Approximately Constant Group Delay FIR Filter Designs

John W. Pierre

Electrical and Computer Engineering Department

University of Wyoming

Abstract: Digital filtering is a well established topic in any undergraduate DSP (digital signal processing) course. DSP is frequently a popular elective course in electrical and computer engineering. In the area of FIR (Finite duration Impulse Response) filter design, many design techniques are frequently discussed including windowing, frequency sampling, and Parks/McClellan methods. Usually, only symmetric and antisymmetric designs are discussed since they result in filters with constant group delay. Students frequently assume all FIR filters have constant group delay, but this is only true when the impulse response has symmetry characteristics. This paper reformulates the frequency sampling technique for a more intuitive design methodology which includes the ability to design filters that do not possess symmetry but have approximately constant group delay over the frequency range of the passband. While these filters do not have constant group delay, they can have lower average group delay than their symmetric filter counterparts and thus faster response times which may be important in some applications. In this paper, the design technique is presented and examples provided. Being a fairly straightforward design method, this is a good way to introduce the topic before going to more advanced techniques such as the Parks/McClellan method which includes an option to also design FIR filters that do not possess symmetry characteristics.

I. INTRODUCTION

A Digital Signal Processing (DSP) course is usually a popular elective course in most undergraduate Electrical and Computer Engineering degree programs. Teaching all the necessary content for an undergraduate DSP course is challenging. There are many topics to be covered and balancing how much time to spend on each topic relative to its importance is not an easy task. Digital filtering is usually one of the topics covered in such a course with the time divided between Finite duration Impulse Response (FIR) filters and Infinite duration Impulse Response (IIR) filters. This paper discusses introducing additional FIR filter topics in a time efficient manner that is pedagogically easy to follow for the students.

Most digital signal processing undergraduate text books cover a number of FIR filter design methods. Usually the discussion is limited to linear-phase FIR Filters. Almost every text book [1-4] covers the Window design method and the Parks/McClellan method. The Window method is usually covered in some detail because of its ease of understanding and straightforwardness of the design. The Parks/McClellan method is a widely used FIR filter design method because of its optimality in many applications. The level of detail for the Parks/McClellan method covered in the text books varies because of the level of complexity of the algorithm and the amount of time it takes to cover the details. Many texts point toward computer tools such as MATLAB to design Parks/McClellan filers. Probably the next most commonly discussed FIR filter design method in undergraduate texts [1,2] is the Frequency Sampling Method. Like the Window method, the Frequency Sampling method is straightforward to understand.

When discussing FIR filters, almost all of the texts focus on linear phase filter designs with only a slight mention of non-linear phase FIR filters or minimum phase FIR filters. By far the most common applications for FIR filters are linear phase FIR filters because of the benefit of constant group delay which is the result of linear phase. Thus the focus of time on FIR filtering should be spent there. But frequently students walk away from a DSP course only knowing linear phase FIR filters. The great emphasis on linear phase FIR filters sometimes causes students to mistakenly think all FIR filters are linear phase.

If only the magnitude response is important, IIR filters frequently become the preference. But there are applications where non-linear phase FIR filters are beneficial. The group delay of a linear phase filter is half the model order. As discussed in [5], there are applications of FIR filters where it is more important that the delay be small than for the phase to be perfectly linear. Karman and McClellan extend the standard Parks/McClellan algorithm to the general complex case which includes the case for non-linear phase FIR filters which only have approximately constant group delay [5-7].

This paper looks at introducing students to lower group delay FIR filter designs that are only approximately linear phase over the passband of the filter. The approach to introducing students to such filters is a direct extension of the Frequency Sampling method and software tools are available in MATLAB for designing complex Parks/McClellan Filters with reduced group delay. The time spent to introduce these topics is less than one period and one homework problem.

Section II of the paper discusses the approach to introducing students to approximately constant group delay FIR filters by a direct extension of the Frequency Sampling design method. The section then goes on to describe the application of what is known as the complex Parks/McClellan algorithm and how to use the command in MATLAB to specifically design approximate constant group delay filters which are more optimal than the simple direct extension of the frequency sampling method. Section III gives a detailed example of designing these approximately constant group delay filters that can be used as an example in class or as a homework problem. Section IV concludes the paper.

