Lecture 02: Multirate Processing

In many practical applications, a sampling rate needs to be converted (either increased or decreased). For instance, a sampling rate of 44.1 kHz is used in audio CD, while video DVD (and DVD audio) format assumes audio signals sampled at 48 kHz. To make an audio CD of a movie sound-track, decrease of sampling rate is needed.

Another example is conversion of composite video signals from NTSC (sampling rate of 14.3181818 MHz) to PAL (sampling rate of 17.734475 MHz) and back. Digital component video signal is sampled at 13.5 MHZ and 6.75 MHz for the luminance and chrominance components respectively, which requires rete conversion too...

Sampling Rate conversion can be done by:
1. D/A conversion followed by A/D conversion at a different rate – advantage: arbitrary sampling rate
   disadvantage: distortion during A/D and quantization noise during D/A;
2. Sampling rate conversion in digital domain – subject of our discussion;

Material comes from Proakis mostly
Introduction

Terminology:
Decimation = downsampling
Interpolation = upsampling

Sampling rate conversion can be viewed as a linear operation:

\[ x(n) \xrightarrow{\text{Linear filter } h(n,m)} y(m) \]

\[ \text{Rate } F_x = \frac{1}{T_x} \quad F_y = \frac{1}{T_y} \]

\( T_x \) and \( T_y \) are the corresponding sampling intervals for \( x_n \) and \( y_m \).

Constrain:

\[
\frac{F_y}{F_x} = \frac{I}{D} \quad (2.3.1)
\]

Where \( I \) and \( D \) are the integer upsampling and downsampling factors.

Introduction

Sampling rate conversion can also be viewed as resampling of the same analog signal.

Thus, obtaining \( y_m \) from \( x_n \) is equivalent to sampling \( x(t) \) at the other sampling rate. \( y_m \) is a time-shifted version of \( x_n \). Such a shift can be done by a linear filter with a flat magnitude and linear phase responses: i.e., with a frequency response of \( \exp(-j\omega \tau_i) \), where \( \tau_i \) is the time delay. Not equal sampling rates imply that the amount of time shift vary from sample to sample.

Therefore, a rate converter can be implemented with a set of linear filters having the same flat magnitude response but generating different time delays.
Decimation by a factor $D$

A signal $x_n$ with a spectrum $X(e^{j\omega})$ needs to be downsampled by an integer factor $D$.

A signal is band limited to $0 \leq |\omega| \leq \pi$ (2.5.1)

Simple selecting every $D^{th}$ sample of $x_n$ would result in aliasing in the frequency domain with a folding frequency $F_s/2D$.

Therefore, to avoid aliasing, the bandwidth should be reduced to $F_{\text{max}} = F_s/2D$ or to $\omega_{\text{max}} = \frac{\pi}{D}$ (2.5.2)

The filter output $v_n$ is

$$v_n = \sum_{k=0}^{\infty} h_k x_{n-k}$$ (2.6.2)

Which is downsampled by the factor $D$ to produce $y_m$:

$$y_m = v_{mD} = \sum_{k=0}^{\infty} h_k x_{mD-k}$$ (2.6.3)
Decimation by a factor $D$

The filtering operation on $x_n$ is linear and time invariant. However, the downsampling results in a time variant system: $x_{n-n_0}$ does not imply $y_{n-n_0}$ unless $n_0$ is a multiple of $D$.

The output in the z-domain:

$$Y(z) = \sum_{m=-\infty}^{\infty} y_m z^{-m} = \sum_{m=-\infty}^{\infty} v_m z^{-m/D}$$

$$= \sum_{m=-\infty}^{\infty} v_m \left[ \frac{1}{D} \sum_{k=0}^{D-1} e^{j \frac{2\pi mk}{D}} \right] z^{-m/D} = \frac{1}{D} \sum_{k=0}^{D-1} \sum_{m=-\infty}^{\infty} v_m \left[ e^{j \frac{2\pi mk}{D}} z^{-D} \right]$$

$$= \frac{1}{D} \sum_{k=0}^{D-1} \left( e^{j \frac{2\pi mk}{D}} \frac{1}{z^D} \right) = \frac{1}{D} \sum_{k=0}^{D-1} H_D \left( e^{j \frac{2\pi mk}{D}} z^{-D} \right) X \left( e^{j \frac{2\pi mk}{D}} \frac{1}{z^D} \right)$$

(2.7.1)

The spectrum of the output can be found by evaluating $Y(z)$ on the unit circle.

Decimation by a factor $D$

The output sampling rate is $F_y$. Therefore, the output frequency variable is

$$\omega_y = \frac{2\pi F}{F_y} = 2\pi FT_y$$

(2.8.1)

Since the output and input sampling rates are related as

$$F_y = \frac{F_x}{D}$$

(2.8.2)

thus, the two frequency variables (output and input)

$$\omega_x = \frac{2\pi F}{F_x} = 2\pi FT_x$$

(2.8.3)

are related as

$$\omega_y = D\omega_x$$

(2.8.4)
Decimation by a factor $D$

Therefore, the frequency range $0 \leq |\omega_x| \leq \pi / D$ is stretched into the frequency range $0 \leq |\omega_y| \leq \pi$ by the downsampling process.

The output spectrum is

$$Y(\omega_y) = \frac{1}{D} \sum_{k=0}^{D-1} H_D \left( \frac{\omega_x - 2\pi k}{D} \right) X \left( \frac{\omega_x - 2\pi k}{D} \right)$$

$$= \frac{1}{D} H_D \left( \frac{\omega_x}{D} \right) X \left( \frac{\omega_x}{D} \right) = \frac{1}{D} X \left( \frac{\omega_x}{D} \right)$$

(2.9.1)

Decimation by a factor $D$

Assuming ideal LPF:

<table>
<thead>
<tr>
<th>$X(\omega_x)$</th>
<th>$Y(\omega_y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| $|H(\omega_x)|$ | $|Y(\omega_y)|$ |
|----------------|---------------|
|               |               |

<table>
<thead>
<tr>
<th>$\omega_x$</th>
<th>$\omega_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-\pi / D$</td>
<td>$0$</td>
</tr>
<tr>
<td>$0$</td>
<td>$\pi / D$</td>
</tr>
<tr>
<td>$\pi / D$</td>
<td>$\pi$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\omega_x$</th>
<th>$\omega_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-\pi / D$</td>
<td>$0$</td>
</tr>
<tr>
<td>$0$</td>
<td>$\pi$</td>
</tr>
<tr>
<td>$\pi / D$</td>
<td>$\pi$</td>
</tr>
</tbody>
</table>
**Interpolation by a factor $I$**

An increase in the sampling rate by an integer factor $I$ can be achieved by interpolating $I-1$ new samples between successive values of the signal. We are interested in the process preserving the spectral shape of the signal $x_n$.

