POLYPHASE FILTERS – A MODEL FOR TEACHING THE ART OF DISCOVERY IN DSP

Mark L. Fowler

Dept. of Electrical and Computer Engineering
State University of New York at Binghamton
mfowler@binghamton.edu

ABSTRACT

By its very nature DSP is a mathematically heavy topic and to fully understand it students need to understand the mathematical developments underlying DSP topics. However, relying solely on mathematical developments often clouds the true nature of the foundation of a result. It is likely that students who master the mathematics may still not truly grasp the key ideas of a topic. Furthermore, teaching DSP topics by merely "going through the mathematics" deprives students of learning the art of discovery that will make them good researchers. This paper uses the topic of polyphase decimation and interpolation to illustrate how it is possible to maintain rigor yet teach using less mathematical approaches that show students how researchers think when developing new ideas.

Index Terms— DSP education, discovery, polyphase filters, decimation and interpolation

1. INTRODUCTION

Teaching DSP necessarily requires heavy use of mathematics – the nature of the material requires mathematics to precisely specify the methods and firmly establish their characteristics and performance. Furthermore, mathematics enables efficient descriptions that help authors provide precise details despite length constraints imposed by book and journal editors. Thus, it is not surprising that DSP educators rely heavily on using mathematics when teaching DSP – and perhaps too often our use of mathematics, although precise, fails to convey the true essence of the topics. Nonetheless, our students persevere and learn the material – but likely in the process fail to learn the art of discovery needed to become creative DSP researchers. DSP educators should strive to present material in ways that simultaneously demonstrate the art of discovery in DSP and present the mathematical results and insights the students need to learn.

The back cover of one of Richard Hamming's books [1] states that the book is "…intended to instill in the reader a style of thinking that will enhance his ability to function as a problem solver of complex technical issues…" The book describes DSP-related developments and "…relates how those discoveries came about, and most importantly, provides analysis about the thought processes and reasoning that took place…" [1]. Another example in the literature that shows the thought process that lies behind something that is otherwise developed mathematically is a description of how the Hough transform was invented [2]; in fact that paper is part of the ongoing "DSP History Column" in the *IEEE Signal Processing Magazine*, which has focused at least partly on *how* DSP results were discovered. Likewise, here we take a specific DSP idea (i.e., polyphase filters) that ordinarily is presented solely in terms of mathematical developments and instead develop it from a more intuitive point of view. This approach allows students to see an example of the thought process used in DSP research – namely that innovation (even in math-heavy areas) more often flows from non-mathematical visualization/intuition than from blindly manipulating equations. Furthermore, this particular development of polyphase filters provides students with a clearer view of exactly what polyphase filters are and how they work – then going through the mathematical development will make more sense.

For conciseness in this paper we will focus only on FIR polyphase filters for decimation and interpolation by an integer factor. This is sufficient to show discovery in action and to show that polyphase filters are essentially a re-configuration that allows convenient computing. We close the paper with some general comments about teaching discovery techniques in DSP classes.

2. STANDARD PRESENTATION OF POLYPHASE

General DSP books (e.g., [3],[4]) and multirate DSP books (e.g., [5]-[7]) generally develop decimation polyphase filters as follows. First, the idea of filter-then-decimate is introduced. Namely, the signal to be decimated, $x[n]$, is first filtered by a filter with impulse response $h[n]$ to give the intermediate signal $v[n]$ given by

$$v[n] = \sum_i x[i]h[n-i] \qquad (1)$$

and then that filtered signal is decimated by an integer factor M to give the lower rate signal $y[n]$ as

$$y[n] = v[nM] = \sum_i x[i]h[nM - i], \qquad (2)$$

which is simply (1) with n replaced by nM, to enact the decimation. In such developments it is then pointed out that although this structure accomplishes the desired goal it is computationally inefficient.