II. APPROXIMATE CONSTANT GROUP DELAY FIR FILTER DESIGNS

This section describes the modified frequency sampling method that results in an approximately linear phase filter with less group delay than the linear phase filter designed by the frequency sampling method. The method is only explained in the context of a modification of Type-I FIR filters which are FIR filters of odd filter length and even symmetry. The approach is easily extended to the other filter types but for the sake of time in class it is just discussed for Type-I. Also, this is just a stepping stone to discuss approximate constant group delay Park/McClellan filters.

First the Frequency Sampling method is taught in an intuitive manner that leads naturally to the design method for approximately constant group delay filters. At this point in class the students are familiar with the DFT (Discrete Fourier Transform) and the IDFT (Inverse DFT). It is fairly straightforward to explain to students the following design strategy, which is at the heart of the frequency sampling method. Specify equally spaced samples of the filters frequency response H(k) and take the IDFT to give the filters impulse response h(n). This guarantees that the frequency response will match the desired frequency response at equally spaced points. Which at first to the students seems to be ideal, but the students are then asked what happens with the frequency response between those sample points. They realize that the frequency response only matches the ideal at those sample points and that there is ripple in the response between those points. The use of transition points is discussed as a means of reducing the ripple. In designing a Type-I low-pass filter for example, the magnitude of the frequency response, |H(k)| is specified as unity over the passband and zero in the stopband with the possibility of one or two transition points between the passband and stopband. The phase on the other hand is specified to be linear with a slope corresponding to the negative of constant group delay of a linear filer which equals (L-1)/2 where L is the filter length.

At this point it is straightforward to extend the method to filters with only approximately constant group delay but lower group delay then the constant group delay filters. The students are asked what would happen if instead of specifying the phase slope to correspond to -(L-1)/2, that the phase slope be specified to be less steep, say -(L-1)/4. When the IDFT is taken, the resulting impulse response will still have a frequency response that goes through the specified magnitude and phase points but will have some ripple in the phase as well as in the magnitude. If the ripple in the phase is not to large, the filter will be approximately constant group delay.

When the Parks/McClellan filter design is explained in class, it can be discussed that it has been modified [5-7] to allow for approximate constant group delay filters. Since Parks/McClellan is optimal in the sense of ripple, this is a preferred method for designing approximate constant group delay filters. The next section gives an example of designing approximate linear phase filters with less group delay than their linear phase counterparts.

III. EXAMPLE FILTER DESIGN EXERCISE

In this section an example exercise is provided illustrating the extension of linear phase filter designs to ones with approximately linear phase but with substantially reduced group delay. The example could be used as an illustrative example in class or as a homework problem. The example compares a low-pass filter designed using four methods: the Frequency Sampling method with constant group delay, Frequency Sampling method with approximately constant group delay, Parks/McClellan method with constant group delay, and Parks/McClellan method with approximate constant group delay. The filter is to be low-pass with a cut-off frequency of $8\pi/17$ and a filter length of L=17. Thus for the constant group delay filters, the group delay is (L-1)/2=8 samples. For the approximate constant group delay filters, the group delay is specified to be approximately 4 samples which is half as much as the constant group delay filters. For the Frequency Sampling filters a single transition point are used with a value of 0.4 [8]. The weighting used for the Parks/McClellan algorithms is chosen such that the ripple in the passband is approximately equal to the ripple in the passband for the Frequency Sampling designs. That passband ripple is approximately 0.6 dB.

The impulse responses corresponding to the four designs are shown in Figure 1. As expected the impulse responses for the constant group delay filters are symmetric. These two impulse responses are very similar. The approximate constant group delay filters impulse responses are more front in loaded as one would expect to get a smaller group delay.

Figure 2 shows the group delay for the four designs. As expected the linear-phase Frequency Sampling filter and the Parks/McClellan filter have a group delay of exactly 8 samples. The other two designs have group delays that are not constant but that are close to 4 samples over the passband frequency range for which they were designed. The non-constant group delay Parks/McClellan filter is between 3.6 and 4.4 samples over most of the passband, while the non-constant group delay Frequency Sample filter's is between 3.5 and 4.2 samples over most of the passband. Both filters group delay increases near the cut-off frequency. The unwrapped phase for the four filter designs is shown in Figure 3. As expected the two constant group delay filters have linear phase. The two approximately constant group delay designs have approximately linear phase over the passband.

