

Performing Simultaneous Arbitrary Spectral Translation and Sample Rate Change, in Polyphase Interpolating or Decimating Filters in Transmitters and Receivers

fred harris
CUBIC Signal Processing Chair
San Diego State University

Chris Dick
Signal Processing Group
Xilinx Inc

ABSTRACT:

We describe and illustrate receivers employing modified N-path polyphase filters that allow resampling by arbitrary ratios while simultaneously baseband aliasing from center frequencies at Nyquist zones that are not multiples of the output sample rate. The technique described here uses a circular input buffer to couple between the input data array and the polyphase filter weight sets and a circular output buffer to couple between the polyphase filter outputs and the phase rotators of the FFT. The buffers enable arbitrary ratios between symbol rate and output sample rate and of center frequency spacing and output sample rates. The structure is applicable to single channel demodulators as well as to multi-channel systems. A number of applications of multi-channel filter banks are presented that illustrate use of the polyphase filter bank and FFT to obtain rational ratio resampling between input and output sample rates along with translations to and from arbitrary center frequencies unrelated to output sample rates.

1. POLYPHASE FILTER BANK

A multirate polyphase filter can perform the tasks of a multi-channel receiver. These tasks are equivalent to the down conversion, filtering, and resampling of multiple narrowband signals. The standard single channel demodulation process is shown in figure 1 and the equation describing the process is shown in equations 1 and 2. This process translates the carrier-centered spectrum to baseband where a filter reduces the bandwidth and a resampler reduces the sample rate in proportion to the bandwidth reduction.

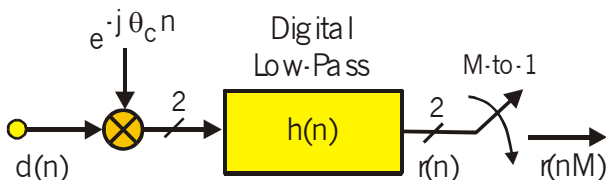


Figure 1. Standard Single Channel Down Converter: Heterodyne, Filter, and Down Sample

$$r(n) = [d(n) e^{-j\theta_c n}] * h(n) \quad (1)$$

$$r(n) = \sum_{k=0}^{N-1} d(n-k) e^{j\theta_c(n-k)} h(k) \quad (2)$$

We can factor part of the complex heterodyne out of the convolution summation and couple the complex heterodyne remaining in the summation to the filter coefficients. This factoring is shown in equation 3. In this form of the process, the filtering occurs at carrier and the down conversion is applied after, as opposed to prior to, the filtering task. The modification just described, known as the equivalency theorem, is shown in figure 2.

$$r(n) = e^{-j\theta_c n} \sum_{k=0}^{N-1} d(n-k) h(k) e^{-j\theta_c k} \quad (3)$$

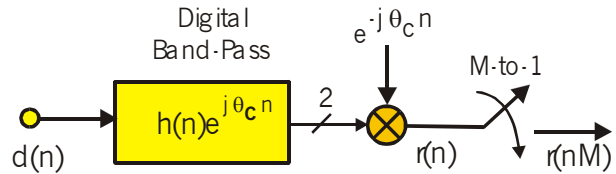


Figure 2. Reordered Single Channel Down Converter: Filter, Heterodyne, and Down Sample

Examining figure 2 we see that the output of the filter is down converted and then down-sampled. To avoid the foolishness of down converting samples and then discard them in the re-sampling operation, we reorder the heterodyne and the down sampler, and only down convert the retained samples. The frequency of the heterodyne at the reduced sample rate is now $M\theta_c$ rad/sample rather than original frequency of θ_c . If the center frequency θ_c is a multiple of the output sample rate $2\pi/M$, that is $k 2\pi/M$, the center frequency is aliased to zero by the M-to-1 resampling operation. Under this condition, the down sampled heterodyne defaults to unity and can be discarded. This is shown in figure 3.