Let $v_m$ is the sequence obtained from a signal $x_n$ by adding $I-1$ zeros between successive values of $x_n$ and, therefore, with a rate $F_y = IF_x$.

$$v_m = \begin{cases} 
    x_{m/I}, & m = 0, \pm I, \pm 2I, \ldots \\
    0, & \text{otherwise}
\end{cases} \quad (2.11.1)$$

Its sampling rate is identical to the rate of $y_m$ and its z-transform is

$$V(z) = \sum_{m=-\infty}^{\infty} v_m z^{-m} = \sum_{m=-\infty}^{\infty} x_m z^{-mI} X(z^I) \quad (2.11.2)$$

Evaluating on the unit circle, the spectrum is

$$V(\omega_y) = X(\omega_y I) \quad (2.11.3)$$

---

**Interpolation by a factor $I$**

The new frequency variable is

$$\omega_y = 2\pi \frac{F}{F_y} = 2\pi \frac{F}{IF_x} = \frac{\omega_x}{I} \quad (2.12.1)$$

The sampling rate increase by addition of $I-1$ zeros between the values of $x_n$ results in a signal whose spectrum is an $I$-fold periodic repetition of the input signal spectrum $X(\omega)$. 

---

11

---

12

---
Interpolation by a factor $I$

Since only the frequency components of $x_n$ in the range $[0, \pi/I]$ are unique, the images of $X(\omega)$ above $\omega_I = \pi/I$ should be rejected by a LPF with the frequency response

$$H_I(\omega_I) = \begin{cases} C, & 0 \leq |\omega_I| \leq \pi/I \\ 0, & \text{otherwise} \end{cases} \quad (2.13.1)$$

Where $C$ is a scale factor required to normalize the output $y_m$. Therefore, the output spectrum is

$$Y(\omega_I) = \begin{cases} CX(\omega_I), & 0 \leq |\omega_I| \leq \pi/I \\ 0, & \text{otherwise} \end{cases} \quad (2.13.2)$$

The scale factor $C$ is selected so that the output $y_m = x_{mI}$ for $m = 0, \pm I, \pm 2I, \ldots$. For $m = 0$

$$y_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} Y(\omega_I) d\omega_I = C \frac{\pi/I}{2\pi} \int_{-\pi/I}^{\pi/I} X(\omega_I) d\omega_I = C \frac{1}{I} \int_{-\pi}^{\pi} X(\omega) d\omega = C \frac{1}{I} x_0 \quad (2.13.3)$$

Therefore, $C = I$ is the needed normalization factor.

Finally, the output sequence is a convolution of the $v_n$ sequence and the LPF’s unit pulse response:

$$y_m = \sum_{k=-\infty}^{\infty} h_{m-k} v_k \quad (2.14.1)$$

Since $v_k = 0$ except at multiples of $I$, where $v_{kI} = x_k$

$$y_m = \sum_{k=-\infty}^{\infty} h_{m-Ik} x_k \quad (2.14.2)$$
Rate conversion by a factor $I/D$

Sampling rate conversion by a rational factor $I/D$ can be achieved by performing the interpolation by a factor $I$ first and then the decimating the output by the factor of $D$.

Performing interpolation followed by decimation allows to preserve the desired spectral characteristics of $x_n$. Also, since two filters $h_u$ and $h_d$ are working at the same rate, they can be combined into a single LPF with the characteristic:

$$H(\omega) = \begin{cases} I, & 0 \leq |\omega| \leq \min \left( \frac{\pi}{D}, \frac{\pi}{I} \right) \\ 0, & \text{otherwise} \end{cases} \quad (2.15.1)$$

Rate conversion by a factor $I/D$

where

$$\omega_v = 2\pi F/F_v = 2\pi F/IF_x = \omega_x/I \quad (2.16.1)$$

The output of the system is

$$y_m = \sum_{n=-\infty}^{\infty} h(mD - \left\lfloor \frac{mD}{I} \right\rfloor I + nI) x \left( \left\lfloor \frac{mD}{I} \right\rfloor - n \right) \quad (2.16.2)$$
Rate conversion by a factor $I/D$

In the frequency domain:

\[ Y(\omega_y) = \frac{1}{D} \sum_{k=0}^{D-1} \left( \frac{\omega_y - 2\pi k}{D} \right) \]  

Here

\[ \omega_y = D\omega_v \]  

and for the alias-free situation

\[ Y(\omega_y) = \begin{cases} \frac{I}{D} X \left( \frac{\omega_y}{D} \right), & 0 \leq |\omega_y| \leq \min \left( \pi, \frac{\pi D}{I} \right) \\ 0, & \text{otherwise} \end{cases} \]  

Filter design and implementation

1. Direct-form FIR filter structure

FIR filter in sampling rate conversion by factor $I/D$ can be designed by any FIR techniques. The structure is simple but inefficient: upsampling (for large $I$) leads to many samples input to the LPF being zeros; downsampling preserves only part of filter output.
Direct-form structure of down-sampler: filter works at a high sampling rate

More efficient structure of down-sampler: filter works at a low sampling rate

An FIR with a linear phase and, therefore, filter coefficients are symmetric. The efficient structure of down-sampler: filter works at a low sampling rate and only a half of filter coefficients are implemented.
Filter design and implementation

Direct-form realization of interpolator: filter works at a high sampling rate.

Filter design and implementation

Transposed form of FIR filter

Up-sampler embedded within the FIR filter

Filter works at a low rate – more efficient
Filter design and implementation

The transpose of a decimator is an interpolator and the transpose of an interpolator is a decimator.

