Fast Convolution Filtering -- From Basics to Filter Banks

Mark Borgerding
Xetron

Overview

- Fast Convolution is
  - Multiplication in the frequency domain
  - A faster FIR filter for more than 10-30 taps
- Additional channels are even cheaper
- Other operations can be done in frequency domain
  - Downsampling/Decimation
  - Mixing*
- Put it all together and shake it all about. That's what Filter Banks are all about.

Goals

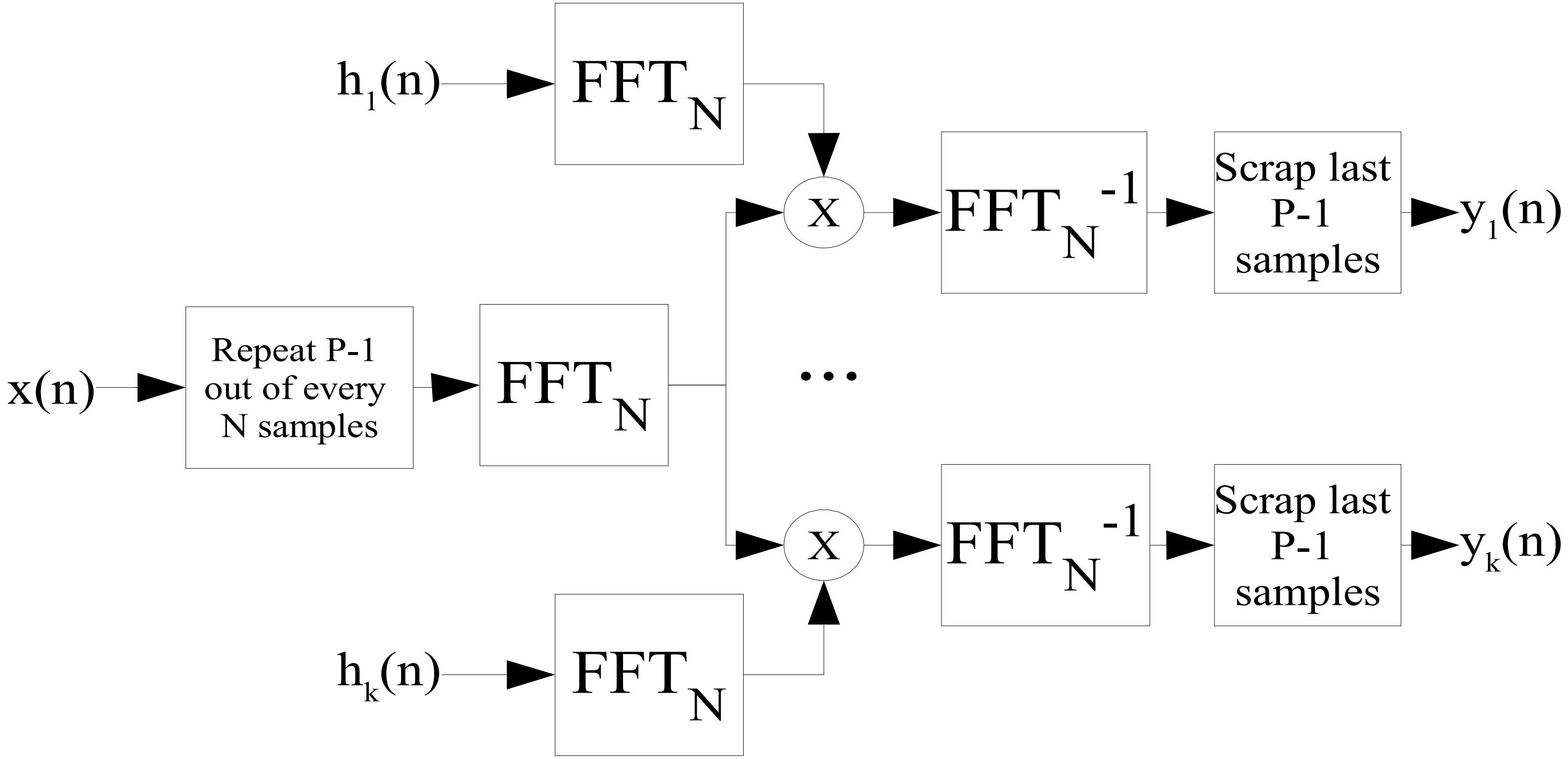
- Explain and demonstrate fast convolution filtering (FCF)
- Efficiently extend FCF to filter banks
  - Parallel filters
  - Mixing
  - Decimation

Parallel Filters

- May want to apply multiple filters against same input
- With FCF, the forward FFT can be reused
- Buy one at full price, get 40% off 2nd, 3rd ...
- Input must be consumed at same rate for all filters
  - All filters must be same order
  - Easily achieved by zero-padding