Here too we realize it would be foolish to compute an output for each input and then have M-1 of the output samples be discarded by the down sampler. We thus modify the process and compute one output for every M-

inputs. The process that accommodates this modification is the partition of the original up-converted filter into a sum of sub-filters that operate at the reduced output rate rather than the original input rate. This partition is most easily seen as a mapping of the filter's Z-transform at the input rate to a sum of Z-transforms at the output rate. This partition is shown in equations 4 through 6 where we see that the phase rotators in each sub-filter is constant for that sub filter and is factored from the coefficient list. Figure 4 shows the block diagram representing equation 6 where the input commutator has replaced the delays Z^{-1} and the M-units of input delay have been replaced with 1-unit of output delay.

$$G(Z) = \sum_{n=0}^{N-1} h(n) e^{+j \frac{2\pi}{M} kn} Z^{-n} \quad (4)$$

$$= \sum_{r=0}^{M-1} \sum_{n=0}^{\frac{N}{M}-1} h(r+nM) e^{+j \frac{2\pi}{M} k(r+nM)} Z^{-(r+nM)} \quad (5)$$

$$= \sum_{r=0}^{M-1} Z^{-r} e^{+j \frac{2\pi}{M} kr} \sum_{n=0}^{\frac{N}{M}-1} h(r+nM) Z^{-nM} \quad (6)$$

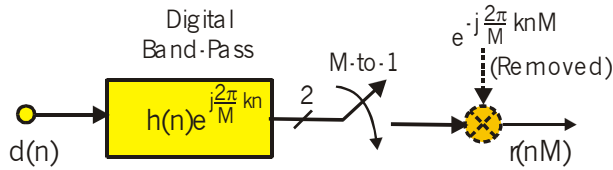


Figure 3. Down Converter with Center Frequency at a Multiple of Output Sample Rate: Aliased to Baseband by M-to-1 Down-Sampling

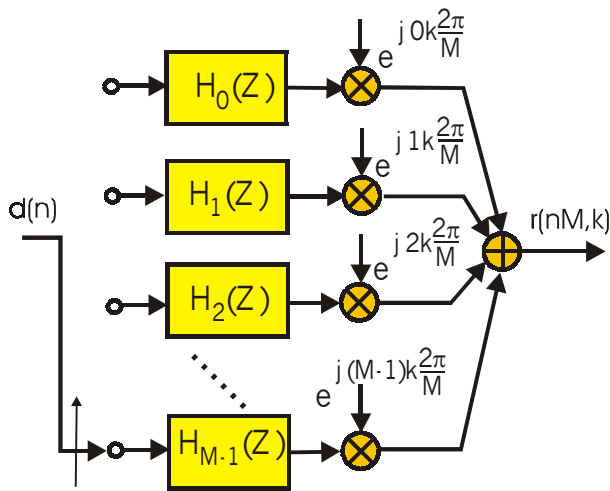


Figure 4. Re-Sampling Performed by Polyphase Filter Partition: Coefficient Heterodyne Becomes Fixed Phase Rotators per Path

The final step in forming the polyphase filter bank is recognizing the sum formed by the phase rotators is in fact one output port of a DFT, and that the DFT can be implemented as an FFT to extract time samples of each narrow-band process located at multiples of the output sample rate that has been aliased to baseband by the re-sampler. A representation of this perspective is seen in equation 7 and in figure 5.