Up to here the development is intuitive and instructive. However, at this point the polyphase structure is then developed one of two ways, neither of which provides much insight or understanding – even when fully understood. The first way is a time-domain development that uses a non-obvious re-indexing of the summation in (2) given by

$$i = i'M + m, \quad \text{with } i' \in \mathbb{Z}, \quad m = 0, 1, \ldots, M-1, \qquad (3)$$

which, after some mathematical manipulation of double summations, leads to the desired polyphase filter structure. The second way is a z-domain development that starts by first demonstrating that the filter transfer function can be re-organized into a sum of the polyphase component transfer functions, as given by

$$H(z) = \sum_{m=0}^{M-1} z^{-m} P_m(z^M), \qquad (4)$$

with $P_m(z) = \sum_n h[nM + m]z^{-n}$. The development is then completed by multiplying this form of $H(z)$ by $X(z)$, applying the z-domain result for decimation, and finally exploiting the noble identities (also called the multirate identities) to move the decimation to the front of the resulting system.

Both of these developments lead to the same polyphase structure but neither gives much insight into what is really going on nor what the resulting structure really does. This leaves students with a shallow understanding of the real essence of polyphase filters and likely unprepared to apply them in any new developments of their own. Furthermore, it gives the student no insight into *how* the development was conceived or discovered. In the first development, the introduction of the re-indexing in (3) seems to "come out of nowhere" and leaves many fundamental questions unanswered. Why is *this* re-indexing used? How did those who first developed these ideas *ever* come up with such an idea? In the second development, the expansion in (4) also seems to "come out of nowhere" and the need to apply the noble identities further shrouds what is really going on. Furthermore, the application of the noble identities leads to (imprecise) statements that "the decimation is now being done before the filtering," which seems contradictory to what they learned first (i.e., filter-then-decimate, as in (2)).

Each approach hides from students what is really going on and gives them the impression that to become DSP innovators they must learn to pull such confounding steps "out of thin air." Perhaps even worse, the resulting structure is touted as an efficient implementation, yet the development gives no insight into where that efficiency comes from (except perhaps that the parallel paths that result admit the potential for parallel implementation, but that is only part of the story). Alternatively, it is desirable to find a development that (i) flows transparently from (2) to the final polyphase structure, (ii) clearly shows where the efficiency in the structure really comes from, and (iii) illustrates how researchers think conceptually/intuitively to arrive at precise mathematically results. Point (iii), obviously, applies to topics beyond the specific coverage of polyphase filters and should be done more in DSP textbooks and class notes. By showing how (iii) can be accomplished for the topic of polyphase filters, hopefully this paper will motivate ways to address point (iii) for other topics. Some discussion of this is given in the last section of the paper.

3. DISCOVERY OF POLYPHASE FILTERS

When DSP researchers seek to discover new ideas it is common to use visual methods rather than just using mathematics. This is clear when glancing at researchers' office whiteboards. For the topic at hand we can start by going back to the inefficient method shown in (2) and visualizing what it says; Figure 1 shows such a visualization. Along the top are the samples of the signal x indexed by i as in (1) and (2). Under that are the various flipped and shifted versions of the impulse response (labeled along the left side with the appropriate $h[n - i]$); the arrows on the right indicate the act of assigning the results of multiplying that row's shifted-flipped impulse response by the $x[i]$ above and summing to create the corresponding $v[n]$. Note that all the possible shifts are shown for $n = 6$ to 11. That is, the figure even shows the shifts in (1) that are ultimately thrown away in (2), which are the ones that have their corresponding $v[n]$ crossed out.

The key insight here (and one that is obvious) is that for efficiency one should not compute the samples that are to be thrown away by the decimation. And that is all that is needed to discover the fundamental insight needed to derive polyphase filters! The top part of Figure 2 retains only those outputs that are kept by the decimation; to aid in visualization we have used color-coded symbols for the samples of the impulse response rather than the math symbols used in Figure 1. Now notice that we could implement this simply by revising our view of convolution: rather than shifting the flipped impulse response ahead by 1 prior to computing each output we now shift it ahead by M each time. This is an efficient way to compute the filter and decimate; but, it is not desirable from an implementation point of view: we can't use standard software code for convolution nor could we use standard hardware blocks for convolution, because they use single-step shifting for convolution. Furthermore, this shift-by-M convolution view is not easily applicable to the implementation of recursive filters for decimation.

