Strathprints Institutional Repository

Roscoe, Andrew J. and Blair, Steven M. (2016) Choice and properties of adaptive and tunable digital boxcar (moving average) filters for power systems and other signal processing applications. In: IEEE Applied Measurements for Power Systems (AMPS 2016), 2016-09-28 - 2016-09-30, E.On Energy Research Center, RWTH Aachen University. , http://dx.doi.org/10.1109/AMPS.2016.7602853

This version is available at http://strathprints.strath.ac.uk/57701/

Choice and Properties of Adaptive and Tunable Digital Boxcar (Moving Average) Filters for Power Systems and other Signal Processing Applications

Andrew J. Roscoe, Steven M. Blair
Department of Electronic and Electrical Engineering
University of Strathclyde
Glasgow, UK
Andrew.J.Roscoe@strath.ac.uk

***Abstract*—The humble boxcar (or moving average) filter has many uses, perhaps the most well-known being the Dirichlet kernel inside a short-time discrete Fourier transform. A particularly useful feature of the boxcar filter is the ease of placement of (and tuning of) regular filter zeros, simply by defining (and varying) the time length of the boxcar window. This is of particular use within power system measurements to eliminate harmonics, inter-harmonics and image components from Fourier, Park and Clarke transforms, and other measurements related to power flow, power quality, protection, and converter control. However, implementation of the filter in real-time requires care, to minimise the execution time, provide the best frequency-domain response, know (exactly) the group delay, and avoid cumulative numerical precision errors over long periods. This paper reviews the basic properties of the boxcar filter, and explores different digital implementations, which have subtle differences in performance and computational intensity. It is shown that generally, an algorithm using trapezoidal integration and interpolation has the most desirable characteristics.**

Keywords— Adaptive filters, Array signal processing, Finite impulse response filters, Power system measurements, Fourier transforms, Frequency measurement, Phase estimation, Power system state estimation, Power system parameter estimation

I. The Ideal Boxcar Filter

The ideal boxcar (moving average) filter in real-time is a rolling average of a real-valued input signal over a definite time which extends from a sample at time "now" (*t*=0) back into the past to a time *t* = -*T*, where *T* is the length of the boxcar window, and the window is rectangular. In literature the terms boxcar filter, moving average filter, and rectangular filter/window are all essentially interchangeable, with the same meaning. The boxcar filter is a Finite Impulse Response (FIR) filter, and is shown in Fig. 1. This filter is causal, and has a group delay equal to *T*/2.

The ideal boxcar filter can be expressed in the Laplace domain by combining two opposing unit step (Heaviside) functions into the expression:

$$H_{Ideal}(s) = \frac{1}{\mathrm{T}}\left(\frac{1}{s} - \frac{e^{-Ts}}{s}\right) = \frac{1}{sT}\left(1 - e^{-Ts}\right) \tag{1}$$

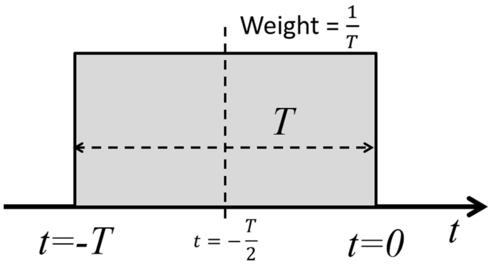

Fig. 1 The ideal boxcar filter.

The frequency domain performance of the ideal boxcar can be understood by evaluating (1) over a range of frequencies and substituting $s = j\omega = j2\pi f$, or by referring to standard tables of Fourier transforms for the rectangular window, time-shifted (the translation theorem) so that the window is not centered on t = 0, but on *t* = - *T*/2. This reveals an amplitude response with a sinc() shape and a linear phase response corresponding to the group delay of *T*/2:

$$H_{Ideal}(f) = \left(\frac{\sin(\pi f T)}{\pi f T}\right) \quad e^{-j\pi f T} \tag{2}$$

The sinc() shape gives the boxcar filter its well-known frequency response (Fig. 2), acting as a low-pass filter, with zeros at every frequency *f* = *n*/*T* where *n* is integer, and a 1st sidelobe with an amplitude of -13.26 dB [1].

By varying the time length of the boxcar, the filter zeros can be moved very easily. For example, if the time length is adjusted to $T = 21$ ms then the filter zeros appear at multiples of 1/0.021 = 47.62 Hz, and the filter is well tuned to measure signals with a fundamental at this frequency. This is one example of an application within a power-systems environment, where the desired value of *T* can be determined in real time, from the power system frequency [2][3][4].