Figure 4 shows the magnitude response of the four filters. All have approximately the same passband ripple and transition bandwidth. As expected the constant group delay Parks/McClellan filter has the largest stopband attenuation of 48.9 dB. Being suboptimal to the Parks/McClellan filter, the frequency sampling filter has a similar magnitude response but with a stopband attenuation of 41.9 dB. The approximate constant group delay Parks/McClellan filter has a stopband attenuation of 42.8 dB, so while the attenuation is not as large as the constant group delay Parks/McClellan filter, the group delay is half as much. As expected the approximate constant group delay Frequency Sampling filter has the poorest stopband attenuation of 19.9 dB. But the value in this filter is a pedagogical step in a better appreciation of non-linear phase filters and the possibility of reducing the group delay. The MATLAB code for generating the filter designs is provided in Figure 5. The code is broken out into sections for each of the four designs.

IV. CONCLUSION

This paper focuses on a method to introduce students to approximately constant group delay FIR filter designs which have lower group delay than their perfectly constant group delay FIR filter counter parts. In an introductory DSP course when discussing FIR filtering, rarely is there time to introduce students to anything but linear-phase (constant group delay) filters. The objective is to be able to introduce students to non-linear phase FIR filters in a time-efficient manner by utilizing a direct method to modify the Frequency Sampling filter design method to produce filter designs with lower group delay but such that the group delay is only approximately constant. This then provides a nice springboard to discuss these approximate constant group delay filters when discussing the Parks/McClellan design algorithm and to introduce the students to a modification of the original Parks/McClellan algorithm to allow for these designs.

