Digital Signal Processing
Polyphase Implementation of Filtering

D. Richard Brown III
A Problem With Exchanging The Order of Up/Downsampling and Filtering

Recall our equivalent structures:

We prefer (a) in both cases because the filtering is done at the lower sampling rate.

This can be directly implemented in some cases, e.g. $M = 2$ and $H(z^M) = 1 + z^{-2}$. It is clear that $H(z) = 1 + z^{-1}$.

But what if $M = 2$ and $H(z^M) = 1 + z^{-1} + z^{-2}$? Does $H(z) = 1 + z^{-1/2} + z^{-1}$ make sense?
Polyphase Sequence Decompositions

Given an integer $M \geq 1$, we can decompose any discrete-time sequence $h[n]$ into $M$ subsequences defined as

$$e_k[n] = h[nM + k]$$

for $k = 0, \ldots, M - 1$.

For example, suppose $h[n] = \{1, 2, 3, 4, 5, 6\}$. An $M = 2$ polyphase decomposition results in

$$e_0[k] = \{1, 3, 5\}$$
$$e_1[k] = \{2, 4, 6\}$$

An $M = 3$ polyphase decomposition results in

$$e_0[k] = \{1, 4\}$$
$$e_1[k] = \{2, 5\}$$
$$e_2[k] = \{3, 6\}$$
Polyphase Components $\rightarrow$ Original Sequence

To recover the original sequence from the polyphase components, we can

1. Upsample each polyphase component by $M$
2. Delay the $k^{\text{th}}$ upsampled component by $k$ samples.
3. Sum.

Decomposition/reconstruction:

$$H(z) = \sum_{k=0}^{M-1} E_k(z^M) z^{-k}$$

Each subfilter is a function of $z^M$
Polyphase Decimation System

Suppose we had an $N$-coefficient FIR filtering system like

![Diagram of the polyphase decimation system]

Note that $M-1$ of the $M$ filter outputs are discarded. Is there a better way to do this?

Direct implementation requires $\approx N$ MACs per input sample.

Polyphase implementation:

- Samples arrive at each polyphase filter at a rate of $\frac{1}{M}$ the original sampling rate.
- Each subfilter has $\frac{N}{M}$ coefficients.

Hence each subfilter requires $\approx \frac{1}{M} \cdot \frac{N}{M}$ MACs per input sample. The total is then $\approx \frac{N}{M}$ MACs per input sample.

Computational savings achieved by filtering at the lower sampling rate.
Polyphase Interpolation System

Along the same lines, suppose we had an \( N \)-coefficient FIR filtering system like

![Diagram of polyphase filtering system]

Note that \( L - 1 \) of the \( L \) filter inputs are zero. Is there a better way to do this?

Direct implementation requires \( \approx LN \) MACs per input sample.

Polyphase implementation:

- Samples arrive at each polyphase filter at the original sampling rate.
- Each subfilter has \( \frac{N}{L} \) coefficients.

Hence each subfilter requires \( \approx \frac{N}{L} \) MACs per input sample. The total is then \( \approx N \) MACs per input sample.

Computational savings achieved by filtering at the lower sampling rate.
Remarks

- Exchanging the order of filtering and up/down-sampling can lead to equivalent systems with less computational requirements.
- Polyphase implementation allows this exchange to be possible for general filters.
- Matlab function `upfirdn` uses a polyphase interpolation structure.
- Also see Matlab function `resample`.

\[ Y = \text{resample}(X, P, Q) \] resamples the sequence in vector \( X \) at \( P/Q \) times the original sample rate using a polyphase implementation.