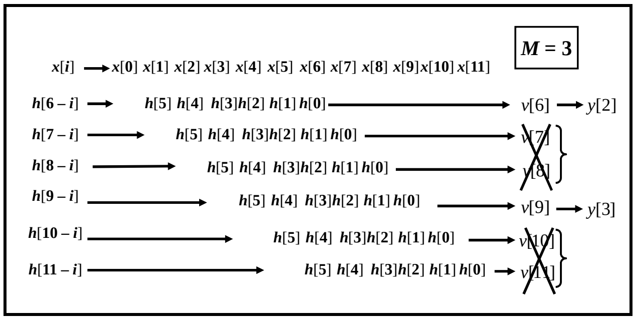
M = 3
x[i] → x[0] x[1] x[2] x[3] x[4] x[5] x[6] x[7] x[8] x[9] x[10] x[11]
h[6 – i] → h[5] h[4] h[3]h[2] h[1] h[0] → v[6] → y[2]
h[7 – i] → h[5] h[4] h[3]h[2] h[1] h[0] → v[7]
h[8 – i] → h[5] h[4] h[3]h[2] h[1] h[0] → v[8]
h[9 – i] → h[5] h[4] h[3]h[2] h[1] h[0] → v[9] → y[3]
h[10 – i] → h[5] h[4] h[3]h[2] h[1] h[0] → v[10]
h[11 – i] → h[5] h[4] h[3]h[2] h[1] h[0] → v[11]

Figure 1: Inefficient form of filter-then-decimate. Illustrated for the case of decimation factor *M* = 3.

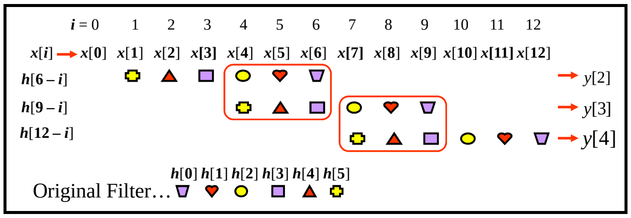
i = 0 1 2 3 4 5 6 7 8 9 10 11 12
x[i] → x[0] x[1] x[2] x[3] x[4] x[5] x[6] x[7] x[8] x[9] x[10] x[11] x[12]
h[6 – i]
h[9 – i]
h[12 – i]
y[2]
y[3]
y[4]
h[0] h[1] h[2] h[3] h[4] h[5]
Original Filter…

Figure 2: Efficient structure from Figure 1 obtained by eliminating outputs that are discarded by decimation.

But this viewpoint makes it clear that there is a definite structure in Figure 2 that can be exploited. Looking carefully (what good researchers do when exploring to find new results) we see that the purple symbols (trapezoid and square) only interact with signal samples that have an index that is a multiple of 3; Thus we see that the purple symbols taken together as a two-point impulse response are convolved in a standard way (i.e., requiring shifts of 1 sample) with the sequence of input samples having an index that is a multiple of 3. Similar relationships can be observed for the yellow symbols (circle and cross) and the red symbols (heart and triangle). This view is highlighted by the rounded rectangular shapes. It leads to the polyphase structure shown in Figure 3, where the input signal gets “dealt” into *M* sub-signals and the filter impulse response gets split into *M* sub-filters – and most importantly each sub-signal is input to each sub-filter for *standard* convolution (i.e., requiring only single-step shifting). Thus, this structure can be easily implemented using standard software/hardware building blocks and furthermore the sub-structures have potential to be extended to more general cases.

Thus, without the use of any precise-yet-obfuscating mathematics it is possible to precisely develop the basis of polyphase filters. The important thing here is that the development shows that the result is not a complicated mathematical development but rather just a re-grouping of the structure of the inefficient filter-then-decimate viewpoint shown in Figure 1 into the efficient polyphase form shown in Figure 3.