$$y(nM, k) = \sum_{r=0}^{M-1} y_r(nM) e^{j \frac{2\pi}{M} kr} \quad (7)$$

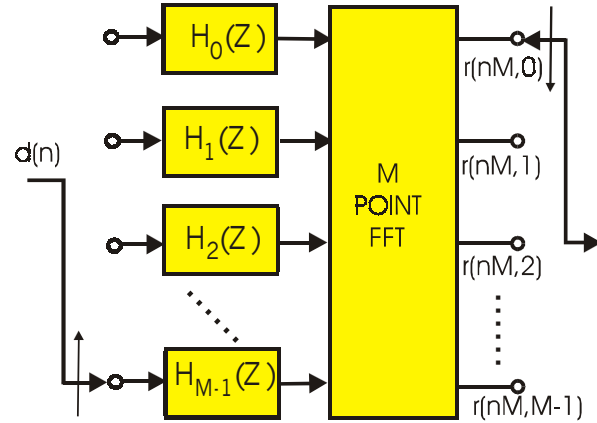


Figure 7. Polyphase Filter Bank: M-Path Polyphase Filter and M-Point FFT

As a specific example, suppose we have a composite signal with the spectrum shown in figure 8. The signal contains 7 band-pass signals on 100 kHz centers and has been sampled at 800 kHz. The input spectrum is shown on circle to emphasize the periodicity of the sampled data spectrum while the down converted and down sampled spectrum of one channel is shown on a line. An 8-stage polyphase filter and an 8-point FFT down convert all seven channels plus the unused 8-th channel of this spectrum. The output sample rate of each channel is a 100 kHz, the sample rate matched to the spacing of the channels. When the sample rate matches the spectral spacing we say the filter bank is maximally decimated.

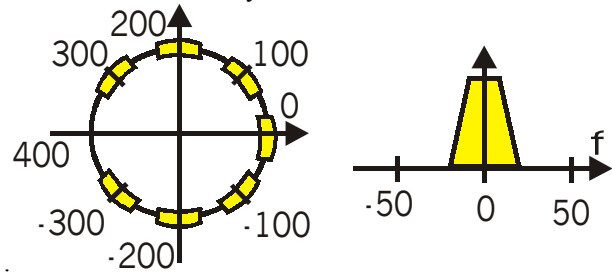


Figure 8. Input Spectrum and Output Spectrum for Single Down-Converted and Down-Sampled Channel

To satisfy Nyquist at the output sample rate, the spectral width of the filter must be less than the 100 kHz sample rate with the actual width otherwise arbitrary and determined by the coefficients of the polyphase filter.

2 NON-MAXIMALLY DECI-MATED FILTER BANK

In the example we just examined, if the symbol rate of the separate channels were precisely 50 k-symbols per second, the 100 kHz output sample rate would represent 2 samples per symbol, a very desirable relationship for subsequent processing such as timing recovery and equalization. The 8-to-1 down sample of the maximally decimated filter bank is perfect for the channel width and the channel spacing. The question arises, “what are our options when the desired output sample rate does not coincide with the maximally decimated sample rate”? Suppose for instance, a system with channel symbol rate of 80 k-symbols per sec for which the desired sample rate of 2-samples per symbol is 160 kHz. Or consider a second system with channel symbol rate of 40 k-symbols per sec, a rate for which the desired sample rate is 80 kHz. In the first example, the desired output rate is higher than the 100 kHz output rate while in the second example the desired output rate is lower than the 100 kHz output rate, which was determined by the channel spacing.

As a point of reference, we note that while the channel spacing and selected sample rate collude to determine the FFT size there we are not obliged to down sample the process in the polyphase filter and FFT by 8-to-1 just because the transform is of length 8. In the two examples cited we would like down sample the polyphase filter and FFT by 5-to-1 and by 10-to-1 respectively. The alternate solution is to use the system to down samples 8-to-1 and then interpolate each channel to the desired sample rate as a post-processing task. This option is shown in figure 9.