The boxcar filter/window has the lowest equivalent noise bandwidth (ENBW) of any filter, but the high -13.26 dB sidelobe is significantly worse than most other windows/filters. Therefore, boxcar filters are often cascaded or combined with other filters (boxcars or others) to form the most appropriate filter for a particular purpose. In particular they have many applications within satellite and terrestrial communications, and software-defined radio, commonly arranged in a CIC (Cascaded Integrator Comb) architecture [5] [6] [7].

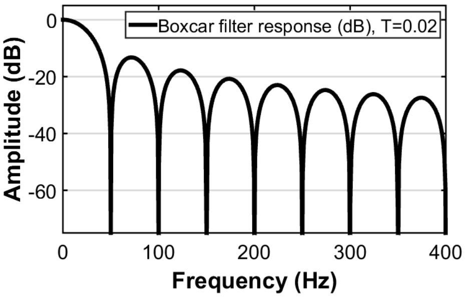

Fig. 2 The ideal boxcar filter response, for $T = 20$ ms window time length.

II. Digital Implementations

When a waveform is sampled digitally at a sample rate f_S, the time between samples is $T_S = 1/f_S$. Aside from the usual issues of aliasing, the main problem in terms of implementing a digital boxcar filter is that the desired value of T, the length of the boxcar, is usually not equal to an integer number of sample times T_S. This is a perceived barrier to implementing digital boxcar filters with general lengths.

Alternative methods focus on using fixed filter lengths with N (integer) samples. When a boxcar (or other) filter with fixed length N samples is used within a Discrete or Fast Fourier Transform (DFT or FFT), a technique called the interpolated DFT/FFT [8] was developed in the 1970s, as a post-processing stage to apply, in an effort to reduce spectral leakage, with associated errors and uncertainties. Many modern power-system measurement algorithms continue to apply this technique, or other fixed-window techniques such as Taylor Fourier Transforms or their derivatives [9] [10] [11]. In fields other than power systems, modern literature still suggests that implementing the general digital boxcar filter is difficult, with various techniques suggested to avoid the problem. For instance, [12] uses a 2-stage technique using variable and fixed multi-rate sampling.

The main point of this paper is to emphasise that dealing with $T \neq N\,T_S$ (N integer), and tuning T in real time, is not difficult, and requires only a few simple calculations. Indeed, a partial solution has existed within MATLAB®/SIMULINK® for a number of years [13], and some of the methods presented below have been used by the author in a number of power-system measurement applications [3] [4], following some much earlier investigations in chapter 3 of [14]. In particular, for the first time, this paper examines the most appropriate curve-fitting and digital filter expressions to use, through formal transfer function investigation, to obtain the best accuracy of zero placement and to maintain good wideband noise rejection.

A. The ideal digital implementation

It is tempting to say that if the desired filter length T happened to result in an exactly integer number of samples, i.e. $T/T_S = N$, with N integer, then the ideal boxcar filter would consist simply of the N most recent samples with equal weighting $1/N$:

$$H_{BackwardEulerIntegerN}(z) = \frac{1}{N}\sum_{i=0}^{N-1} z^{-i} \qquad (3)$$

An equivalent expression can be written, by considering the windowing to be equivalent to a backward Euler digital integration $T_S z/(z-1)$, divided by the total integration time NT_S, times $(1-z^{-N})$, as shown in (4), which can also be derived by summing the geometric progression in (3). This is, at first sight, an accurate digital version of (1) i.e.

$$H_{BackwardEulerIntegerN}(z) = \frac{1}{N}\frac{z}{(z-1)}\left(1-z^{-N}\right) \qquad (4)$$

However, there are two linked properties of this filter which raise issues. Firstly, if the most recent sample is considered at time $t = 0$, then the group delay of the filter is $(N\text{-}1)T_S/2$, which is not equal to $T/2$. The group delay is ½ sample less than expected. Secondly if two such filters are cascaded in the time domain, the combined filter weights are the convolution of the two filters. As a general property, convolution of two filters with N_1 and N_2 weights results in a filter with $(N_1+N_2\text{-}1)$ weights. When convoluting two digital filters with N weights in each, the combined filter will therefore have $(2N\text{-}1)$ weights. Cascading a further identical filter will lead to a filter with $(3N\text{-}2)$ weights. The filter gets shorter than expected by one sample, every time a new filter is cascaded, and it is linked to the fact that the filter group delay is ½ sample shorted than expected.