It is possible to build an efficient interpolator using a symmetry property similarly to a decimator seen

2. Polyphase filter structure

Considering the upsampling process that inserts \( I - 1 \) zeros between samples, only \( K \) out of \( M \) input values stored in the FIR filter at any one time are nonzero. At some time instance, these nonzero values are multiplied by the filter coefficients \( h_0, h_1, h_2, \ldots, h_{M-I} \). In the following time instant, the nonzero values are multiplied by the coefficients \( h_1, h_{I+1}, h_{2I+1}, \ldots, h_{M-I+1} \) and so on. Therefore, we define a set of small polyphase filters defined by

\[
p_{k,n} = h_{k+nl}, \quad k = 0, 1, \ldots, I-1; \quad n = 0, 1, \ldots, K-1
\]

(2.24.1)

where an integer

\[
K = \frac{M}{I}
\]

(2.24.2)

The commutator rotates counterclockwise.
Filter design and implementation

Polyphase filters work at the low sampling rate $F_s$ and the rate conversion results from the fact that $I$ output samples are generated, one from each of the filters and for each input sample.

The polyphase structure is consistent with the observation that the rate conversion is achieved by filtering the input sequence by a time-variant filter

$$g_{n,m} = h_{nI+(mD)}$$

(2.25.1)

We note that $g_{n,m}$ varies periodically with period $I$. Consequently, a different set of coefficients is used to generate the set of $I$ output samples $y_m$.

Also, $p_{k,n}$ is obtained from $h_n$ by decimation with a factor $I$. Therefore, if the original filter frequency response is flat over $0 \leq |\omega| \leq \pi/I$, each polyphase filter will have a (relatively) flat response over $0 \leq |\omega| \leq \pi$. I.e., the polyphase filters are all-pass filters and differ by their phase characteristics.

The polyphase filter can also be viewed as a set of $I$ subfilters connected to a common delay line. Their frequency responses are

$$p_{k,n} = e^{i\omega k/I}$$

(2.25.2)

Filter design and implementation

Transposing the polyphase structure for the interpolating filter, we obtain a decimating polyphase filter. The unit pulse responses of the polyphase filters are

$$p_{k,n} = h_{nD-k}$$

(2.26.1)

$k = 0,1,...,D-1; n = 0,1,...,K-1$

where an integer

$$K = \frac{M}{D}$$

(2.26.2)

For a clockwise rotation, polyphase filters are

$$p_{k,n} = h_{nD-k}$$

(2.26.3)

$$k = 0,1,...,D-1$$

and

$$p_{k,n} = h_{nI-k}$$

(2.26.4)

$k = 0,1,...,I-1$
Filter design and implementation

3. Time-variant filter structure

Sampling rate conversion by a factor $I/D$ can be achieved in general by a linear time-variant filter with the response

$$g_{n,m} = h_{nI + mD}$$

(2.25.1)

where $h_n$ is the impulse response of the LP FIR filter. Selecting the filter length as $I$ and since $g_{n,m}$ is periodic with period $I$, the filter output may be expressed as

$$y_m = \sum_{n=0}^{K-1} g_{n,m} \frac{m}{I} x_{mD + n}$$

(2.25.2)

These computations can be performed by processing blocks of data of length $K$ by a set of $K$ filter coefficients $g_{n,m}$. There are $I$ such sets of coefficients, one set for each block of $I$ output points of $y_m$. For each block of $I$ output points, there is a corresponding block of $D$ input points of $x_n$ entering in the computation.

Filter design and implementation

Block processing algorithm can be implemented by the structure:

$D$ input samples are buffered and shifted into a second buffer of length $K$ one sample at a time. For each output sample $y_m$, the samples from the second buffer are multiplied by the corresponding set of filter coefficients $g_{n,m}$ for $n = 0, 1, \ldots, K-1$, and the $K$ products are accumulated to form $y_m$ for $m = 0, 1, \ldots, I-1$.

Therefore, $I$ outputs result from this computation, which is repeated for a new set of $D$ input samples…
Filter design and implementation

Alternatively, block processing algorithm can be implemented by an FIR filter with periodically varying structure:

Input samples of \( x_n \) are passed into a shift register operating at the sampling rate \( F_s \) and is of length \( K=M/I \), where \( M \) is the length of the time-invariant FIR filter with the frequency response

\[
H(\omega) = \begin{cases} 
I & \text{if } 0 \leq |\omega| \leq \min\left(\frac{\pi}{D}, \frac{\pi}{I}\right) \\
0 & \text{otherwise}
\end{cases}
\] (2.29.1)

Each stage of the register is connected to a hold-and-sample device coupling the input sampling rate \( F_s \) to the output sampling rate \((V/D)F_s\). The sample at the input of each hold-and-sample device is held until the next input sample arrives and the discarded. The output samples of the hold-and-sample device are taken at times \( mD/I, m=0,1,2,... \). When both the input and the output sampling times coincide, the input to the hold-and-sample is changed first and then the output samples the new input.

The \( K \) outputs from the \( K \) hold-and-sample devices are multiplied by the periodically time-varying coefficients \( g_{n,m} \) and the resulting products form \( y_m \).
Multistage implementation of sampling rate conversion

In many situations, decimation/interpolation with large factors may be needed. Traditional implementations would be inefficient.

Assuming that large interpolation and decimation factors \((I >> 1; D >> 1)\) can be factored into products of positive integers:

\[
I = \prod_{i=1}^{L} I_i \quad \text{and} \quad D = \prod_{i=1}^{J} D_i \tag{2.31.1}
\]

interpolation can be performed by a cascade of \(L\) stages of interpolation and filtering. The filter in each of the interpolators eliminates the images introduced by the up-sampling in the corresponding interpolator.

Similarly, decimation by a factor \(D\) can be achieved by cascading \(J\) stages of filtering and decimation.

The sampling rate at the output of the \(i^{th}\) stage is

\[
F_i = \frac{F_i}{D_i}, \quad i = 1, 2, \ldots, J \tag{2.32.1}
\]

The input sampling rate for the sequence \(x_n\) is \(F_0 = F_x\).

To avoid aliasing, filters in each stage must be selected to avoid aliasing within the overall frequency band of interest.