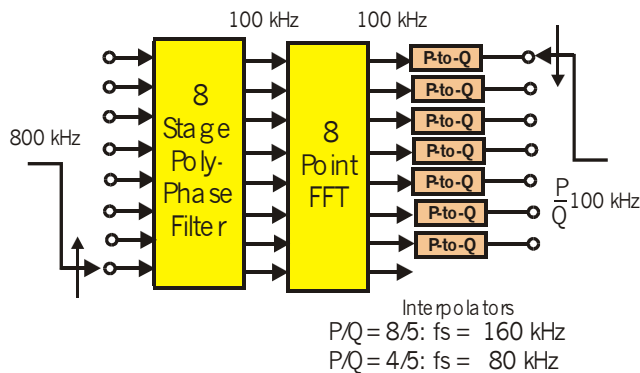


Figure 9 Resampling 100 kHz Polyphase Output Sample Rate From 100 kHz to 160 kHz or to 80 kHz

An alternate option is to embed the resampling in the polyphase commutator, in the interaction between input data registers and the polyphase coefficients, and in the

interaction between the polyphase outputs and the FFT input. This option has no computational cost, requiring only a state machine to schedule the interactions we are about to examine. We accomplish resampling in the polyphase engine by moving data through the two dimensional memory in stride of length 5 modulo-8. The least common multiple (LCM) of 5 and 8 is 40, from which we learn that the state engine cycles in 40 inputs, and since we deliver data 5-points at a time, there are 8-distinct states in the state machine.

We recall that the polyphase filter partition was applied to a prototype filter structure. In this structure we can resample by any integer, 5 in our example, by simply moving all data in the tapped delay line 5 addresses to the right and loading 5-new inputs before computing the next output. If we fold the one-dimensional tapped delay line into the two-dimensional memory of the polyphase filter we can visualize the data shift as a serpentine shift between columns, and the data load as the commutator delivering data to the first 5-registers of left most column. This shifting and loading sequence is shown in figure 10. We see that data moves down the first column in stride of 5 modulo-8 and when the modulo is invoked, the address wraps and shifts to the next column. This is standard residue mapping from a 1-D array to a 2-D array.

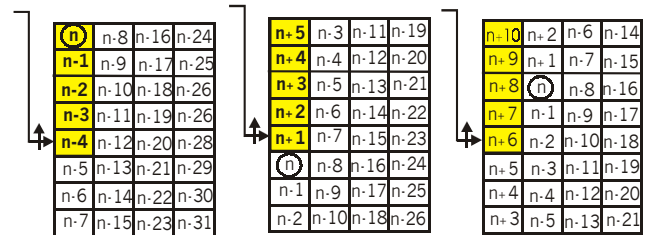


Figure 10. Successive Serpentine Data Shifts in Polyphase Memory and Data Load of 5-New Inputs for an 8-Stage Polyphase Filter

Data does not actually move by the serpentine shift but rather by circular wrapping of block memory, an address control task illustrated in figure 11. The block memory

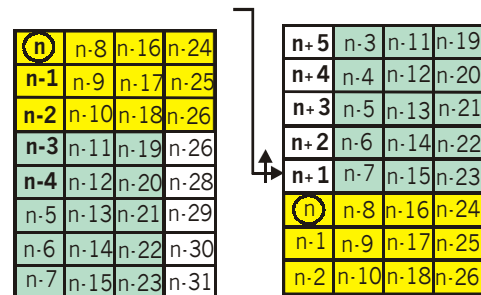


Figure 11. Circular Roll of Memory to Synthesize Serpentine Data Shift

shifts as follows: The top three rows (0, 1, & 2) of memory move to the bottom of the memory block making the remaining rows (3, 4, 5, 6, & 7) the top set. The data in these top rows all shift one sample to the right. The now empty first column receives the five new inputs required to compute the next output from the 8-filter paths. Remember that in this description of data memory management the filter coefficients are not moved from their original polyphase partition.

A final matter to address is that as data moves through the memory in stride of length 5-modulo-8 the data time origin precesses with respect to the FFT's time origin, which resets to the top of memory each time it is called. We keep the two origins aligned by circularly shifting the computed outputs of the polyphase filter by the residue address of the data time origin modulo 8. We have tagged the data time origin as the circled index "n" in figures 10 and 11. The time origin being cyclically shifted in memory is also periodic in the LCM of the 5 and 8, thus the cyclic shift of the polyphase path outputs has the same period as the memory shifts and is controlled by the same state machine controlling input memory shifts. The cyclic shift of the output buffer for one of the system states is shown in figure 12.