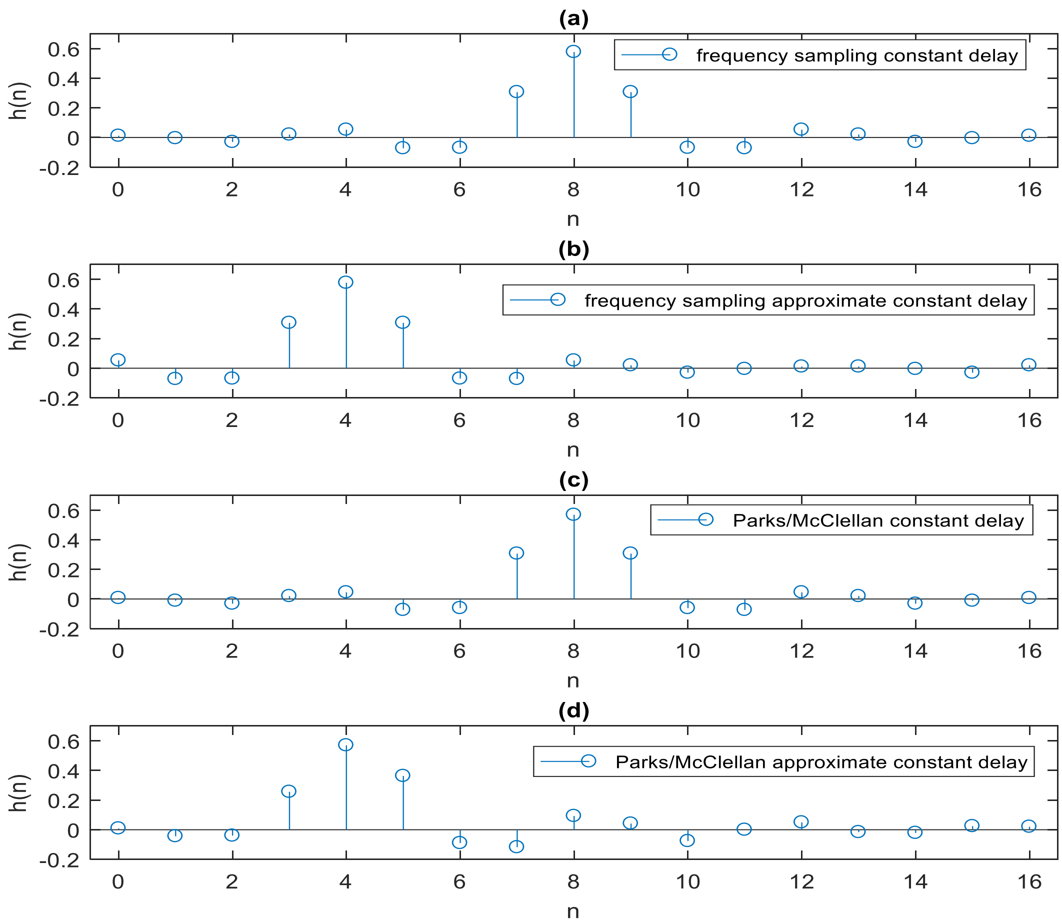

Figure 1: Impulse Response of the four filter designs: (a) Frequency Sampling Design with linear phase (constant group delay); (b) Frequency Sampling Design with approximately linear phase (approximately constant group delay); (c) Parks/McClellan Design with linear phase (constant group delay); (d) Parks/McClellan Design with approximately linear phase (approximately constant group delay).

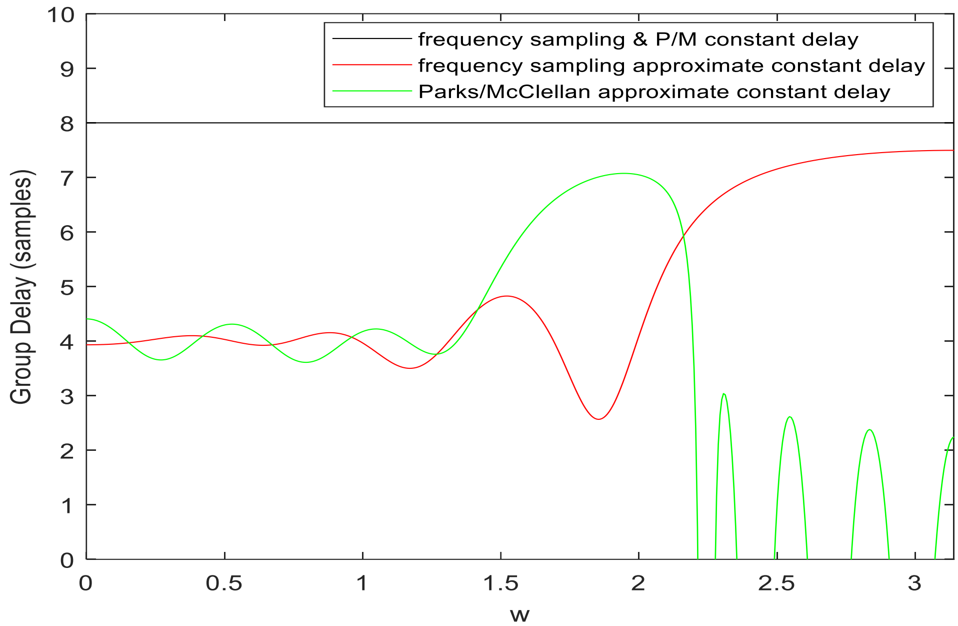

Figure 2: Group delay for four filter designs: Black - Frequency Sampling Design and Parks/McClellan Design with linear phase (constant group delay); Red - Frequency Sampling Design with approximately linear phase (approximately constant group delay); Green - Parks/McClellan Design with approximately linear phase (approximately constant group delay).