If the desired passband and transition band of the overall decimator are:

\[
\begin{align*}
\text{Passband:} & \quad 0 \leq F \leq F_{pc} \\
\text{Transition band:} & \quad F_{pc} \leq F \leq F_{sc}
\end{align*}
\tag{2.32.2}
\]
Multistage implementation of sampling rate conversion

Here $F_{sc} \leq F_{r}/2D$. Therefore, the aliasing in the band $0 \leq F \leq F_{sc}$ is avoided by selecting the frequency bands for each filter stage as

\[
\begin{align*}
\text{Passband} : & \quad 0 \leq F \leq F_{pc} \\
\text{Transition band} : & \quad F_{pc} \leq F \leq F_{1} - F_{sc} \\ 
\text{Stopband} : & \quad F_{1} - F_{sc} \leq F \leq F_{i-1}/2
\end{align*}
\]

For example, in the first filter stage: $F_{1} = F_{r}/D_{1}$, and the filter will have

\[
\begin{align*}
\text{Passband} : & \quad 0 \leq F \leq F_{pc} \\
\text{Transition band} : & \quad F_{pc} \leq F \leq F_{1} - F_{sc} \\
\text{Stopband} : & \quad F_{1} - F_{sc} \leq F \leq F_{0}/2
\end{align*}
\]

After decimation by $D_{1}$, aliasing from the signal components falling in the filter transition band occurs. However, this aliasing will be at frequencies above $F_{sc}$. Therefore, there is no aliasing in the frequency band $0 \leq F \leq F_{sc}$.

Multistage implementation of sampling rate conversion

For the multistage interpolator, the sampling rate at the output of the $i^{th}$ stage is

\[
F_{i-1} = I_{i}F_{i}, \quad i = J, J-1, \ldots, 1
\]

(2.34.1)

The output sampling rate is

\[
F_{0} = IF_{J}
\]

(2.34.2)

when the input sampling rate is $F_{p}$. The corresponding frequency band specifications are

\[
\begin{align*}
\text{Passband} : & \quad 0 \leq F \leq F_{p} \\
\text{Transition band} : & \quad F_{p} \leq F \leq F_{1} - F_{sc}
\end{align*}
\]

(2.34.3)
Sampling rate conversion of bandpass signals

Quite frequently (especially in communication) bandpass signals need to be re-sampled. Assuming that $F_c \gg B$ (central frequency much larger than the bandwidth), bandpass signals have equivalent lowpass representation that can be obtained by frequency translation of $F_c$ (usually, the center frequency).

An analog bandpass signal may be represented as:

$$x(t) = A(t) \cos(2\pi F_c t + \phi(t)) = A(t) \cos \phi(t) \cos 2\pi F_c t - A(t) \sin \phi(t) \sin 2\pi F_c t$$

$$= u_c(t) \cos 2\pi F_c t - u_s(t) \sin 2\pi F_c t = \text{Re} \{u(t)e^{j2\pi F_c t}\} \quad (2.36.1)$$

where

$$u_c(t) = A(t) \cos \phi(t) \quad (2.36.1)$$
$$u_s(t) = A(t) \sin \phi(t) \quad (2.36.2)$$
$$u(t) = u_c(t) + ju_s(t) \quad (2.36.3)$$

$A(t)$ is the amplitude (envelope) of the signal, $\phi(t)$ is the phase, and $u_c(t)$ and $u_s(t)$ are the quadrature components of the signal.

To translate $x(t)$ to its lowpass equivalent, it needs to be multiplied (mixed) by the quadrature carriers $\cos 2\pi F_c t$ and $\sin 2\pi F_c t$ and lowpass filtering the two products to eliminate the double-frequency terms (components concentrated around $2F_c$).

All the information content of the bandpass signal will be preserved in the lowpass signal.

To avoid aliasing and preserve the content of a bandpass signal with the bandwidth $B$, it must be samples at the rate:

$$2B \leq F_s \leq 4B \quad (2.36.4)$$
Decimation and interpolation by frequency conversion

A bandpass signal $x_n$ sampled at a rate $F_s$ can be converted to its lowpass representation as shown.

The sampling rate conversion can be performed then on the lowpass signal by the techniques described previously.

The LPF for obtaining the two quadrature components may be designed with linear phase within the frequency range of interest and approximating the ideal frequency response:

$$H(\omega) = \begin{cases} 1, & |\omega| \leq \omega_B / 2 \\ 0, & \text{otherwise} \end{cases} \quad (2.37.1)$$

where $\omega_B \leq \pi$ is the bandwidth of the discrete-time bandpass signal.

Decimation and interpolation by frequency conversion

In the case of decimation by an integer factor $D$, the antialiasing filter (preceding the decimation) can be combined with the LPF in the frequency converter to approximate the ideal frequency response:

$$H_D(\omega) = \begin{cases} 1, & |\omega| \leq \omega_B / D \\ 0, & \text{otherwise} \end{cases} \quad (2.38.1)$$

where $\omega_B$ is any desired frequency in the range $0 \leq \omega_B \leq \pi$.

In the case of interpolation by an integer factor $I$, the filter for rejecting the images in the spectrum should approximate the ideal frequency response:

$$H_I(\omega) = \begin{cases} I, & |\omega| \leq \omega_B / 2I \\ 0, & \text{otherwise} \end{cases} \quad (2.38.2)$$

In this case, the LPF used to reject the double-frequency components is redundant since its function is performed by the filter $H_I$. 
Decimation and interpolation by frequency conversion

The sampling-rate conversion by any rational factor $I/D$ can be accomplished by the system shown.

Only one image rejection/antialiasing filter is needed in this situation, whose frequency response approximates the following ideal characteristic:

$$H_I(\omega) = \begin{cases} I, & 0 \leq |\omega| \leq \min\left(\frac{\omega_c}{2I}, \frac{\omega_c}{2D}\right) \\ 0, & \text{otherwise} \end{cases} \quad (2.39.1)$$

After sampling rate conversion, a bandpass signal can be regenerated by amplitude modulating the quadrature components by corresponding signal components and adding them together. The center frequency $\omega_c$ can be selected in the range:

$$\min\left(\frac{\omega_c}{2I}, \frac{\omega_c}{2D}\right) \leq |\omega_c| \leq \pi \quad (2.39.2)$$

Modulation-free decimation and interpolation

Carrier modulation can be avoided in frequency translation by restricting the frequency range of the signal.

Consider the decimation of the sampled bandpass signal, whose spectrum is shown:

and is confined to the frequency range

$$\frac{m\pi}{D} \leq \omega \leq \frac{(m+1)\pi}{D} \quad (2.40.1)$$

where $m$ is a positive integer. A bandpass filter is used to eliminate unwanted frequency components and then a direct decimation of the bandpass signal by the factor $D$ results in a spectrum.
Modulation-free decimation and interpolation

If \( m \) is odd

If \( m \) is even

For an odd \( m \), the signal spectrum is inverted, which can be undone by multiplying each sample of the decimated signal by

\[
(-1)^n, \quad n = 0, 1, \ldots
\]

(2.41.1)

If (2.40.1) is violated, aliasing will occur.