For this reason, the most accurate representation of (1) in a digital format, with N integer, is to take the filter represented by (3) & (4) and instead to use trapezoidal integration $T_S(z+1)/2(z-1)$. Effectively the filter weights resulting from (3) are convoluted by a tiny additional filter with 2 equal weights of [½ ½], where the convolution by [½ ½] is a handy method of converting backward Euler integration into trapezoidal integration. The resulting expression is:

$$H_{Ideal}(z) = \frac{1}{2N}\frac{(z+1)}{(z-1)}\left(1-z^{-N}\right) \qquad (5)$$

This filter has $(N+1)$ weights, with the most recent at time t = 0 and the oldest with time t = $-NT_S$, and possesses a group delay of $NT_S/2$. The group delay matches (1). If two filters are cascaded (convoluted), the resulting filter will have $2(N+1)\text{-}1 = 2N+1$ weights. Cascading a further identical filter will lead to a filter with $(3N+1)$ weights, etc. Therefore, successive cascading no longer results in a filter which seems to shrink each time.

Although the whole argument above considered T/T_S to result in N being conveniently integer, there is nothing mathematical which prevents (5) being evaluated with N replaced by a variable m which is not integer, to deduce an "ideal" digital boxcar filter response.

$$m = \frac{T}{T_S} \qquad (6)$$

Evaluating (5) directly using m allows an ideal digital boxcar filter response H_{Ideal} to be bode-plotted in the frequency domain. However, when m is not integer, it is not practical to actually use this approach to design and/or execute a real filter. The required filter weights are not obvious, but can be estimated by either performing a Fourier transform on the bode plot, or by placing the filter zeros at the expected notch frequencies $f = \pm n/T$, plus an extra zero at Nyquist $f=f_S/2$, and multiplying out the resulting polynomial in z. For either method, this will be computationally expensive if m is large and will make real-time filter reconfiguration impractical. Additionally, the evaluation of the filter output will require complete re-computation of the entire sum of products of FIR weights and samples. Again, when f_S and/or m is large, this prevents practical implementation. The following methods, by contrast, are designed to operate with a minimum of mathematical computations, to maximize the speed of both filter evaluation and reconfiguration. In fact, reconfiguration can be done at the full $f_S=1/T_S$ rate.

B. *General procedure for filter design*

In a general case, for a desired boxcar time length T, the following expressions are the starting point for configuring a practical filter:

$$N = ceil(m) \text{ , i.e. } N = \lceil m \rceil \quad \text{(the ceiling function)} \tag{7}$$

$$x = N - m \quad \text{with } 0 \le x < 1 \tag{8}$$

The actual number of samples lengths required is m, while N is the integer number of samples over which an initial integration – an approximate (overlong) boxcar window – is performed. x represents the fractional part-sample worth of window which must be subtracted from the initial overlong window.

Several implementation variants/versions will be described. The overall filter response H_{vv} for each variant vv is constructed in stages according to the equation:

$$H_{vv} = \frac{1}{T}\left[H_{I\_vv} H_{D\_vv} - H_{C\_vv} \right] H_N \tag{9}$$

First the signal is passed through an integrator with response H_{I_vv}, which contains only a few computations and states. The result of the integration from a time NT_S in the past is then subtracted from the present integrator output using a function which makes the integral definite: $H_{D\_vv} = \left(1 - z^{-N}\right)$ for all practical digital implementations. This requires a memory buffer of length N samples – in fact two are usually required so that each integrator can be periodically reset to remove the possibility of long-term numerical precision errors in the floating-point integrator [4]. The result at this point is the overlong integration – overlong by a period of xT_S seconds. A small correction is then required to "back out" the overlong section. This requires a short expression with a response H_{C_vv} using 1, 2 or 3 samples which were taken around the period of the window beginning, a time NT_S into the past. The expression used, and number of samples used in it, must be based on and exactly match the integration method in H_{I_vv}. This requires another memory buffer, so that 3 buffers are (usually) required in total. H_N is an optional extra step which simply implements a 2-sample filter and places a zero at the Nyquist frequency $f=f_S/2$:

$$H_N = \frac{\left(1 + z^{-1}\right)}{2} \tag{10}$$

Note that the number of computational steps in this whole procedure is independent of the value of N. The only implication of using large numbers of samples is the expanding memory requirement for the buffers.