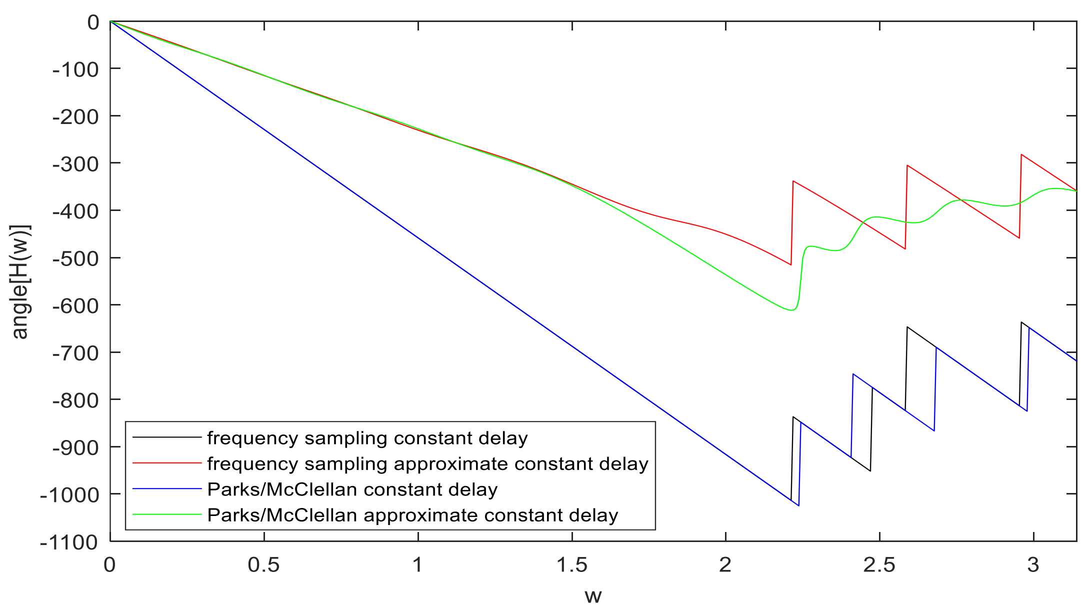

Figure 3: Unwrapped phase for the four filter designs: Black - Frequency Sampling Design with linear phase (constant group delay); Red - Frequency Sampling Design with approximately linear phase (approximately constant group delay); Blue - Parks/McClellan Design with linear phase (constant group delay); Green Parks/McClellan Design with approximately linear phase (approximately constant group delay).

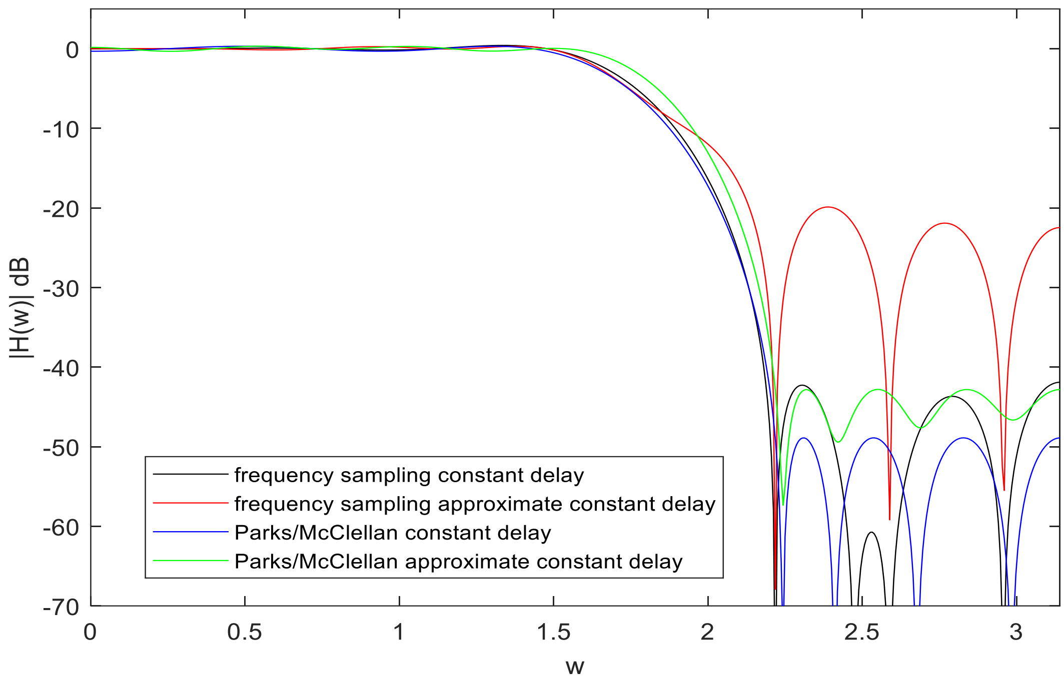

Figure 4: Magnitude Response in dB for the four filter designs: Black - Frequency Sampling Design with linear phase (constant group delay); Red - Frequency Sampling Design with approximately linear phase (approximately constant group delay); Blue - Parks/McClellan Design with linear phase (constant group delay); Green Parks/McClellan Design with approximately linear phase (approximately constant group delay).

