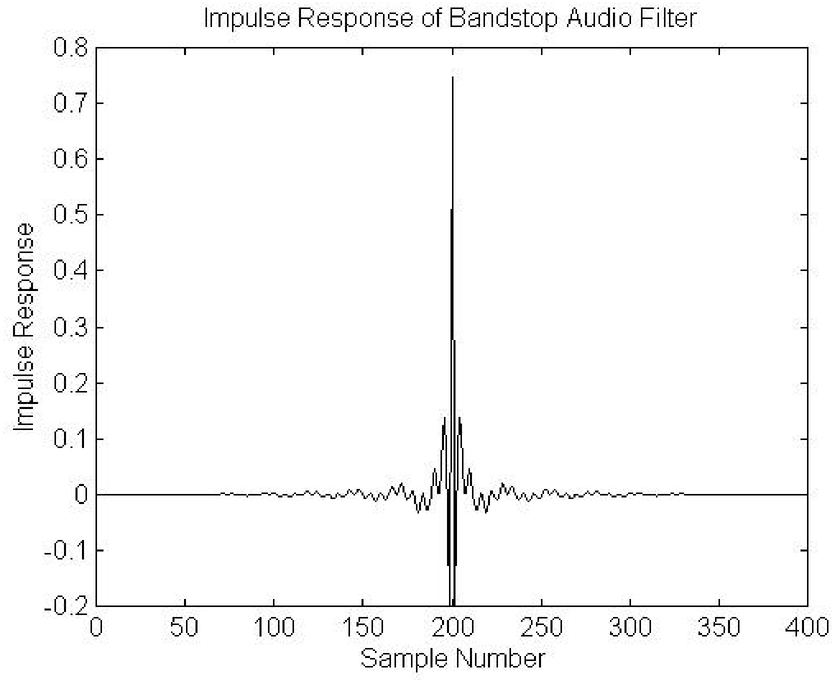

Figure 3.15 Impulse Response of Bandstop Audio Filter

Extra samples of the audio are extracted before and after the block to be modified to avoid filter transient effects in the block itself. When the modified block is inserted back into the original signal, the extra buffering samples on the ends are not replaced. Thus transient effects are avoided. Infinite impulse response (IIR) designs, especially the elliptic [11], were also considered for the bandstop filter. These filters had the benefits of a lower order and fewer computations for a given cutoff rate. However, the phase response is very non-linear, and the transient effects are infinite in duration (although they will decay to small values at some finite time). The transient effects of FIR filters terminate completely at M+1 samples, where M is the order of the filter. Furthermore FIR filters are easily constrained to have a linear phase response by forcing their coefficients to be symmetric. Although the non-linear phase response of the IIR would be cancelled by the bi-directional filtering, its transient effects could still cause noticeable distortion of the audio, especially at the endpoints of the block. Therefore, because of its phase response and in particular its finite transient duration, the FIR filter was the final choice.

Once the target location of the audio signal has been bandstop-filtered, the sinusoids are added. Figure 3.16 below shows the time-averaged power spectral density (PSD) of an example window of audio signal. Note the lowpass nature of the signal, which is typical for television audio. Figure 3.17 shows the PSD of the same window after the bandstop filtering and addition of the sinusoids.