Parallel Filtering

- For k filters, only k+1 transforms are needed each buffer



Downsampling in the Frequency Domain

or "*How I learned to stop worrying and love the aliasing*"

- Sampling a signal aliases any energy above Nyquist rate
- Just as true in decimation as in analog to digital conversion
- Equivalent methods
  - N-sized IFFT then decimate by D
  - Alias high frequency energy into lowest N/D bins then perform N/D-sized IFFT
- Second faster than first because it shrinks the N in O(N*log*N) at cost of fewer than N additions
- Implies N/D is integer

Downsampling in the Frequency Domain

Demonstration Matlab/octave script

```
% make up a completely random frequency spectrum
X_freq=randn(1,1024) + i*randn(1,1024);

% BASELINE: traditional decimation path -- inverse transform then decimate
x_highrate = ifft( X_freq );
x_baseline = x_highrate(1:4:1024);  % every fourth sample

% ALTERNATIVE: alias in frequency domain, then smaller transform back to time
X_freq_alias = X_freq(1:256) + X_freq(257:512) + X_freq(513:768) + X_freq(769:1024);
x_alt = ifft(X_freq_alias) / 4;  % scale

max_error = max( abs( x_alt - x_baseline ) )
```

Perfect reconstruction!!

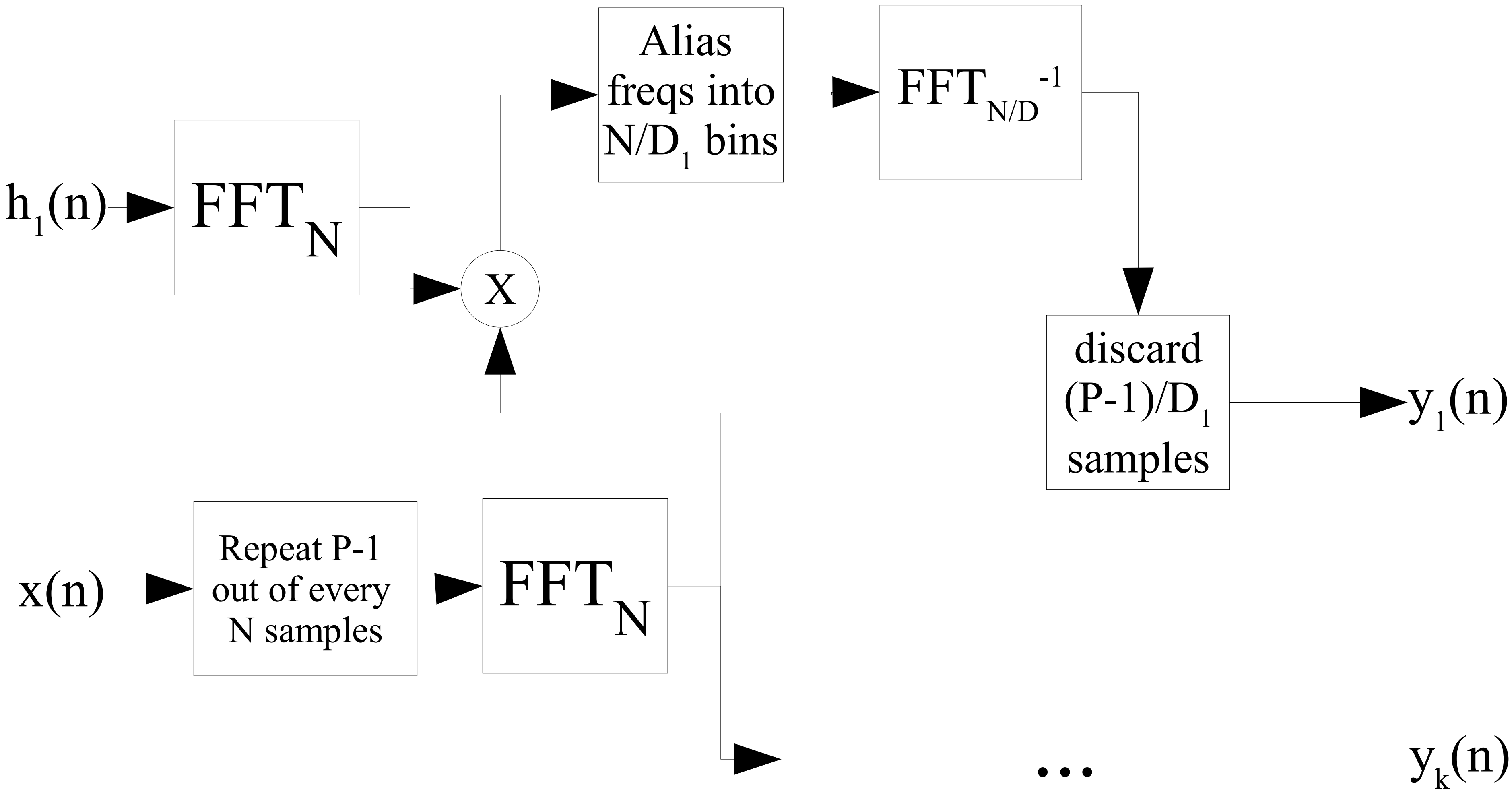
Frequency Domain Downsampling Applied to Fast Convolution