Modulation-free decimation and interpolation

Modulation-free interpolation of a bandpass signal by an integer factor \( I \) can be accomplished in a similar manner. Inserting zeros between samples of \( x_n \) produces \( I \) images in the band \( 0 \leq \omega \leq \pi \). The desired image is selected by bandpass filtering.

Modulation-free sampling rate conversion of a bandpass signal by a rational factor \( I/D \) can be accomplished by cascading a decimator with an interpolator. A bandpass filter preceding the sampling converter is usually needed to isolate the signal frequency band of interest.
Rate conversion by an arbitrary factor

Sampling rate conversion by a rational factor using methods discussed so far may be sometimes inefficient or even impossible:

1) When \( I \) is a large integer, a polyphase filter with \( I \) (for example, 1023) subfilters would be extremely inefficient with memory usage.
2) In some applications, the exact rate conversion factor may be unknown, which makes the conventional approaches inapplicable.

To account for such (and similar) situations, nonexact conversion schemes are used. Such methods introduce distortion in the output signal. However, distortions exist even in exact rate convertors since there are no ideal filters… Therefore, a total distortion introduced by a nonexact converter must not exceed the specifications.

Depending on the application, first-order, second-order, and higher-order approximation methods can be used.

Rate conversion by an arbitrary factor: 1\textsuperscript{st} order approximation

Assume that an input signal \( x_n \) needs to be re-sampled with a rate conversion factor \( r \). Therefore, the output signal samples must be separated in time by \( T_x/r \) where \( T_x \) is the sample interval for \( x_n \). By constructing a polyphase filter with a large number of subfilters as described above, we can approximate such a sequence with a nonuniformly spaced sequence with

\[
\frac{1}{r} = \frac{k}{I} + \beta
\]

where \( k \) and \( I \) are positive integers and \( \beta \) is a number in the range:

\[
0 < \beta < 1/I
\]

Therefore:

\[
\frac{k}{I} < \frac{1}{r} < \frac{k+1}{I}
\]

We note that \( I \) is the interpolation factor (and a number of polyphase filters) that needs to be determined to satisfy specifications on the tolerable distortion.
Rate conversion by an arbitrary factor: 1st order approximation

For example, suppose that \( r = 2.2 \) and we have determined that \( I = 6 \). Then:

\[
\frac{k}{I} = \frac{2}{6} < \frac{1}{r} = \frac{1}{2.2} < \frac{k+1}{I} = \frac{3}{6}
\]

We notice that the decimation factor is 2.727, which falls between \( k = 2 \) and \( k = 3 \).

Using the 1st-order approximation, we achieve the desired decimation rate by selecting the output sample from the polyphase filter that is closest in time to the desired sampling time.

Rate conversion by an arbitrary factor: 1st order approximation

So, in general, to perform a rate conversion by a factor \( r \), we use a polyphase filter performing interpolation (increase the sampling frequency) by a factor \( I \).

The time spacing between the samples of the interpolated sequence is \( T_x/I \).

If the ideal (desired) sampling time of the \( m^{th} \) sample, \( y_m \), of the desired output sequence is between the sampling times of two samples of the interpolated sequence, we select the sample closer to \( y_m \) as its approximation.
Rate conversion by an arbitrary factor: 1st order approximation

Assuming that the input sequence $x_n$ has a flat spectrum from $-\omega_x$ to $\omega_x$ with a magnitude $A$, a total power can be found by Parseval’s theorem as

$$P_s = \frac{1}{2\pi} \int_{-\omega_x}^{\omega_x} |X(\omega)|^2 d\omega = \frac{A^2 \omega_x^2}{\pi} \quad (2.47.1)$$

We can show that the total error power due to the sampling-time distortion is

$$P_e \leq \frac{1}{2\pi} \int_{-\omega_x}^{\omega_x} A^2 \left( \frac{0.5}{I} \right)^2 \omega^2 d\omega = \frac{A^2 \omega_x^3}{12\pi I^2} \quad (2.47.2)$$

The error power is inversely proportional to the square of the number of subfilters $I$.

The ratio of signal-to-distortion due to a sampling-time error for the 1st-order approximation is

$$SDR_1 = \frac{P_s}{P_e} \geq \frac{12I^2}{\omega_x^2} \quad (2.47.3)$$

Rate conversion by an arbitrary factor: 2nd order approx (linear interpolation)

The major disadvantage of 1st-order approximation method is a large number of subfilters needed to achieve the specified distortion requirements.

Linear interpolation (2nd-order approximation) allows achieving the same performance with less subfilters.

Instead of using the closes sample as the approximation, two adjacent samples are computed. Linearly interpolated (between these samples) result is used.
Rate conversion by an arbitrary factor: 2nd order approx (linear interpolation)

We can show that, assuming an input signal with a flat spectrum as before, the total error power due to the sampling-time distortion is

$$P_e \leq \frac{1}{2\pi} \int_{-\omega_s}^{\omega_s} A^2 \left( \frac{0.25}{I} \right)^2 \omega^4 d\omega = \frac{A^2 \omega_s^5}{80\pi I^4} \tag{2.49.1}$$

The error power is inversely proportional to $I^4$ (error magnitude proportional to the square of the number of subfilters).

The ratio of signal-to-distortion due to a sampling-time error for the 2nd-order approximation is

$$\frac{SD_i R2}{P_s} = \frac{80I^4}{\omega_s^4} \tag{2.49.2}$$

The signal-to-distortion ratio is proportional to $I^4$.

Applications of MR SP: phase shifters

Assume that we need a network delaying the signal by a fraction of a sample:

$$d = \frac{k}{I} T_x \tag{2.50.1}$$

where $k$ and $I$ are integers and $T_x$ is the sampling interval. This delay corresponds to a linear phase shift:

$$\theta(\omega) = -\frac{k\omega}{I} \tag{2.50.2}$$

Design of all-pass linear-phase filter with such a delay is not very simple. We may use the methods of sampling rate conversion instead...