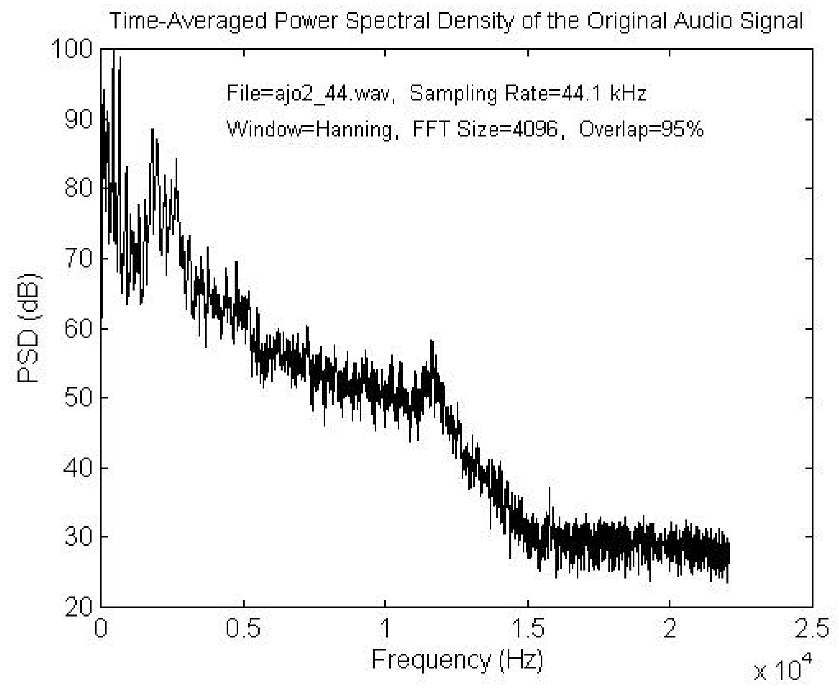

Figure 3.16 Time-Averaged Power Spectral Density of the Original Audio Signal

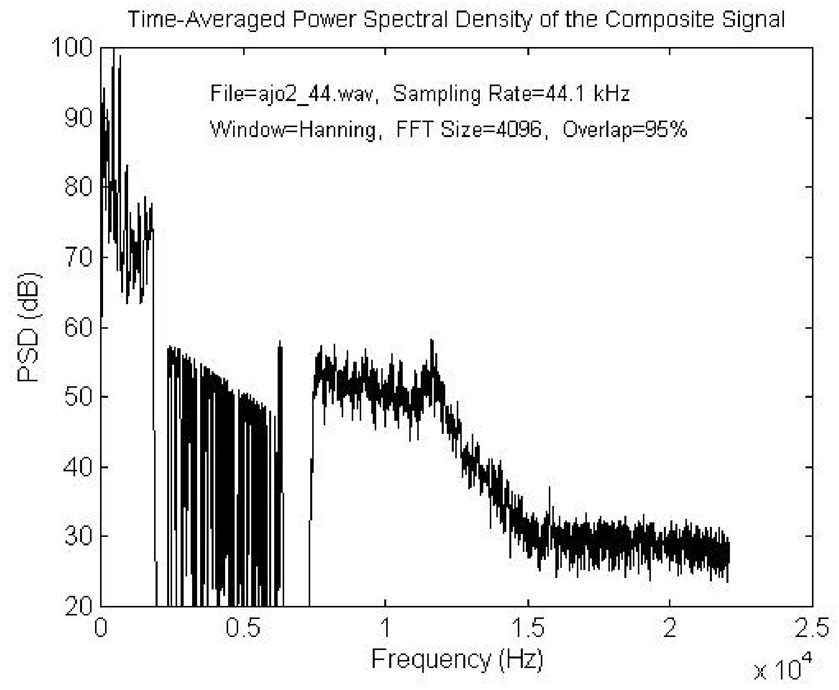

Figure 3.17 Time-Averaged Power Spectral Density of the Composite Signal

It is evident from Figure 3.16 and Figure 3.17 that the sinusoid amplitudes vary logarithmically to correspond roughly with the rolloff of the original audio spectrum in the region. This rolloff aids in the psychoacoustic masking. In order to set the amplitudes, especially in the final C++ program for code insertion, it was easiest to compute them based on FFT bin numbers. Given a start bin number x_1 and a stop bin number x_2, and a desired change in amplitude $\Delta_{dB} = y_2 - y_1$ in dB, we can compute the amplitudes for each sinusoid in the region (see Figure 3.18 below).