C. *Implementation using backward Euler integration*

The backward Euler method simply has integration:

$$H_{I\_BE} = \frac{zT_S}{(z-1)} \tag{11}$$

Also, the extra integrated area which must be removed due to x can simply be written as :

$$H_{C\_BE} = T_S x z^{-(N-1)} \tag{12}$$

The method is shown graphically in Fig. 3, and also corresponds to (4) if N is integer and $x = 0$.

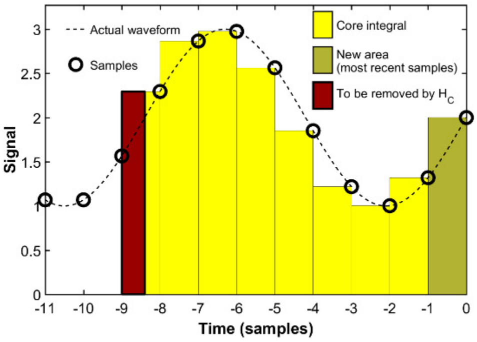

Fig. 3 Example of the backward Euler method H_{BE} with $T = 8.4\ T_S$

D. *Implementation using Trapezoidal integration*

Essentially the incoming signal value is curve-fit with a 1st-order polynomial such that its value y can be estimated:

$$y = k_0 + k_1 x' \tag{13}$$

$$\text{where : } x' = \frac{t - (-T_S)}{T_S} = 1 + \frac{t}{T_S} \tag{14}$$

In (14), x' is defined so that the newest sample is considered at t=0, where x'=1, while the previous sample is effectively at t=-T_S, x'=0 and the idea is to find the expression for the new trapezoidal area to add using the accumulation function $z/(z$-1):

$$H_{I\_T} = T_S \int_0^1 (k_0 + k_1 x')\, dx' \frac{z}{(z-1)} = T_S\left(k_0 + \frac{k_1}{2}\right)\frac{z}{(z-1)} \tag{15}$$

For the 1st order (linear) interpolation:

$$k_0 = z^{-1} \quad \text{and} \quad k_1 = \left(1 - z^{-1}\right) \tag{16}$$

Combining (15) and (16) reveals the not-unexpected form of the trapezoidal integration expression.

$$H_{I\_T} = \frac{\left(1 + z^{-1}\right)}{2}\left(\frac{zT_S}{(z-1)}\right) \tag{17}$$

More importantly, the same curve fitting process can be applied to the samples in the past, to reveal the expression for H_{C_T}. To do this, x' and the k_is are redefined using the much older samples:

$$x' = \frac{t - (-NT_S)}{T_S} = N + \frac{t}{T_S} \tag{18}$$

$$k_0 = z^{-N} \quad \text{and} \quad k_1 = \left(z^{-(N-1)} - z^{-N}\right) \tag{19}$$

Now H_{C_T}, the correction for the overlong initial integration, can be deduced as (using x from (8)):

$$OverlongArea = H_{C\_T} = T_S \int_0^x \left(k_0 + k_1 x'\right) dx' = T_S \left(k_0 + \frac{k_1}{2}\right) \qquad (20)$$

Combining (19) and (20) reveals:

$$H_{C\_T} = T_S \left[z^{-(N-1)} \left(\frac{x^2}{2} \right) + z^{-N} \left(\frac{2x - x^2}{2} \right) \right] \qquad (21)$$

The entire process is shown graphically in Fig. 4, which can be compared to Fig. 3

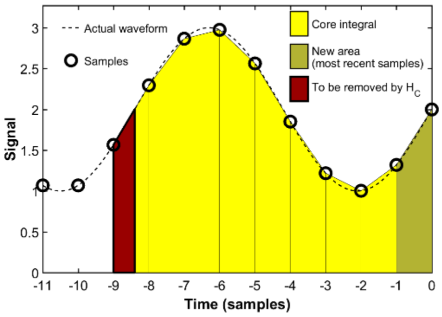

Fig. 4 Example of the Trapezoidal method H_T with $T = 8.4\ T_S$

The Simulink block "Mean (Variable Frequency)" [13], in existence since pre-2004, implements a similar filter. However, [13] begins the averaging at t = $-NT_S$ and ends at t = $-x$, so the group delay changes with x (and T). Also [13] contains only one floating-point integrator which runs continually, and is therefore liable to precision errors due to long-term DC accumulations or individual input samples with very large values.