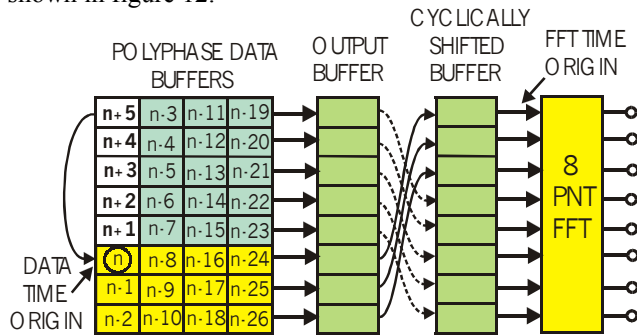


Figure 12 Cyclic Shift of Polyphase Output Buffer to Align Time Origin Of Cyclically Shifted Input Buffer With FFT's Reset Time Origin

Notice in figure 12 that the cyclic downshift of the three top rows to the bottom of the stack is undone by the cyclic up-shift at the output of the sub filters. Rather than cycle the data registers we anchor them and rotate the coefficient set. This equivalent mapping scheme converts the cyclic shift of the input array and the output buffer to a sliding cyclic load by the input commutator to a fixed set of registers and a cyclic shift of the coefficient memory. This is the process described in the state machine listed in table-1. We note that the Load Sequence is always to the next 5-registers where the indexing is performed modulo-8. Thus the next register to accept data as we leave state-1 and move to state-2 is R-1, which is actually R7. In a similar fashion the filter weights assigned to perform the inner products with the registers are always offset +5 modulo 8 relative to the previous filter set. Once the pattern is estab-

lished as we transition from state k to state k+1 it is easy to continue it to form the entire state machine.

| STATE | LOAD | FILTERS |
|-------|----------------|-------------------------|
| 0 | R4 R3 R2 R1 R0 | C0 C1 C2 C3 C4 C5 C6 C7 |
| 1 | R7 R6 R5 R4 R3 | C5 C6 C7 C0 C1 C2 C3 C4 |
| 2 | R2 R1 R0 R7 R6 | C2 C3 C4 C5 C6 C7 C0 C1 |
| 3 | R5 R4 R3 R2 R1 | C7 C0 C1 C2 C3 C4 C5 C6 |
| 4 | R0 R7 R6 R5 R4 | C4 C5 C6 C7 C0 C1 C2 C3 |
| 5 | R3 R2 R1 R0 R7 | C1 C2 C3 C4 C5 C6 C7 C0 |
| 6 | R6 R5 R4 R3 R2 | C6 C7 C0 C1 C2 C3 C4 C5 |
| 7 | R1 R0 R7 R6 R5 | C3 C4 C5 C6 C7 C0 C1 C2 |

Table-1 State Machine for Loading and Filtering Polyphase Filter for 5-to-1 Down Sample in 8-Point FFT

The process we just examined performed a 5-to-1 down sample in an 8-point polyphase filter, which down converted spectral regions from multiples of $f_s/8$ or 800/8 or 100 kHz and maintained a sample rate of $f_s/5$ or 800/5 or 160 kHz. This sample rate was selected to be 2-samples per symbol of channel symbol rate of 80 k-symbols/sec. Note that the transform determines the center frequencies but the commutator interacting with the polyphase filter determines the output sample rate. In a similar fashion we can down sample 10-to-1 in an 8-point polyphase filter to down convert the same center frequencies multiples of 100 kHz while maintaining an output sample rate of $f_s/10$ or 800/10 or 80 kHz, the sample rate appropriate for 2-samples per symbol of a channel symbol rate of 40 k-symbols/sec.