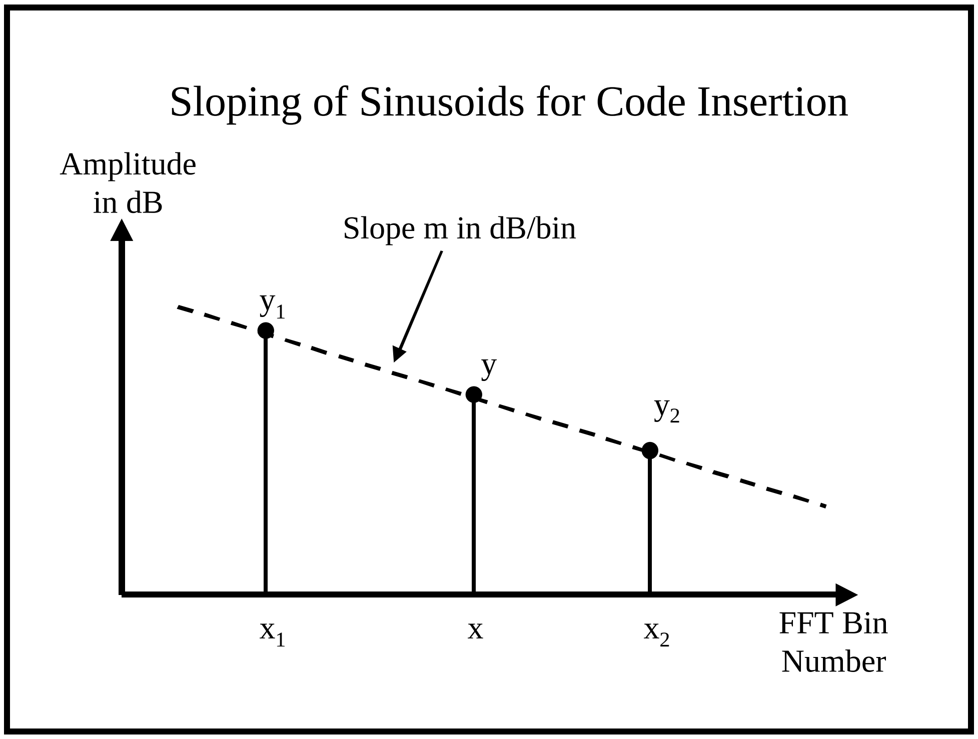

Figure 3.18 Sloping of Sinusoids for Code Insertion

Defining the slope in decibels as

$$m = \frac{(y_2 - y_1)}{(x_2 - x_1)} = \frac{\Delta_{dB}}{(x_2 - x_1)}$$

(3.4)

we can compute the amplitude *y* for any *x* in the region. The value for *y* in decibels will be given by

$$y_{dB} = 20\log_{10}(y) = m(x - x_1)$$

(3.5)

and so by substitution the final value for *y* is computed by

$$y = 10^{\frac{y_{dB}}{20}} = 10^{\frac{m(x-x_1)}{20}} = 10^{\frac{\frac{\Delta_{dB}}{(x_2-x_1)}(x-x_1)}{20}}$$

(3.6)

Only the relative and not the absolute amplitudes matter, since all sinusoids will later be scaled to achieve a desired volume level relative to the filtered audio signal. Because the locking sinusoids are critical for successful reception during frequency shifts, their amplitudes are not varied and are instead set to the maximum of all of the sinusoids. Their protuberance from the weaker sinusoids at the high frequency end is evident in Figure 3.17 above.

In order to achieve a desired analog-to-digital (power) ratio (ADR), the power in the sum of the sinusoids is computed, as well as the power in the filtered audio signal. These powers are then used to compute a scale factor for the sinusoids, so when the two are mixed the desired ADR is achieved.

Section 3.2.3. Practical Notes Regarding Code Insertion

When choosing a target temporal location for the buried code in an audio signal, we search for areas where the audio has promising psychoacoustic characteristics. For example, locations where strong low frequency content is present afford greater masking ability. Also, if the audio is changing rapidly the human auditory system is less likely to discern a hidden code. In contrast, if the audio is constant or slowly varying, the inserted code will be perceived as disruptive. Spectrograms are very useful tools in finding such candidate insertion locations. Figure 3.19 below shows an example spectrogram of a section of a television commercial, and Figure 3.20 shows the result after a code has been inserted. The sinusoids are evident between 2.7 and 3.0 seconds, in the frequency range between 2.4 and 6.4 kHz. For the spectrograms, a 2048 point FFT with a Hanning window was applied to the signal, with fifty percent overlap in the FFT blocks. Since the sampling rate was 44.1 kHz, the frequency resolution of the plot is 21.53 Hz and the time resolution is 46.44 ms. Successive signal blocks were separated by 23.22 ms because of the chosen overlap. In Figure 3.20 the deep blue horizontal bands at the code insertion point are the valleys on each side of the code created by the bandstop filtering. Figure 3.17 above also reveals these valleys in the frequency domain. Note that there is relatively strong low frequency content present at the code location point. Furthermore, strong frequency content between 2.4 and 6.4 kHz exists just prior to the insertion. Both of these factors aided in the psychoacoustic masking for this example.