Once this simple re-structuring provides the key idea behind polyphase decimation for the FIR case it is possible to (i) derive the result using the time-domain re-indexing approach and the z-domain polyphase decomposition/noble identities approach and (ii) extend the idea to IIR polyphase ideas. Doing such derivations *after* seeing this development will make it more obvious to the student why such a re-indexing or filter decomposition is used.

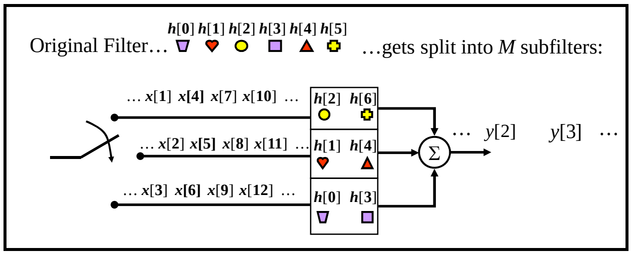

Figure 3: The resulting polyphase decimation structure obtained by re-interpreting the structure in Figure 2.

Of course it is possible to develop polyphase interpolation structures using similar approaches. There the insight comes from the fact that the effect of the inserted zeros on the convolution can be exploited by re-grouping the structure. Figure 4 shows the signal $\tilde{x}[i]$, which is the input signal with inserted zeros, as it is filtered using the impulse response $h[n]$. The tall dashed rectangles show the multiplications that need not be performed. The solid-filled symbols of the impulse response need to be multiplied and the large solid-lined rounded rectangles collect those into groups that all operate on the same subset of input samples.

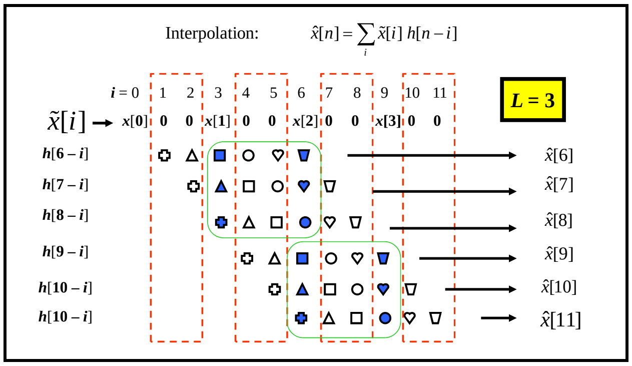
Interpolation: $\tilde{x}[n] = \sum_i \tilde{x}[i]\, h[n-i]$
L = 3
i = 0 1 2 3 4 5 6 7 8 9 10 11
$\tilde{x}[i]$ → x[0] 0 0 x[1] 0 0 x[2] 0 0 x[3] 0 0
h[6 – i]
h[7 – i]
h[8 – i]
h[9 – i]
h[10 – i]
h[10 – i]
$\tilde{x}[6]$
$\tilde{x}[7]$
$\tilde{x}[8]$
$\tilde{x}[9]$
$\tilde{x}[10]$
$\tilde{x}[11]$

Figure 4: Inefficient form of zero-stuff-then-filter form of interpolation.

The result of eliminating from Figure 4 the operations that involve only the inserted zeros is shown in Figure 5, where it is clear that the original signal (i.e., without inserted zeros) gets filtered with *M* different sub-filters to create *M* successive output samples and then get shifted ahead to the next non-zero sample. Figure 5 is precisely the structure of FIR polyphase interpolation, and again we see the same sub-filter structure (the polyphase filters) come into play – although in a slightly different way. It now makes sense to derive the results mathematically and it will be obvious why the mathematical steps are taken.

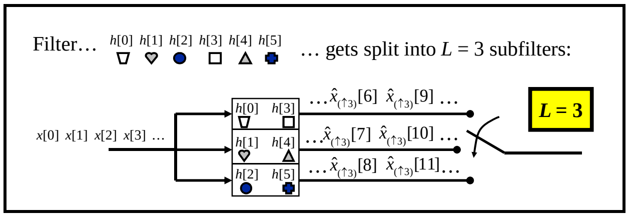
Filter…
h[0] h[1] h[2] h[3] h[4] h[5]
… gets split into L = 3 subfilters:
x[0] x[1] x[2] x[3] …
h[0] h[3]
h[1] h[4]
h[2] h[5]
…$\hat{x}_{(\uparrow 3)}[6]$ $\hat{x}_{(\uparrow 3)}[9]$ …
…$\hat{x}_{(\uparrow 3)}[7]$ $\hat{x}_{(\uparrow 3)}[10]$ …
…$\hat{x}_{(\uparrow 3)}[8]$ $\hat{x}_{(\uparrow 3)}[11]$…
L = 3

Figure 5: The resulting polyphase interpolation structure obtained from Figure 4.