E. Implementation using 2nd order polynomial integration

The procedure for defining a process using 2nd order curve fitting is the same as for the trapezoidal (1st order) method, using the same expressions for x, and x' but with longer expressions for:

$$H_{I\_2} = T_S \int_0^1 \left(k_0 + k_1 x' + k_2 x'^2\right) dx' \frac{z}{(z-1)} \qquad (22)$$

$$H_{I\_2} = T_S \left(k_0 + \frac{k_1}{2} + \frac{k_2}{3}\right) \frac{z}{(z-1)} \qquad (23)$$

$$k_0 = z^{-1}\ ,\ k_1 = \frac{\left(1 - z^{-2}\right)}{2},\ k_2 = \frac{\left(1 + z^{-2}\right)}{2} - z^{-1} \qquad (24)$$

Combining (23)and (24) reveals H_{I_2} which is shown in TABLE I. Then, again by looking into the past, H_{C_2} can also be determined similarly, using (25) and (26), with the result shown in TABLE I. Simulink source code for the 2nd-order algorithm was given in chapter 3 of [14].

$$k_0 = z^{-N}, k_1 = \frac{\left(z^{-(N-1)} - z^{-(N+1)}\right)}{2}, k_2 = \frac{\left(z^{-(N-1)} + z^{-(N+1)}\right)}{2} - z^{-N} \qquad (25)$$

$$H_{C\_2} = T_S \int_0^x \left(k_0 + k_1 x' + k_2 x'^2\right) dx' = T_S \left(k_0 + \frac{k_1}{2} + \frac{k_2}{3}\right) \qquad (26)$$

The expression for H_{I_2} in TABLE I is interesting in that it is heavily asymmetric. Since the boxcar filter is ideally symmetrical, it raises the possibility that a better filter could be built by manipulating the filter so that it looks more symmetrical. This can be done, by increasing the group delay of the filter from the ideal value of $T/2$, by $T_S/2$. This is done by modifying the integrals (22) and (26), specifically by subtracting 0.5 from all of the upper and lower bounds of the definite integrals, and re-evaluating to give alternative versions of H_{I_2S} and H_{C_2S}, both shown in TABLE I, leading to an alternative boxcar response H_{2S}. Graphical representations of the processes to achieve H_2 and H_{2S} are shown in Fig. 5 & Fig. 6.

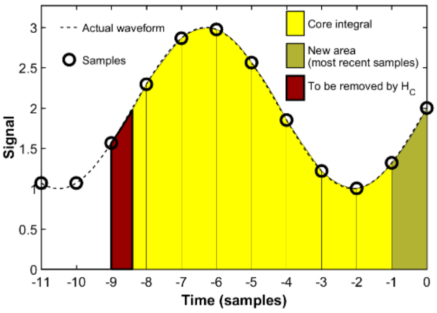

Fig. 5 Example of the 2nd-order method H_2 with $T = 8.4\ T_S$

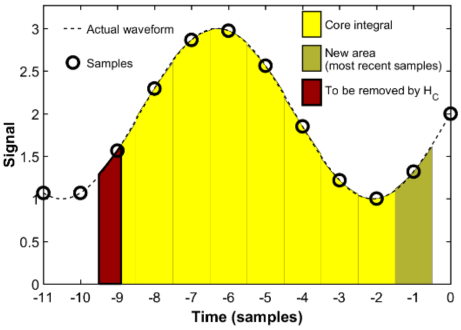

Fig. 6 Example of the 2nd order method H_{2S} (symmetrical) with $T = 8.4\ T_S$

III. Comparisons of Performance & Choice of Algorithm

Before examining the overall transfer functions H_{vv} it is useful to examine just the transfer functions H_{I_vv}(z-1)/zT$_S$, i.e. H_{I_vv}/H_{I_BE}, the adjustments made to the integral/accumulation expression compared to the basic backward Euler expression, due to the curve fitting regime. These are shown in Fig. 7. The trapezoidal integration tends to provide the best rejection of noise at higher frequencies towards Nyquist. Meanwhile, the heavily unequal weights [1,22,1]/24 of the "more symmetric" version of the 2nd order curve fit H_{I_2S} provide a performance not dissimilar to that of the simple backward Euler method. The "normal" 2nd order curve fit also provides significantly less attenuation at Nyquist than the trapezoidal method.