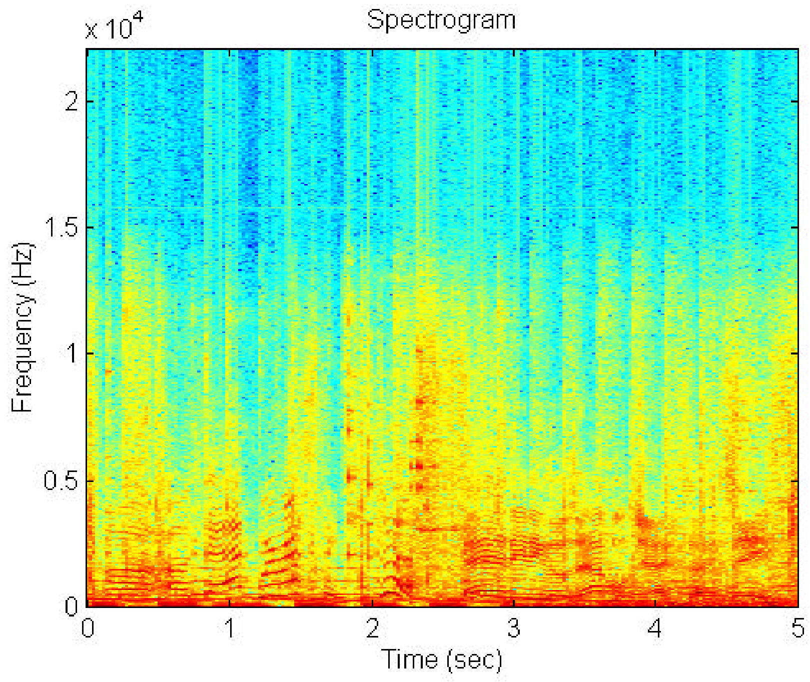

Figure 3.19 Spectrogram of Original Audio Signal

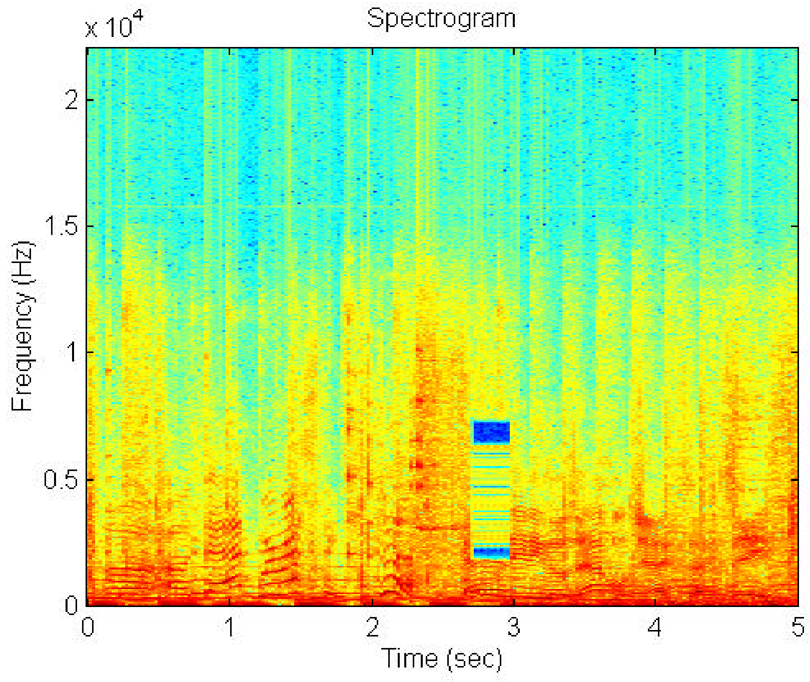

Figure 3.20 Spectrogram of Modified Audio Signal

Unfortunately, seemingly perfect locations often fail to hide the codes successfully, and the insertion quality must still be verified by trial and error at the candidate locations. Therefore, establishing criteria and a measure for potential "hideability" are topics for future research.

Several other factors also affect how well the codes are hidden. If observers are explicitly listening for the code, they might be able to hear it, but the code presence is not considered objectionable even in this case. Observers are more likely to be able to identify the code if they are young with nearly perfect hearing, and if they already know what it sounds like. Since the AudioLink codes are imbedded into the audio of television programming, commercials in particular, the television video also serves as a distraction that assists concealment. One option that is appropriate in some circumstances is to insert the code into a prompting sound, such as a buzzer or beep. The addition of the code then simply changes the nature of the prompt slightly, creating a new buzz or beep. This option is notably useful in interactive game applications. In fact, the raw code itself with no masking audio at all has been used as a prompt for an interactive trivial pursuit game (see Section 7.3).

When attempting concealment, the relative volume level for insertion must be chosen carefully. If the relative volume is set too high, the code is easily heard. But setting the volume too low has drawbacks other than the obvious reduced reception capability. Practice has revealed that if the code volume is too low, observers can detect a "hole" in the audio produced by the bandstop filtering. The code seems to act as a partial replacement for the original audio content in the frequency region. Here too, as in the temporal location selection, finding a good insertion level requires trial and error, and a more automated approach is left as a suggestion for future research.