Overall delay by $k/I$ samples
Applications of MR SP: phase shifters

The interpolator can be implemented efficiently by using a polyphase filter. The delay of $k$ samples is achieved by placing the initial position of the commutator at the output of the $k^{th}$ subfilter. Since $D = I$ in this case, the commutator position may be fixed to the output of the $k^{th}$ subfilter. Therefore, the delay by $k/I$ samples is achieved.

Note 1: the polyphase filter introduces additional delay by $(M-1)/2$ samples, where $M$ is the length of its impulse response.

Note 2: if the delay is a non-rational factor of the sample interval $T_x$, either 1st-order or 2nd-order approximation method may be used.

Applications of MR SP: interfacing systems with different sampling rates

Frequently, two digital systems controlled by independently operating clocks must be interfaced. The simplest approach is to use basic sample-rate conversion methods.

The system A output at rate $F_x$ is upsampled by a factor of $I$, fed at the rate $IF_x$ to a digital sample-and-hold that serves as an interface to system B, and read out into system B at the clock rate $DF_y$.

Therefore, after decimation by $D$, two systems will be synchronized.

In a special case when $I = D$, two clock rates are comparable but not identical.
Applications of MR SP: implementation of narrowband LPF

When a filter with narrow passband and transition bands needs to be implemented, using sampling-rate conversion may be advantageous and provide more efficient realization. Under these conditions, a linear-phase FIR LPF may be more efficiently implemented in a multistage decimator-interpolator configuration. More specifically, we may use a multistage implementation of a decimator of size $D$ followed by a multistage implementation of an interpolator of size $I$ where $D = I$.

The major advantage of multistage implementation is the reduction in the passband ripple requirements (stopband ripples are the same):

$$\delta_{p,\text{stage}} = \frac{\delta_p}{\# \text{ of stages}} \quad (2.53.1)$$

Another computational advantage comes from the processing at lower frequency.

Applications of MR SP: implementation of narrowband LPF

Example: design a linear-phase FIR filter satisfying the following specifications:

- Sampling frequency: 8000 Hz
- Passband: $0 \leq F \leq 75$ Hz
- Transition band: $75 \leq F \leq 80$ Hz
- Stopband: $80 \leq F \leq 4000$ Hz
- Passband ripple: $\delta_p = 10^{-2}$
- Stopband ripple: $\delta_s = 10^{-4}$

A single-rate implementation would require the filter length (Kaiser's formula) of

$$\hat{M} \approx 5152$$

Reduction of sampling rate by a factor of $D = 50$ would reduce the Nyquist frequency to 80 Hz and, therefore, the system would automatically satisfy the specifications. We need to ensure that the antialiasing filters meet the ripples specifications. A single-rate filter would be of length of approximately 5480. Implementing two stages with $D_1 = 25$, $D_2 = 2$, $I_1 = 2$, and $I_2 = 25$, will reduce filter lengths to approximately 177 and 233.
Applications of MR SP: implementation of narrowband LPF

Example

\[ f_p = 0.00475; f_s = 0.005; \delta_p = 0.001; \delta_s = 0.0001; (F_s = 1) \]

<table>
<thead>
<tr>
<th></th>
<th>Direct Form</th>
<th>1-stage</th>
<th>2-stage</th>
<th>3-stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimation rates</td>
<td>-</td>
<td>100</td>
<td>50; 2</td>
<td>10; 5; 2</td>
</tr>
<tr>
<td>Filter lengths</td>
<td>15,590</td>
<td>16,466</td>
<td>423; 347</td>
<td>50; 44; 356</td>
</tr>
<tr>
<td>Multiplications per second</td>
<td>7,795</td>
<td>165</td>
<td>11.9</td>
<td>9.4</td>
</tr>
<tr>
<td>Rate reduction</td>
<td>1</td>
<td>47.2</td>
<td>655</td>
<td>829</td>
</tr>
<tr>
<td>Coefficient storage</td>
<td>7,795</td>
<td>8,233</td>
<td>385</td>
<td>225</td>
</tr>
</tbody>
</table>

Rate reduction since the processing is done at a lower frequency  
Gain up to some point

In multistage implementations, generally, the greatest decimation factor goes first and the smallest – last. Interpolation is reversed (i.e., the greatest factor goes last).

Applications of MR SP: implementation of filter banks

There are 2 types of filter banks: analysis and synthesis FB.

Due to the frequency characteristics of its filters, an analysis FB splits the signal into a number of subbands.

Analysis FBs are frequently used in spectral analysis.
Applications of MR SP: implementation of filter banks

A synthesis FB sums its filtered input signals to synthesize the output.

Applications of MR SP: implementation of filter banks

Analysis filter banks can be used to perform DFT analysis – DFT filter bank. A DFT filter bank consisting of \( N \) filters \( \{H_k(z), k = 0,1,\ldots, N-1\} \) is called a uniform DFT filter bank if \( H_k(z) \) are derived from a prototype filter \( H_0(z) \) as follows:

\[
H_k(\omega) = H_0 \left( \omega - \frac{2\pi k}{N} \right), \quad k = 1,2,\ldots, N-1
\]  

(2.58.1)

The frequency responses of the filters are obtained by uniformly shifting the frequency response of the prototype filter by multiples of \( 2\pi/N \); or in the time domain:

\[
h_{k,n} = h_{0,n} e^{j2\pi nk/N}, \quad k = 0,1,\ldots, N-1
\]  

(2.58.2)

where \( h_{0,n} \) is the impulse response of the prototype filter.
Applications of MR SP: implementation of filter banks

The uniform DFT FB can be realized as shown.

Here, the input sequence is translated in frequency to lowpass by multiplying $x_n$ with the complex exponents. The result is filtered by a LPF $h_{0,n}$.

Since the filter outputs are relatively narrowband, the signals may be decimated by $D \leq N$ to form the output:

$$X_{k,m} = \sum_n h_{0,mD-n}x_n e^{-j2\pi n k/N}, \quad k = 0,1,...,N-1; \quad m = 0,1,...$$

(2.59.1)

where $X_{k,m}$ are samples of the DFT at frequencies $\omega_k = 2\pi k/N$.

Applications of MR SP: implementation of filter banks

In the synthesis FB, the input signals are upsampled by $D$, filtered to remove mirror images, translated in frequency, and finally summed to form

$$v_n = \frac{1}{N} \sum_{k=0}^{N-1} e^{j2\pi n k/N} \left[ \sum_m g_{0,n-m} \right] = \sum_{m} g_{0,n-m} \left[ \frac{1}{N} \sum_{k=0}^{N-1} Y_{k,m} e^{j2\pi n k/N} \right] = \sum_{m} g_{0,n-m} Y_{n,m}$$

(2.60.1)

where $1/N$ is the normalization factor and $Y_{n,m}$ represent samples of the IDFT $Y_k(m)$. 
Applications of MR SP: implementation of filter banks

Alternatively, analysis and synthesis FBs can be implemented as follows. The filters are **bandpass**: 

\[ h_{k,n} = h_{0,n} e^{j2\pi nk/N} \]  

\[ k = 0,1,...,N-1 \]

Computationally efficient FBs can be achieved by use of polyphase filters. When \( D = N \), the FB is called critically sampled.

In this situation, polyphase filters will have the impulse responses:

\[ p_{k,n} = h_{0,n} e^{j2\pi nk/N} \]  

\[ k = 0,1,...,N-1 \]

Here \( h_{0,n} \) and \( g_{0,n} \) are the responses of the prototype filters.

The outputs in frequency domain can be described:

\[ X_{k,n} = \sum_{n=0}^{N-1} \left( \sum_{l=0}^{D-1} p_{h,n} x_{n,m-l} \right) e^{-j2\pi nk/N}, \quad k = 0,1,...D-1 \]  

Here \( N = D \).
Applications of MR SP: implementation of filter banks

The synthesis FB can be implemented as follows assuming \( I = D = N \).

\[
q_{k,n} = g_{0,nN+k} \quad (2.63.1)
\]

\( k = 0, 1, \ldots, N - 1 \)

The output of the \( l \)th polyphase filter can be expressed as:

\[
v_{l,n} = \sum_{m} q_{1,n+m} \left[ \frac{1}{N} \sum_{k=0}^{N-1} y_{k,m} e^{-j2\pi kN^{-1}} \right] = \sum_{m} q_{1,n+m} v_{l,m}, \quad l = 0, 1, \ldots, N - 1 \quad (2.63.2)
\]

Applications of MR SP: subband coding of speech signals

Most of the speech energy is contained in low frequencies. Therefore, it is beneficial to encode the low-frequency band with more bits than the high-frequency band. Subband coding – the method where the (speech) signal is subdivided into several frequency bands and each band is encoded separately.

If the input is sampled at \( F_s \), the 1\(^{st} \) frequency subdivision splits it into a lowpass signal \([0, F_s/4]\) and a highpass signal \([F_s/4, F_s/2]\). The 2\(^{nd} \) frequency subdivision splits the lowpass signal from the 1\(^{st} \) stage into a lowpass \([0, F_s/8]\) and a highpass \([F_s/8, F_s/4]\) signals. The 3\(^{rd} \) frequency subdivision splits the lowpass signal from the 2\(^{nd} \) stage into a lowpass \([0, F_s/16]\) and a highpass \([F_s/16, F_s/8]\) signals.
Applications of MR SP: subband coding of speech signals

Decimation by a factor of 2 after frequency subdivision reduces the bit rate of the digital speech signal.

Filter design is very important since aliasing must be negligible. The "ideal", brick-wall filters are physically unrealizable and a practical solution to the problem is provided by the quadrature mirror filters (QMF).

Applications of MR SP: subband coding of speech signals

The synthesis FB is a reversed version of the analysis FB.

Subband coding is an effective method for achieving a bandwidth compression in a digital signal (or image), when its energy is in a particular frequency band.
Applications of MR SP: QMF

A two-channel QMFB:

A multirate filter employing 2 decimators in the analysis section and 2 interpolators in the synthesis.

For the input signal $x_n$, the outputs of the analysis section (after decimators) are:

$$X_{a0}(\omega) = \frac{1}{2} \left[ X \left( \frac{\omega}{2} \right) H_0 \left( \frac{\omega}{2} \right) + X \left( \frac{\omega - 2\pi}{2} \right) H_0 \left( \frac{\omega - 2\pi}{2} \right) \right]$$

$$X_{a1}(\omega) = \frac{1}{2} \left[ X \left( \frac{\omega}{2} \right) H_1 \left( \frac{\omega}{2} \right) + X \left( \frac{\omega - 2\pi}{2} \right) H_1 \left( \frac{\omega - 2\pi}{2} \right) \right]$$

(2.67.1)

Applications of MR SP: QMF

If no processing is done between the analysis and synthesis FBs (i.e., the signals (2.67.1) are input to the synthesis FB), the output would be:

$$\hat{X}(\omega) = X_{a0}(2\omega)G_0(\omega) + X_{a1}(2\omega)G_1(\omega) = \{\text{no processing}\}$$

$$= \frac{1}{2} \left[ H_0(\omega)G_0(\omega) + H_1(\omega)G_1(\omega) \right] X(\omega) + \frac{1}{2} \left[ H_0(\omega - \pi)G_0(\omega) + H_1(\omega - \pi)G_1(\omega) \right] X(\omega - \pi)$$

Desired output

aliasing

Therefore, to avoid aliasing, we require that

$$H_0(\omega - \pi)G_0(\omega) + H_1(\omega - \pi)G_1(\omega) = 0$$

(2.68.2)

which can be satisfied by selecting

$$G_0(\omega) = H_1(\omega - \pi), \quad G_1(\omega) = -H_0(\omega - \pi)$$

(2.68.3)
Applications of MR SP: QMF

Assuming that $H_0$ is a LPF and $H_1$ is its mirror-image HPF expressed as:

\[ H_0(\omega) = H(\omega) \]
\[ H_1(\omega) = H(\omega - \pi) \]  

(2.69.1)

or, on the time domain:

\[ h_{0,n} = h_n \]
\[ h_{1,n} = (-1)^n h_n \]  

(2.69.2)

Therefore, $H_0$ and $H_1$ have mirror-image symmetry about the frequency $\omega = \pi/2$.

For consistency, we select:

\[ G_0(\omega) = 2H(\omega) \]
\[ G_1(\omega) = -2H(\omega - \pi) \]  

(2.70.1)

or, on the time domain:

\[ g_{0,n} = 2h_n \]
\[ g_{1,n} = -2(-1)^n h_n \]  

(2.70.2)

The scale factor 2 corresponds to the interpolation factor used to normalize the overall frequency response of the QMF.