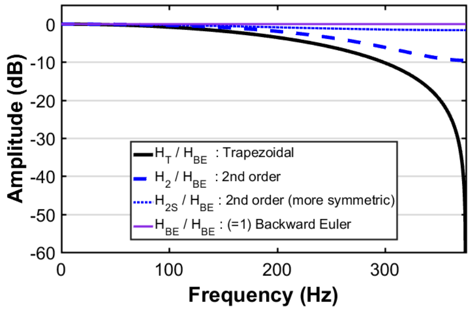

Fig. 7 Curve fitting expression H_{I_vv}/H_{I_BE} response, f_S=750 Hz.

TABLE I. SUMMARY OF CANDIDATE BOXCAR IMPLEMENTATIONS, TRANSFER FUNCTIONS AND GROUP DELAYS

Boxcar version	Transfer function H_{vv}	H_{I_vv} (Integration)	H_{D_vv}	Part-sample correction H_{C_vv}	Extra zero at Nyquist H_N	Group delay
Ideal	$H_{Continuous}$	$\frac{1}{s}$	$\left(1-e^{-Ts}\right)$		1	$\frac{T}{2}$
Ideal digital	H_{Ideal}	$\frac{\left(1+z^{-1}\right)}{2}\left(\frac{zT_S}{(z-1)}\right)$	$\left(1-z^{-m}\right)$		1	$mT_S=\frac{T}{2}$
Backward Euler	H_{BE}	$\frac{zT_S}{(z-1)}$	$\left(1-z^{-N}\right)$	$T_S x z^{-(N-1)}$	1	$(m-0.5)T_S$ $=\frac{(T-T_S)}{2}$
Trapezoidal	H_T	$\frac{\left(1+z^{-1}\right)}{2}\left(\frac{zT_S}{(z-1)}\right)$	$\left(1-z^{-N}\right)$	$T_S\left[z^{-(N-1)}\left(\frac{x^2}{2}\right)+z^{-N}\left(\frac{2x-x^2}{2}\right)\right]$	1	$mT_S=\frac{T}{2}$
Trapezoidal with additional zero at Nyquist	H_{TN}	$\frac{\left(1+z^{-1}\right)}{2}\left(\frac{zT_S}{(z-1)}\right)$	$\left(1-z^{-N}\right)$	$T_S\left[z^{-(N-1)}\left(\frac{x^2}{2}\right)+z^{-N}\left(\frac{2x-x^2}{2}\right)\right]$	$\frac{\left(1+z^{-1}\right)}{2}$	$(m+0.5)T_S$ $=\frac{(T+T_S)}{2}$
2nd order polynomial	H_2	$\frac{\left(5+8z^{-1}-1z^{-2}\right)}{12}\left(\frac{zT_S}{(z-1)}\right)$	$\left(1-z^{-N}\right)$	$T_S\begin{bmatrix} z^{-(N-1)}\left(\frac{2x^3+3x^2}{12}\right) \\ +z^{-N}\left(\frac{12x-4x^3}{12}\right) \\ +z^{-(N+1)}\left(\frac{2x^3-3x^2}{12}\right)\end{bmatrix}$	1	$mT_S=\frac{T}{2}$
2nd order polynomial (more symmetric)	H_{2S}	$\frac{\left(1+22z^{-1}+1z^{-2}\right)}{24}\left(\frac{zT_S}{(z-1)}\right)$	$\left(1-z^{-N}\right)$	$T_S\begin{bmatrix} z^{-(N-1)}\left(\frac{4x^3-3x}{24}\right) \\ +z^{-N}\left(\frac{12x^2-8x^3+18x}{24}\right) \\ +z^{-(N+1)}\left(\frac{4x^3-12x^2+9x}{24}\right)\end{bmatrix}$	1	$(m+0.5)T_S$ $=\frac{(T+T_S)}{2}$

The next 4 bode plots show the relative performance of the whole boxcar responses. Fig. 8 and Fig. 9 show performance for a very slow sample rate of 750 Sa/s (nominally 15 Sa/cycle at 50 Hz). Fig. 10 and Fig. 11 show performance for a much higher sample rate of 10 kSa/s (nominally 200 Sa/cycle at 50 Hz). All these 4 figures are produced by picking an awkward boxcar length so that x=0.75 and the performance of the H_{C_vv} components are important to achieve good rejection at the notches at $f = n/T$. Other values of non-zero x such as 0.5 are also of interest, giving similar results. When x=0, all methods produce ideal notch rejection at $f = n/T$, while high-frequency noise rejection remains consistent with Fig. 7, Fig. 8 and Fig. 11.

At the low sample rates, Fig. 8 shows that the backward Euler and 2nd order methods provide slightly worse performance than the trapezoidal method at the higher frequencies towards Nyquist. Fig. 9 shows that the backward Euler method provides very poor rejection at the second notch (crucial for image rejection inside a DFT) when $x \neq 0$. Fig. 9 also shows that at low sample rates, the 2nd order (less symmetric) method provides the best notch rejection, being marginally better than the trapezoidal method. However, the 2nd order method with the more symmetric integration function is worse than all other methods except the backward Euler. This is somewhat of a surprise.

At higher sample rates, Fig. 10 shows that the notch performance of all methods except the backward Euler becomes essentially the same, while Fig. 11 shows that, as for the lower sample rates, the trapezoidal method is always the best practical filter at rejecting noise and high-frequency components near Nyquist. If additional rejection at Nyquist is required, it can be done by adding the simple H_N filter to make H_{TN}, which adds just a half-sample group delay. However, if multiple boxcar filters are cascaded, even without H_N, the combined attenuation at

Nyquist may already be several hundred dB, giving ample attenuation for a practical application without adding H_N.

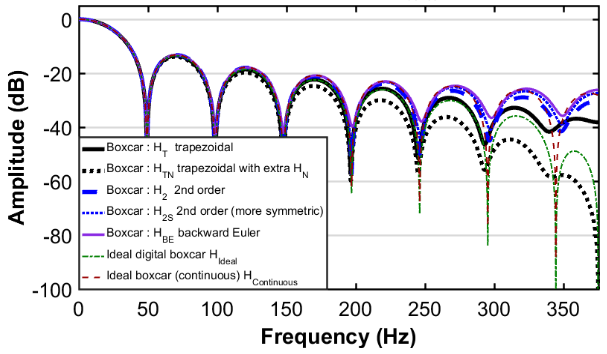

Fig. 8 Response, T=1/49.1803 s, f_S=750 Hz, m=15.25, x=0.75

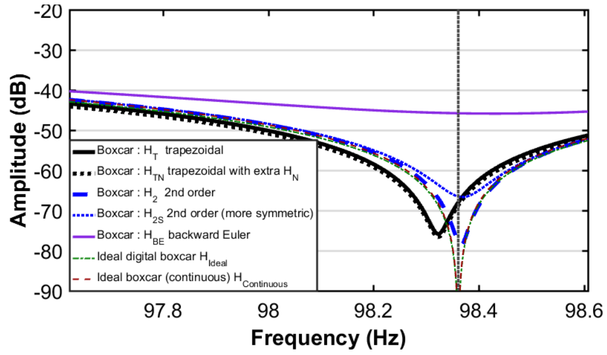

Fig. 9 Response at 2nd notch, T=1/49.1803 s, f_S=750 Hz, m=15.25, x=0.75

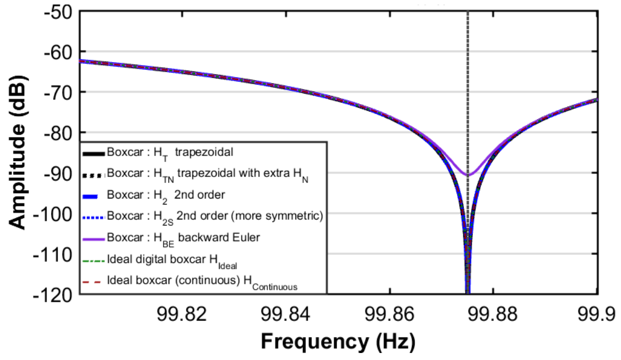

Fig. 10 Response at 2nd notch, T=1/49.9376 s, f_S=10 kHz, m=200.25, x=0.75

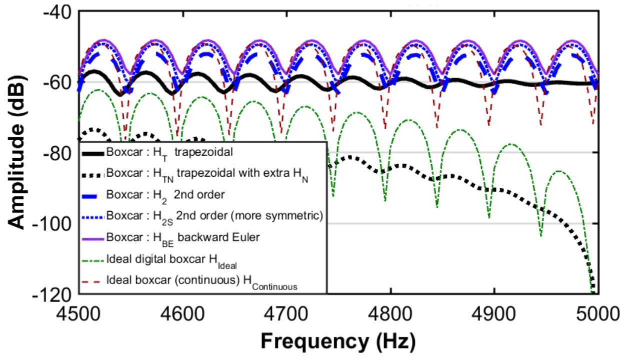

Fig. 11 Response near-Nyquist, T=1/49.9376 s, f_S=10 kHz, m=200.25, x=0.75

IV. Conclusions and Further Work

The conclusion from this work is that the digital boxcar filter using trapezoidal integration is normally the most appropriate, since it offers high accuracy (at anything other than the lowest sample rates), the correct group delay, a low level of processing, and a reasonable attenuation at the Nyquist frequency. If, however, sample rate is low compared to the input signal frequency (~15 samples/cycle or less), the 2nd order method may be more appropriate, giving deeper and more accurate notches (zeros), at the expense of some additional computational intensity and slightly poorer rejection at the Nyquist frequency. For all methods, if required, an additional zero can always be placed at the Nyquist frequency by cascading an additional 2-sample filter $\left(1+z^{-1}\right)/2$, with minimal execution time or group delay penalties.

