Sample Rate Conversion

David MacMahon, September 2014

Overview

Changing the sample rate of a sampled signal can be done using FIR filters when the ratio of the input and output sample rates can be expressed as a ratio of integers. For example, converting from $5f$ to $3f$ results in an output sample rate that is $3/5$ of the input sample rate. Generally, the ratio of output sample rate over input sample rate is expressed as $L/M$.

Simplistic Approach

Changing the sample rate by $L/M$ can be thought of as two steps: increasing the sample rate by $L$ then decreasing the sample rate by $M$. The trivial way to do this is to up-sample by $L$ then down-sample by $M$ as shown in these two examples of a $3/5$ sample rate conversion:

Original : |A D G J M |
Upsampled : |A00D00G00J00M00| (zero stuffing)
Downsampled : |A 0 0 | (decimation)

Original : |A D G J M |
Upsampled : |AAADDGGGJJJMMM| (sample replication)
Downsampled : |A D J | (decimation)

These examples show two different techniques for up-sampling: zero stuffing and sample replication. Both examples produce 3 output samples for 5 input samples as expected from a $3/5$ sample rate conversion, but there are clear problems with both. The zero-stuffing method simply outputs every fifth input sample with two zero samples inserted between (effectively down-sampling by $M$ and then up-sampling by $L$!). The sample replication method uses more but not all of the input samples, so it grossly distorts the signal.

Another way of thinking about these problems is in the frequency domain. Both up-sampling methods increase the bandwidth of the original band-limited signal by a factor of $L$. This additional bandwidth is not devoid of power; it contains spectral content replicated from the original band-limited spectrum. The trivial down-sampling method of taking every $M$th sample will alias this additional spectral content into the output. This unwanted spectral content corresponds to the time domain distortions evident in the examples above.

Filtered Approach

To avoid the unwanted spectral content, the up-sampled signal can be low pass filtered before down-sampling. This is shown in the following example:

Original : |A D G J M |
Upsampled : |A00D00G00J00M00| (zero stuffing)
Filtered : |ABCDEFGHIJKLMNO| (low-pass to $1/M$)
Downsampled : |A F K | (decimation)

The low pass filter operates at the up-sampled sample rate, i.e. $L$ times in input sample rate. The filter is designed to pass $1/M$ of the up-sampled band so that the filter output signal is essentially $M$ times oversampled. Taking every $M$th sample will alias in whatever spectral content exists in the filter’s stop band. In the ideal case, the filter passes no power in the stop band. Realizable filters will pass some power in the stop band, but to be useful it should not pass more than can be tolerated by the application.
Filter Considerations

General FIR Filter

The filter used for sample rate conversion can be of any design that meets the requirements. The most commonly used filter design for sample rate conversion is the finite impulse response (FIR) filter. A FIR filter is described by the following expression:

\[ y[n] = \sum_{i=0}^{N-1} b[i] \cdot x[n - i] \]

Where:

- \( x[n] \) represents the input signal
- \( y[n] \) represents the output signal
- \( b[i] \) represents the \( N \) coefficients (taps) of the order \( N + 1 \) FIR filter

Filtering an Up-sampled Signal

For the purposes of sample rate conversion, the signal being filtered has been up-sampled, so many of the \( x[n] \) samples are zero. The summation shown above only needs to be evaluated for non-zero samples of \( x[n] \), which occur whenever \((n - i) \mod L = 0\) or, equivalently, whenever \( i = (n \mod L) + kL \). If the length of the filter is constrained to \( N = KL \), then above summation can be written as the summation of only \( K \) terms:

\[ y[n] = \sum_{k=0}^{K-1} b[(n \mod L) + kL] \cdot x[n - (n \mod L) - kL] \]

This means that for each output sample, only \( K \) filter coefficients are used. In other words, even though the filter has \( KL \) taps, only \( K \) of them need to be implemented.

The input sample (i.e. the \( x \) factors) in the above expression can be expressed in terms of the pre-up-sampled sequence \( x_{in} \):

\[ y[n] = \sum_{k=0}^{K-1} b[(n \mod L) + kL] \cdot x_{in}[(n/L) - k] \]

Downsampling the Filtered Signal

Because the filter output will be downsampled by a factor of \( M \), the output sequence only needs to be evaluated for \( n = mM \), where \( m \) is an integer. The downsampled output sequence \( y_{out}[m] \) becomes:

\[ y_{out}[m] = \sum_{k=0}^{K-1} b[(mM \mod L) + kL] \cdot x_{in}[(mM/L) - k] \]

This is the general expression for an \( L/M \) sample rate conversion filter.
Worked Example

This shows the expansion of the above expression for the case $L = 3$, $M = 5$, and $K = 4$. In this case, the low pass filter will have $KL = 12$ taps and a normalized cutoff frequency of $1/M$. The method of deriving suitable filter coefficients is outside the scope of this document.

\[
y[0] = \sum_{k=0}^{3} b[0 + 3k] \cdot x[0 - k]
\]

\[
y[1] = \sum_{k=0}^{3} b[2 + 3k] \cdot x[1 - k]
\]

\[
y[2] = \sum_{k=0}^{3} b[1 + 3k] \cdot x[3 - k]
\]

\[
y[3] = \sum_{k=0}^{3} b[0 + 3k] \cdot x[5 - k]
\]

\[
y[4] = \sum_{k=0}^{3} b[2 + 3k] \cdot x[6 - k]
\]

\[
y[5] = \sum_{k=0}^{3} b[1 + 3k] \cdot x[8 - k]
\]

...:

Expanded:

\[
y[0] = b[0] \cdot x[0] + b[3] \cdot x[-1] + b[6] \cdot x[-2] + b[9] \cdot x[-3]
\]

\[
\]

\[
\]

\[
\]

\[
\]

\[
\]

...: