The problem at hand is to design effective fractional delay filters, filters that are capable of implementing a time delay that is a non-integer multiple of the clock sampling period of the system.

For an analog signal, the input signal delayed by an amount $D$ is represented as:

$$y(t) = x(t - D)$$

The delay $D$ can take on any value and the above expression is valid for all values of $D$.

Most, if not all, practical signals are analog in nature and these are sampled to obtain signal values at discrete instants of time to enable storage of signal values at different time instants in digital processors. Thus, every analog signal is stored as a discrete time sequence in digital processors. The requirements to preserve accuracy of representation of the discrete signal impose a minimum sampling rate – the rate at which samples of the signal are taken – in accordance with the Nyquist sampling theorem – “The minimum sampling rate required to preserve the information content of a signal is equal to twice the frequency of the maximum frequency component of the signal”.

The time delay implementation for a discrete time signal is represented as:

$$y[n] = x[n - D]$$

The implicit assumption here is that the delay $D$ is an integer, for it is meaningless to delay the time sequence by a fractional number of samples. However, all practical signals being stored as discrete time sequences, it must be necessary to be able to obtain accurate representations of signal values at all time instants. This interpolation (more specifically band-limited interpolation) is what a fractional delay filter achieves.

The process of band-limited interpolation described above can be thought of as a three-step process:

1. Conversion of the sampled discrete time signal into the analog signal.
2. Implementation of the required fractional delay in the analog signal.
3. Re-conversion of the shifted signal into a discrete time sequence by re-sampling at the same sampling rate.

Considering the $z$ transform of the basic fractional delay filter expression, we can derive the expression for the transfer function of a fractional delay filter. Taking the $z$ transform yields:

$$Y(z) = z^{-D}X(z) \Rightarrow H(z) = \frac{Y(z)}{X(z)} = z^{-D}$$
Once again, this expression is meaningful only for integer values of $D$, strictly speaking, since the expression for fractional values of $D$ necessitates expression as an infinite power series. Thus, the above expression is implementable to absolute accuracy only for integer values of $D$. For non-integer values of $D$, as in the case of a fractional delay filter, only an approximation, accurate to within a minimum tolerance level, can be implemented.

The inverse $z$ transform of the transfer function yields the following expression for the ideal impulse response of a fractional delay filter:

$$h[n] = Z^{-1}(z^{-D}) = \frac{\sin(\pi(n-D))}{\pi(n-D)} = \text{sinc}(n-D)$$

Here, the sinc function is also known as the interpolating function.

Thus, the ideal transfer function of a fractional delay filter is a sinc function involving the non-integer delay $D$ to be implemented. It is readily seen that for non-integer values of $D$, this is an infinite, unbounded and non-causal sequence.

Thus, suitable approximation methods must be designed to implement this transfer function in hardware as a finite impulse response (FIR) filter. The design methods covered in this document are:

1. Lagrange Interpolation
2. Least Squares Approximation
3. Offset Windowing

Below is a brief exposition of these methods.

### 1. Lagrange Interpolation

This is the simplest method of implementing a fractional delay filter. It involves minimizing the error function at a particular frequency, generally zero, achieved by setting all derivatives of the error function at that frequency equal to zero:

$$\frac{dE^n}{d\omega^n} = 0, n = 0,1,2,\ldots,N$$

where $E(\omega) = H(\omega) - H_d(\omega)$, $H_d(\omega)$ being the ideal transfer function and $H(\omega)$ the implemented transfer function.

The resulting set of $(N+1)$ equations yields the following expression for the impulse response:

$$h[n] = \prod_{k=0, k\neq n}^{N-1} \frac{D-k}{n-k}, n = 0,1,2,\ldots,N-1$$

This expression is now implemented for desired accuracy by choosing an appropriate value of $N$. 
2. Least Squared Error Approximation

This method involves suitably choosing the filter length \( N \) to minimize the mean-squared error given by:

\[
\int \! \! \! \int \! |E(\omega)|^2 \, d\omega = \int \! \! \! \int \! |H(\omega) - H_d(\omega)|^2 \, d\omega
\]

By Parseval’s theorem, this error has an exact representation in the time domain as:

\[
\sum_{-\infty}^{\infty} |h[n] - h_d[n]|^2
\]

The method implements the ideal transfer function \( h_d[n] \), truncated to a particular number of coefficients. Thus, the transfer function is given by:

\[
h[n] = \text{sinc}(n - D), \quad M \leq n \leq M + N - 1
\]
\[
0, \quad \text{otherwise}
\]

For a causal filter, we must have \( M \geq 0 \). Generally, \( M = 0 \) is taken as the standard value. The least squared error criterion leads to the following relation between \( N \) and \( D \).

The delay \( D \) should be located between the two central taps of the filter when \( N \) is odd or within half a sample from the central tap when \( N \) is even, since then the approximation error is smallest. This means that the delay \( D \) should be chosen so that the following relation is satisfied:

\[
\frac{(N-1)}{2} \leq D \leq \frac{N+1}{2}
\]

For an odd-order FIR interpolator (\( N \) is odd), this implies that the integer part of the delay \( D_{\text{int}} \) and \( N \) satisfy the relation:

\[
D_{\text{int}} = \frac{N-1}{2}
\]

For an even-order FIR interpolator (\( N \) is even), the following relations hold between \( D_{\text{int}} \), \( d \) and \( N \) (where \( D = D_{\text{int}} + d \)):

\[
D_{\text{int}} = \frac{N}{2} \quad \text{when } 0 \leq d \leq \frac{1}{2}
\]
\[
\frac{N}{2} - 1 \quad \text{when } \frac{1}{2} \leq d < 1
\]
3. Offset Windowing

Windowing an ideal impulse response is a common method of reducing the so-called Gibbs phenomenon – ripples produced in the frequency domain as a result of truncating the ideal impulse response to a finite number of coefficients. Truncating is in effect, multiplying by a rectangular window, which has an infinite rate of tapering. By using a window that has a gentle roll off to zero, the ripples in the frequency domain are greatly reduced, at the expense of a wider transition width.

In designing FIR fractional delay filters, offset windows are used, the amount of offset being the required delay $D$. Thus, the impulse response takes the form:

$$h[n] = w[n-D] \text{sinc}(n-D) \quad \text{for } M \leq n < M+N$$

$$0 \quad \text{otherwise}$$

Here, the center of the window function of length $N-1$ has been shifted by an amount $D$ so that the shifted sinc function will be windowed symmetrically with respect to its center.

The windowing method is suitable for real-time systems where the fractional delay is changed since the coefficients can be updated quickly. The samples of the sinc and the window function can be stored in memory for several values of $D$. The filter coefficients for delay values between the stored ones may be obtained, e.g., by linear interpolation.

The window functions used are the raised cosine windows. Some commonly used windows are the following:-

1) Hanning (Von Hann) Window

$$w[n] = 0.5 + 0.5 \cos\left(\frac{2\pi n}{N}\right)$$

2) Hamming Window

$$w[n] = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N}\right)$$

3) Blackman Window

$$w[n] = 0.42 + 0.5 \cos\left(\frac{2\pi n}{N-1}\right) + 0.08 \cos\left(\frac{4\pi n}{N-1}\right)$$

4) Optimal Three Term Window

$$w[n] = 0.375 + 0.5 \cos\left(\frac{2\pi n}{N}\right) + 0.125 \cos\left(\frac{4\pi n}{N}\right)$$

Here, the window functions are defined over the range $|n| \leq (N-1)/2$. 