We now select an example to demonstrate simultaneous up sampling and down sampling in the same polyphase transform. Suppose the channel symbol rate is 60 k-symbols/sec for which we desire an output sample rate of 120 kHz. The required resampling ratio from the process is then 800/120 or 20/3. We can realize this ratio by first up sampling the input stream by 3 and then down sampling by 20. The up sampling is performed by zero packing the input data and the down sampling by serpentine shifting data through the filter in stride of length 20. This process is illustrated for two data load iterations in figure 13. There is

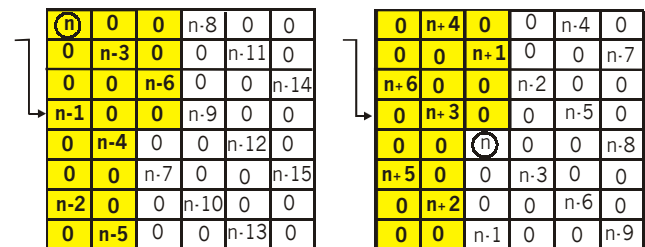


Figure 13. Successive Serpentine Data Shifts in Polyphase Memory and Data Load for a 20/3 Re-Sampling in an 8-Stage Polyphase Filter

no actual zero packing in the final configuration, we use it here to teach us how the (non-zero) data memory interacts with coefficient memory. Worthy of note is that in the first data load, the left figure, 7-actual data samples are delivered to the 20 register addresses, while in the second load 6-actual data samples are delivered to the 20 register addresses. The data loading procedure is found to be periodic in 6-load cycles for which we will require 6-states to control the process. Table-2 lists the memory loading instructions of the 6-states in a process that anchors the data registers and cycles the data load and coefficient sets. Note that in the 6-states, we deliver a total of 40 inputs and take from the polyphase engine 6 outputs to realize our desired embedded 40/6 or 20/3 resampling. The loading scheme is seen to be a constant offset of -3 modulo 8 within a sequence as well as in the transition between sequences. The -3 offset is a consequence of the 1-to-3 up sampling represented by the zero packing but not actually implemented in the process.

| STATE | NUMBER of INPUTS | LOADING SEQUENCE |
|-------|------------------|----------------------------|
| S0 | 7 | R2, R7, R4, R1, R6, R3, R0 |
| S1 | 6 | R5, R2, R7, R4, R1, R6 |
| S2 | 7 | R3, R0, R5, R2, R7, R4, R1 |
| S3 | 7 | R6, R3, R0, R5, R2, R7, R4 |
| S4 | 6 | R1, R6, R3, R0, R5, R2 |
| S5 | 7 | R7, R4, R1, R6, R3, R0, R5 |

Table-2 Loading Sequences for 6-State Control of 20-to-3 Resampling in 8-Stage Polyphase Filter

There are normally 8 filters in the polyphase partition for an 8-stage polyphase filter. Because of the 1-to-3 up sampling implemented by the zero packing, only on third of the weights in each stage actually contributes to the sub-filter output. Thus each stage is further partitioned into 3 sub sets of weights, which results in a total of 24 filter weight sets. These sets are denoted by C0, C1, ... , C23 where the integer is the starting index from the original non-partitioned prototype filter. Each filter starts with its index and increments in stride of length 24. Table lists the filter assignment to the 8-successive data registers for the 6-states of the process.

| STATE | FILTERS | | | | | | | |
|-------|---------|-----|-----|-----|-----|-----|-----|-----|
| S0 | C0 | C9 | C18 | C3 | C12 | C21 | C6 | C15 |
| S1 | C20 | C5 | C14 | C23 | C8 | C17 | C2 | C11 |
| S2 | C16 | C1 | C10 | C19 | C4 | C13 | C22 | C7 |
| S3 | C12 | C21 | C6 | C15 | C0 | C9 | C18 | C3 |
| S4 | C8 | C17 | C2 | C11 | C20 | C5 | C14 | C23 |
| S5 | C4 | C13 | C22 | C7 | C16 | C1 | C10 | C19 |

Table-3 State Machine for Filtering Coefficient Sets in 20-to-3 Resampling in 8-Stage Polyphase Filter

We note from the table that in a given state the successive filter index increments by 9 modulo-24 and between states, the filter index increments by -4 modulo-24. The integer 9 is the offset between two data samples in the zero-packed load in two adjacent rows. The -4 index is the same as a $+20$ index which is the number of zero-packed data points introduced per data load cycle.