Special restrictions

- Filter order should be a multiple of decimation rate(s) i.e. $P' = K_1 D$ *(K_1 integer)*
  - Filter order is length of transient response
  - Transient response should be decimated to a whole number of samples
- FFT length should be a multiple of decimation rate(s) i.e. $N = K_2 D$ *(K_2 integer)*

Note: If overlap factor V is integer, then first restriction implies second. i.e. $N = VP' = VK_1D$, $\therefore$ $K_2 = VK_1$

Parallel, Downsampling FCF



Mixing

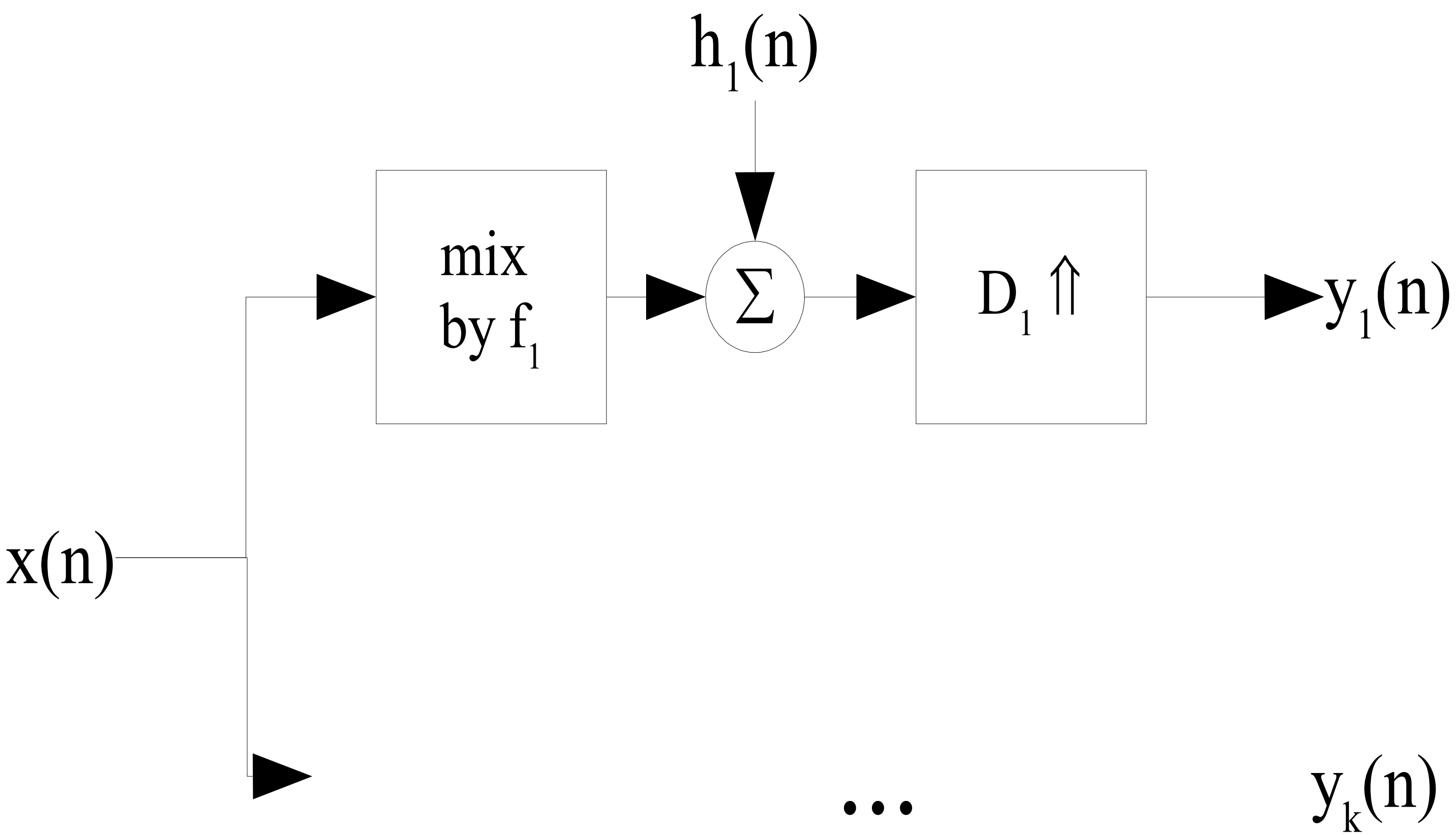
- Multiplication by complex sinusoid same as shifting frequency spectrum
  - Possible to mix by shifting FFT bins
    - Limited by frequency resolution of FFT
    - Further limited by the overlapping nature of FCF, i.e. The overlapped regions must be phase-coherent
  - Better to mix in the time domain