References

[1] F. J. Harris, "Use of Windows for Harmonic-Analysis with Discrete Fourier-Transform," *Proceedings of the IEEE,* vol. 66, pp. 51-83, 1978.

[2] D. Jovcic, "Phase locked loop system for FACTS," *IEEE Transactions on Power Systems,* vol. 18, pp. 1116-1124, Aug 2003.

[3] A. J. Roscoe, I. F. Abdulhadi, and G. M. Burt, "P and M Class Phasor Measurement Unit Algorithms using Adaptive Cascaded Filters," *IEEE Transactions on Power Delivery,* vol. 28, pp. 1447-1459, 2013.

[4] A. J. Roscoe, G. M. Burt, and J. R. McDonald, "Frequency and fundamental signal measurement algorithms for distributed control and protection applications," *IET Generation, Transmission & Distribution* vol. 3, pp. 485-495, May 2009.

[5] E. B. Hogenauer, "An Economical Class of Digital-Filters for Decimation and Interpolation," *IEEE Transactions on Acoustics Speech and Signal Processing,* vol. 29, pp. 155-162, 1981.

[6] G. Molnar and M. Vucic, "Closed-Form Design of CIC Compensators Based on Maximally Flat Error Criterion," *IEEE Transactions on Circuits and Systems Ii-Express Briefs,* vol. 58, pp. 926-930, Dec 2011.

[7] L. Lo Presti, "Efficient modified-sine filters for sigma-delta A/D converters," *IEEE Transactions on Circuits and Systems Ii-Express Briefs,* vol. 47, pp. 1204-1213, Nov 2000.

[8] V. K. Jain, W. L. Collins, and D. C. Davis, "High-Accuracy Analog Measurements Via Interpolated FFT," *IEEE Transactions on Instrumentation and Measurement,* vol. 28, pp. 113-122, 1979.

[9] J. A. D. L. O. Serna, "Dynamic phasor estimates for power system oscillations," *IEEE Transactions on Instrumentation and Measurement,* vol. 56, pp. 1648-1657, Oct 2007.

[10] P. Romano and M. Paolone, "Enhanced Interpolated-DFT for Synchrophasor Estimation in FPGAs: Theory, Implementation, and Validation of a PMU Prototype," *IEEE Transactions on Instrumentation and Measurement,* 2014.

[11] G. Barchi, D. Fontanelli, D. Macii, and D. Petri, "On the Accuracy of Phasor Angle Measurements in Power Networks," *IEEE Transactions on Instrumentation and Measurement,* vol. 64, pp. 1129-1139, May 2015.

[12] W. Sun, J. X. Shen, K. Wang, and M. J. Jin, "Motor Control Application of Fixed-Sampling-Interval and Fixed-Depth Moving Average Filters," *IEEE Transactions on Industry Applications,* vol. 52, pp. 1831-1841, Mar-Apr 2016.

[13] MATLAB Simulink, "Mean (Variable Frequency)," 2004 Available: http://uk.mathworks.com, accessed May 2016.

[14] A. J. Roscoe, "Measurement, control and protection of microgrids at low frame rates supporting security of supply," PhD thesis, Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, 2009, Available: http://strathprints.strath.ac.uk/39631/, accessed June 2013.