REFERENCES

1. J. Proakis and D. Manolakis, Digital Signal Processing: Principles, Algorithms, and Applications, 4th Edition, Pearson, 2007.
2. B. Lathi and R. Green, Essentials of Digital Signal Processing, Cambridge, 2014.
3. A. Oppenheim and R. Schafer, Discrete-Time Signal Processing, 3rd edition, Prentice-Hall, 2010.
4. S. Mitra, Digital Signal Processing: A Computer-Based Approach, 4th ed., McGraw Hill, 2011.
5. The Digital Signal Processing Handbook, Edited by Vijay K. Madisetti and Douglas B. Williams, CRC & IEEE Press, 1998.
6. X. Chen and T.W. Parks, "Design of FIR Filters in the Complex Domain," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-35, No. 2, February 1987.
7. L.J. Karam and J.H. McClellan, "Complex Chebyshev Approximation for FIR Filter Design," IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, Vol. 42, No. 3, March 1995.
8. L.R. Rabiner, B. Gold, and C.A. McGonegal, "An Approach to the Approximation Problem of Nonrecursive Digital Filters," IEEE Trans. Audio and Electroacoustics, vol. AU-18, pp. 83-106, June 1970.

```
% The four filter designs
%Specifications for all 4 designs
M=16;  %Filter Order
L=M+1; %Filter Length
om=linspace(0,2*pi,1001);  %omega frequency running from 0 to 2pi
k=0:M;  %frequency sample index

%(a) Frequency Sampling (constant group delay)
alpha=M/2;  %group delay, M/2 corresponds to linear phase
p=0.4; %transition point value
Q=floor(M/2);
Dw=2*pi/L;  %delta omega, width between frequency samples
Ad=[1 1 1 1 1 p 0 0 0 0 0 0 p 1 1 1 1];  %desired amplitude response
thetadfs=-alpha*Dw*[(0:Q), -(L-(Q+1:M))]; %desired phase response for
Freq Sampling
Hdfs=Ad.*exp(j*thetadfs);  %Desired Frequency Response
hfs=real(ifft(Hdfs));  %Frequency Sampling Impulse Response
Hfs=freqz(hfs,1,om);  %compute the actual frequency response

%(b) Modified Frequency Sampling (approximate constant group delay)
%everything is same as Frequency sampling except different alpha
alpha=M/4;  %group delay, M/4 (approximate group delay)
p=0.4; %transition point value
Q=floor(M/2);
Dw=2*pi/L;  %delta omega, width between frequency samples
Ad=[1 1 1 1 1 p 0 0 0 0 0 0 p 1 1 1 1];  %desired amplitude response
thetadfsm=-alpha*Dw*[(0:Q), -(L-(Q+1:M))]; %desired phase response for
Freq Sampling Modified
Hdfsm=Ad.*exp(j*thetadfsm);  %Desired Frequency Response
hfsm=real(ifft(Hdfsm));  %Modified Frequency Sampling Impulse Response
Hfsm=freqz(hfsm,1,om);  %compute the actual frequency response

%(c) Parks/McClellan Design (constant group delay)
hpm=firpm(M,[0 8/17 12/17 1],[1 1 0 0],[1 10],{128}); %lowpass
Hpm=freqz(hpm,1,om);   %compute the actual frequency response

%(d) Complex Parks/McClellan Desing (approximate constant group delay)
hcpm=cfirpm(M,[0 8/17 12/17 1],{'lowpass',-4},[1 6],{128}); %lowpass
with approximate 4 less samples for group delay
Hcpm=freqz(hcpm,1,om);   %compute the actual frequency response
```

Figure 5: Matlab code for the four filter designs (a) Frequency Sampling Design with linear phase (constant group delay); (b) Frequency Sampling Design with approximately linear phase (approximate constant group delay); (c) Parks/McClellan Design with linear phase (constant group delay); (d) Parks/McClellan Design with approximately linear phase (approximately constant group delay).