The process we just examined performed a 20-to-3 down sample in an 8-point polyphase filter, which down converted spectral regions from multiples of $f_s/8$ or 800/8 or 100 kHz and maintained a sample rate of $f_s \cdot 3/20$ or 2400/20 or 120 kHz. When we designed the process we up sample the data by a factor of three on the way into the filter so that the prototype filter has to be designed to operate at 3 f_s or 2400 kHz. Consequently, the filter becomes three times longer than the standard design but since we only use one-third of it per processing cycle we pay no processing penalty.

There is a little bookkeeping quirk to which we must direct our attention. The 1-to-3 zero-packed signal presents 3 spectral copies to the processing stream of filter and FFT. Figure 14 indicates the frequency indices of the spectra prior to up sampling and after the 1-to-3 up sampling. When we examine the spectra of the up sampled data at the frequencies that correspond to the pre-up sampled locations, i.e. multiples of $\pi/8$ we find different spectral centers. For instance when we look at the spectra at $\pi/8$ at the input rate we see frequency "1", but at the output rate we see frequency "3". The spectral locations are reordered as a result of processing the up-sampled data in the polyphase filter. This reordering is an expected result and is seen in the Good-Thomas (or Prime Factor) algorithm as the result of residue addressing of a two dimensional array, which is exactly what we are doing here. Thus the 8-point FFT processing the polyphase data outputs frequencies in the order [0, 3, 6, 1, 4, 7, 2, 5], which is seen to be indexing stride of 3 modulo-8. Our response to the observed rearranged indices is to reorder them back to their natural order.

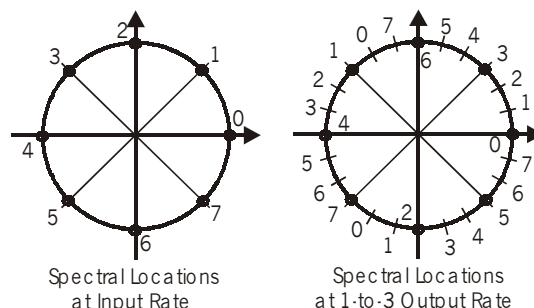


Figure 14. Positions of Spectral Regions at Input Rate and at 1-to-3 Up Sampled Output Rate

3-DEMONSTRATION OF PROCESSES

We now demonstrate the performance of an 8-stage polyphase filter and DFT operating as an 8-channel channelizers. These demonstrations match the operating conditions described earlier in this paper, 8-channels at 100 kHz centers operating at 800 kHz sample rate. In the first mode 6 of the 7 available channels are occupied by 50 kHz bandwidth signals. The channel at -100 kHz is empty. The input data is channelized and down sampled 8-to-1 for an output rate of 100 kHz by the system operating in the standard maximally decimated mode. The 8-polyphase filter stages are each 8-taps that are anchored to the 8-input registers being fed by the periodic input commutator. Figure 15 presents the spectrum of the input signal and the spectra of the 7 output channels at 100 kHz output rate.