Mixing in FCF

- Mixing in frequency domain not flexible – time is better
- Mixing output is the general solution, since input is used by all channels
- Implies all filtering must be done on input (non-mixed) signal

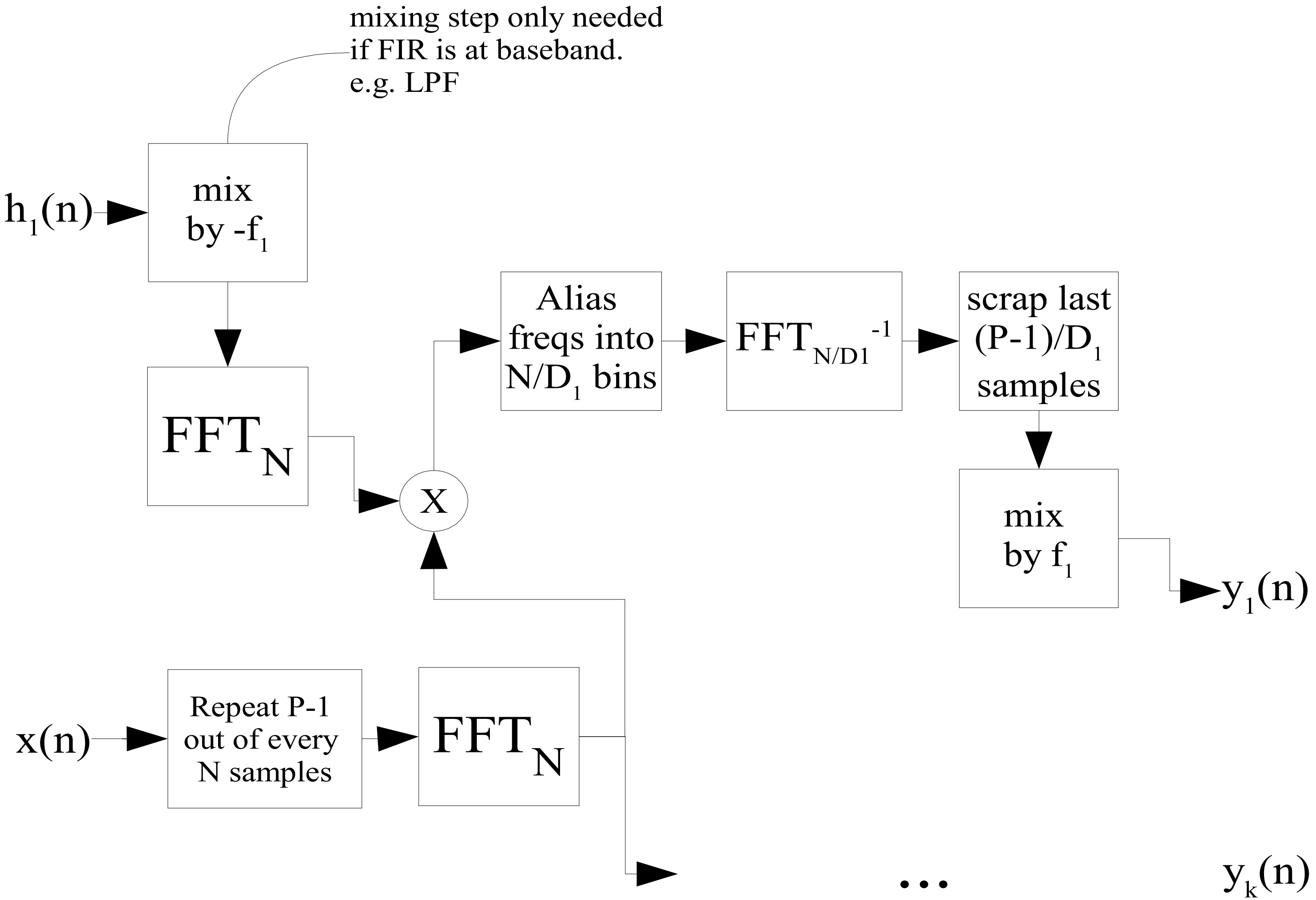
Summary – Conceptual Filter Bank

- It is useful to think of a filter bank as



Summary – Fast Filter Bank

- It is faster to realize the design as



Topics for Discussion

- Questions ???
- Further speedups possible by eliminating multiply & add for bins with negligible power
- (How) would Fast Filter Bank benefit from multirate/multistage techniques?
- Comparison to Analysis-Synthesis Filter Banks