Therefore, the synthesis QMFB output is

\[ \hat{X}(\omega) = [H^2(\omega) - H^2(\omega - \pi)]X(\omega) \]  

(2.70.3)

Ideally, a 2-channel QMFB should have a unity gain:

\[ |H^2(\omega) - H^2(\omega - \pi)| = 1, \quad \forall \omega \]  

(2.70.4)
Applications of MR SP: QMF

While using a linear phase FIR filter $H(\omega)$:

$$H(\omega) = H_r(\omega)e^{-j\omega\frac{N-1}{2}}$$  \hspace{1cm} (2.71.1)

where $N$ is the filter length. Then

$$H^2(\omega) = H_r^2(\omega)e^{-j\omega(N-1)} = \left|H_r(\omega)\right|^2e^{-j\omega(N-1)}$$  \hspace{1cm} (2.71.2)

$$H^2(\omega - \pi) = H_r^2(\omega - \pi)e^{-j(\omega-\pi)(N-1)} = (-1)^{N-1}\left|H_r(\omega - \pi)\right|^2e^{-j\omega(N-1)}$$  \hspace{1cm} (2.71.3)

Therefore, the overall transfer function of the 2-channel QMFB that uses linear phase FIR filters is

$$\frac{\hat{X}(\omega)}{X(\omega)} = \left[\left|H(\omega)\right|^2 - (-1)^{N-1}\left|H(\omega - \pi)\right|^2\right]e^{-j\omega(N-1)}$$  \hspace{1cm} (2.71.4)

The overall delay of the QMFB is $N - 1$ samples.

Applications of MR SP: QMF

The overall magnitude characteristic of a FB is:

$$A(\omega) = \left|H(\omega)\right|^2 - (-1)^{N-1}\left|H(\omega - \pi)\right|^2$$  \hspace{1cm} (2.72.1)

Note that when $N$ is odd, $A(\pi/2) = 0$, which is undesirable. When $N$ is even:

$$A(\omega) = \left|H(\omega)\right|^2 + \left|H(\omega - \pi)\right|^2$$  \hspace{1cm} (2.72.2)

The ideal QMFB should satisfy

$$A(\omega) = \left|H(\omega)\right|^2 + \left|H(\omega - \pi)\right|^2 = 1, \hspace{0.5cm} \forall \omega$$  \hspace{1cm} (2.72.3)

However, any real (and nontrivial) linear phase FIR filter will introduce some amplitude distortion. A perfect reconstruction QMFB can be designed using a linear phase half-band FIR filter that is defined as a zero-phase FIR with response

$$b_{2n} = \begin{cases} \text{constant}, & n = 0 \\ 0, & n \neq 0 \end{cases}$$  \hspace{1cm} (2.72.4)
Applications of MR SP: QMF

All even-numbered samples are 0 except at \( n = 0 \). The frequency response of such filter is

\[
B(\omega) = \sum_{n=-K}^{K} b_n e^{-j\omega n}
\]

(2.73.1)

where \( K \) is odd. Furthermore, the filter satisfies the condition:

\[
B(\omega) + B(\pi - \omega) = \text{const}, \quad \forall \omega
\]

(2.73.2)

The typical frequency response is symmetric about \( \pi/2 \); the band edges are symmetric about \( \pi/2 \); the stopband and passband ripples are equal. The filter can be made causal by introducing a \( K \)-sample delay.

Applications of MR SP: QMF

Suppose that we design an FIR half-band filter of length \( 2N - 1 \), where \( N \) is even, with the frequency response shown. From \( B(\omega) \) we construct another half-band filter with the frequency response

\[
B_+ (\omega) = B(\omega) + \delta e^{-j\omega(N-1)}
\]

(2.74.1)

that is non-negative and has the spectral factorization

\[
B_+ (z) = H(z) H(1/z) z^{-(N-1)}
\]

(2.74.2)

or, equivalently:

\[
B_+ (\omega) = |H(\omega)|^2 + e^{-j\omega(N-1)}
\]

(2.74.3)
Applications of MR SP: QMF

where $H(\omega)$ is the frequency response of an FIR filter of length $N$ with real coeffs. Due to symmetry:

$$B_v(z) + (-1)^{N-1} B_v(-z) = \alpha z^{-(N-1)}$$

(2.75.1)

or:

$$B_v(\omega) + (-1)^{N-1} B_v(\omega - \pi) = \alpha e^{-j\omega(N-1)}$$

(2.75.2)

where $\alpha$ is a constant. Therefore:

$$H(z)H\left(\frac{1}{z}\right) + H(-z)H\left(-\frac{1}{z}\right) = \alpha$$

(2.75.3)

An aliasing-free perfect reconstruction QMFB is achieved when

$$H_0(z) = H(z)$$

$$H_1(z) = -z^{-(N-1)}H_0\left(-\frac{1}{z}\right)$$

$$F_0(z) = z^{-(N-1)}H_0\left(\frac{1}{z}\right)$$

$$F_1(z) = z^{-(N-1)}H_1\left(\frac{1}{z}\right) = -H_0\left(-z\right)$$

(2.75.4)

However, $H(z)$ is not a linear phase filter.

A 2-channel QMFB can be realized efficiently when using polyphase filters with the responses:

$$p_{i,m} = h_{2m+i}, \quad i = 0, 1$$

(2.76.1)
Applications of MR SP: oversampling A/D and D/A converters

The idea of oversampling A/D converters is to increase the sampling rate of the signal to the point where a low-resolution quantizer will be sufficient. Therefore, the dynamic range of the signal values between successive samples can be reduced. An oversampling A/D converter is implemented by a cascade of an analog sigma-delta modulator (SDM) followed by a digital antialiasing decimation filter and a digital HPF.

An analog SDM produces a 1-bit per sample output at a very high sampling rate. This 1-bit output is passed through a LPF producing a high precision (multiple-bit) output that is decimated to a lower sampling rate. The decimated output is passed through a HPF that attenuates the quantization noise at the lower frequencies.

Applications of MR SP: oversampling A/D and D/A converters

An oversampling D/A converter performs the reverse operations.

A digital signal passes through a HPF, whose output is interpolated (resampled and filtered by an anti-imaging filter). This high sampling rate high precision digital signal is input to the digital SDM that provides a high sampling rate 1-bit per sample output. This 1-bit output is converted to an analog signal by lowpass filtering and further smoothing with an analog filter.

Questions?