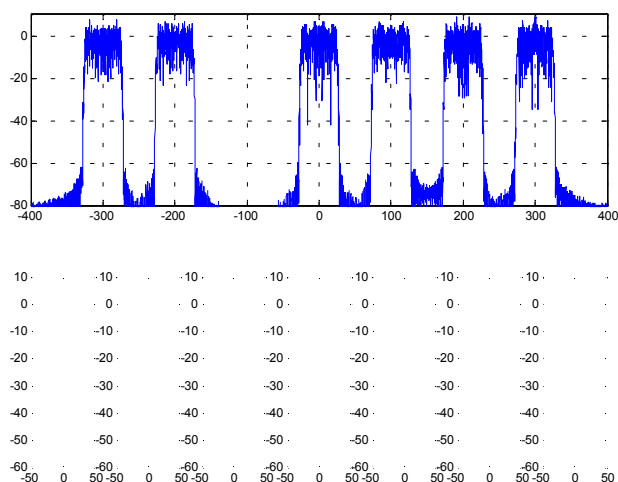


Figure 15. Spectra of Composite Input Signal at 800 kHz Sample rate and of Channelized Output Signals at 100 kHz Sample Rate

The signal structure in the second mode is similar to the first except that the occupied channels have 60 kHz bandwidth. The input data is channelized and down sampled 20-to-3 for an output rate of 120 kHz by the system operating in the non-maximally decimated mode. Due to the narrower transition bandwidth, the 24-polyphase filter stages are each 12-taps. The filters are periodically rotated through the 8-input registers in accord with the state-machine described in Table-3. The periodic sliding input commutator described in Table-2 feeds the 8-data registers. Figure 16 presents the spectrum of the input signal and the spectra of the 7 output channels at 120 kHz output rate. As described earlier, we have rearranged the output spectra so they appear in natural order. We note this is the same processing engine used in the previous demonstration. All we have done to it is control it with a simple state machine. In the two demonstrations, the sample rate was

selected to be twice the channel bandwidth rather than the channel separation.

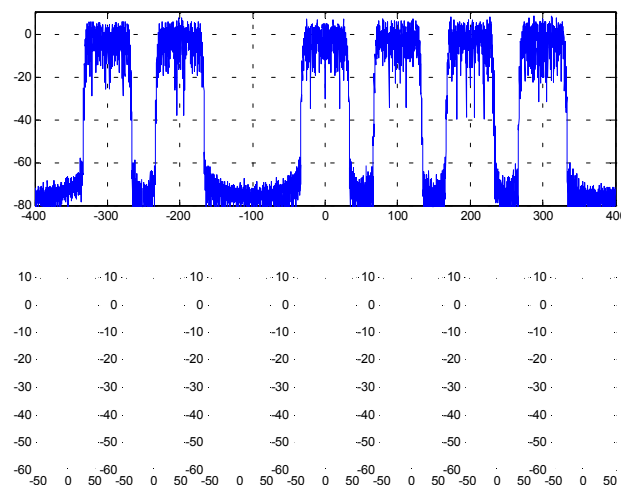


Figure 16. Spectra of Composite Input Signal at 800 kHz Sample rate and of Channelized Output Signals at 120 kHz Sample Rate

4-CONCLUSIONS

We remind the reader that the polyphase channelizer is a remarkably versatile engine. In it, the transform and the number of stages control the center frequencies of the process, the filter weights control the bandwidth of the process, and the cyclic sliding commutator and cyclic sliding weight sets control the resampling ratio, hence output sample rate of the process. The three functions shown in figure 1, frequency selection, bandwidth reduction, and sample rate reduction are totally uncoupled even when implemented in the polyphase partitioned filter and FFT structure.

5- REFERENCES

- [2] Robert Lowdermilk, Dragan Vuletic, and fred harris, "Smart Radios Using Multirate Filters to Process, Identify, and to Reconfigure Itself to Receive and Demodulate Multi-Standard IRIG Signals", 2002-Software Defined Radio Technical Conference, San Diego, 11-12 November 2002.
- [3] fred harris, "A Fresh View of Digital Signal Processing for Software Defined Radios, Part I" and "Part II". International Telemetry Conference, San Diego, CA, 21-24, October 2002
- [5] fred harris, "On Multichannel Receivers with Arbitrary, Uncoupled Selection of Channel Bandwidth, Channel Spacing, and Channel Sample Rate", 2002 International Symposium on Advanced Radio Technologies (ISART), Mar. 4-6, 2002, Boulder Colorado