4. TEACHING DISCOVERY IN DSP

In order to extend to other DSP topics the ideas of teaching discovery that have been illustrated above, we need to identify what characteristics in the above descriptions are general approaches that can be carried over. Looking back at how the discovery of polyphase filters was made in Section 3 we see that there were three main components:

1. Visualize the problem
   - Block Diagrams
   - Sketches of simple plots of functions, etc.
2. Start with small examples
3. Look for patterns and familiar structures

Equation (2) was visualized for a small example using block diagrams and signal sketches. Then the familiar pattern of individual convolutions was discovered.

Although not present in the polyphase filter discovery given above, there are other components that can be useful for discovery. A few of them are:

4. Start with simplified assumptions then remove them
   - For example, in developing uniform DFT filter banks first assume that the filter length is the same as the number of channels [8]
5. Use analogies to relate new ideas to topics already covered
   - For example, in developing the idea of I & Q signals it is useful to first explore the idea using the familiar concept of phasors [8]
6. Divide and conquer
   - For example, deriving the FFT algorithm is often done using this approach.

Here are some other DSP topics that can be used as vehicles for teaching the art of discovery.

Uniform DFT Filter Banks [8]: This uses components 1 – 4 by visualizing a small example under the simplified assumption that the filter length equals the number of channels and that there is no decimation. Once this scenario is understood through visualization, the mathematical view is extracted and it is possible (through continued visualization) to successively remove assumptions to discover the uniform DFT filter bank with decimation and filter length not equal to the number of channels.

Bandpass Sampling [8]: This uses components 1 – 4 by visualizing a small example under the simplified assumption that the upper band edge is an integer multiple of the signal's bandwidth. Then, as understanding is gained, the discovery progresses to the general case of an upper band edge that is not an integer multiple of the bandwidth.

DFT & FT Results: An example is the proof of

$$\sum_{n=0}^{N-1} e^{j2\pi kn/N} = \begin{cases} 0, & k = 2, 3, \ldots, N-1 \\ N, & k = 0 \end{cases} \qquad (5)$$

The proof is commonly available in DSP books (e.g., [4]) but, again, students may wonder how one *discovers* such results (it is easy to prove, but may be elusive to *discover*). However, visualizing this as the sum of vectors in the complex plane helps with the discovery: when k = 0 all N vectors in the sum line up on the positive real axis and therefore sum to N, but for the other values of k you get N vectors that are uniformly spaced in angle and thus sum to 0. Visualization together with looking for patterns in small examples leads to this discovery.

REFERENCES

[1] R.W. Hamming, *The Art of Doing Science and Engineering: Learning to Learn*. Gordon and Breach, Amsterdam, 1997.

[2] P.E. Hart, "How the Hough Transform Was Invented," *IEEE Signal Processing Magazine*, DSP History Column, pp. 11 – 22, Nov. 2009.

[3] J.G. Proakis and D.G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications, 4th Ed.*, Pearson Prentice Hall, Upper Saddle River, NJ, 2007.

[4] B. Porat, *A Course in Digital Signal Processing*, John Wiley and Sons, New York, 1997.

[5] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[6] N.J. Fliege, *Multirate Digital Signal Processing: Multirate Systems, Filter Banks, Wavelets*, John Wiley and Sons, West Sussex, England, 1994.

[7] f.j. harris, *Multirate Signal Processing for Communication Systems*, Prentice Hall, Upper Saddle River, NJ, 2004.

[8] M.L. Fowler, Lecture Notes for EECE521, Binghamton University, available at http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE521.htm