ELECTRONOTES
1016 Hanshaw Road
Ithaca, NY 14850

APPLICATION NOTE NO. 407

March 30, 2014

INVERTING A MOVING AVERAGE OPERATION

INTRODUCTION

When we filter something, in the everyday sense of something like a coffee filter, we are separating components. In the case of coffee, there is a component we usually keep (the liquid coffee) and the one we toss (the grounds) at least as far as the compost heap. That is, we primarily want the filtered product. This is what **filtering** is about. Or we might turn down a treble control when playing an old vinyl recording so as to remove the noise of accumulated surface dirt. Again - we filter out something and discard it.

We can also think of filtering in terms of **analysis**. I believe they use certain ceramic filters in biological laboratories to see if certain tiny critters make it through or are blocked. Perhaps we don't particularly care to have either of the tiny critters, but we are trying to find out what a certain substance contains by using a specific filter. A bank of electrical filters can analyze a sound in this general manner.

So there can be two cases in which filtering is used: for actual removal of something undesired, and for analysis.

Electrical and mathematical filters are designed to have general capabilities, and in addition, are often addressed to a specific application. There are way too many cases to even briefly list here. One question we do address here is whether it is possible to "un-filter"? In the mechanical filter instances above, likely we can, perhaps by just pulling the partitioning filter out and allowing remixture. In the electrical case, we might be able to remix – as for example the reconstruction (usually called synthesis) of prefect reconstruction filters (PRF), which can follow the analysis bank of the PRF.

It should be obvious that in order to "un-filter" (sometimes called equalization and sometimes deconvolution), you must have somehow kept the material originally filtered out. In the case of the PRF, the full set of analysis filters contains the various parts. In the case of ordinary filtering, we may be out of luck if we expect a complete recovery. Clearly if we have merely attenuated a particular frequency or band of frequencies, we can amplify them back. But, if we have in fact completely nulled a frequency (at least in terms of a realistic steady-state), it's gone. If we have a signal that has frequencies 100 Hz and 300 Hz, and a filter with notches at 100 Hz and 200 Hz, the output will contain only 300 Hz and you won't know about 100 Hz or 200 Hz. While thought of here in terms of constant frequencies, note that more transient signals will only present a more difficult situation.

TALKING ABOUT AVERAGES AND MOVING AVERAGES

We can begin with an example of a Moving Average (MA) filter, also known as a Running Mean or traditionally as a "Boxcar", and probably by other terms. For example, a length-7 MA would have an impulse response of:

$$h_{MA7} = [1/7\ 1/7\ 1/7\ 1/7\ 1/7\ 1/7\ 1/7] = (1/7)[1\ 1\ 1\ 1\ 1\ 1\ 1] \quad (1)$$

Now, a MA is not an average. To get a length-7 average, we add up 7 things and divide by 7, and that's it. To get a MA, we would have this average as one of several or many successive outputs. That is, a MA is not a number, but a sequence (or signal). Particularly as we might have a very long MA we note the computational situation that a new MA is available from the old value by removing the oldest contribution and adding the newest incoming contribution. (This is sometimes called a "Head/Tail" computation of a MA – quite obvious). Also, we usefully just think of the computation as a convolution of the impulse response with the contributions. What is the relationship of the MA sequence to the contributions?

First, what is the relationship of the average to the contributions. Clearly the average does not retain the original information. If you and 6 friends are contemplating having lunch at a burger joint and are assessing in-pocket funds, you might find that the average is $55.22. This tells us that all together, the seven of you have $386.54, which should be enough for your modest lunch plans. What it does not tell you, for example, is, if you all forget to take you wallets except for Sally, does Sally have enough money to cover the total bill? In fact, she could have any amount from $0 to $386.54. In taking the one-time average, information is lost.

So the next thing we might want to investigate is whether the MA sequence contains more information, and clearly it does. We can watch the computation advance as each friend is queried. If the first average is $2.87 we conclude that this first person has $2.87 x 7 or $20.09. If Sally is next and the average jumps to $12.91, this number, and the previous value of the MA, allows us to conclude that Sally has $70.28. Indeed, each person has 7 times the increment in the MA. Obvious enough.

Being free to have imaginary friends and corresponding funds, let's suppose the 7 friends have funds at:

$$x = \quad 20.09 \quad 70.28 \quad 0.77 \quad 35.14 \quad 128.10 \quad 49.49 \quad 82.67 \quad (2)$$

for which the sum is 386.54 and the average is 55.22. As we described in words above, we can compute moving averages by convolving x with a sequence [1 1 1 1 1 1 1]. This convolution of two length-7 sequences is of course a length-13 sequence:

$$y = 20.09\ \ 90.37\ \ 91.14\ 126.28\ 254.38\ \ 303.87\ \ 386.54\ \ 366.45\ \ 296.17\ \ 295.40\ \ 260.26\ 132.16\ \ 82.67 \quad (3)$$

Here we have listed the total amounts. To get the average, we just divide by 7. We perhaps suppose that here we have obtained more information as we have 13 numbers where we previously had only 7.

But it is also clear that the sequence y, for the first seven values, is the accumulated total. If we further convolve y with a sequence [1 -1] we can extract the differences. Thus:

$$\begin{array}{lrrrrrrr} z = & 20.09 & 70.28 & 0.77 & 35.14 & 128.10 & 49.49 & 82.67 \\ & -20.09 & -70.28 & -0.77 & -35.14 & -128.10 & -49.49 & -82.67 \end{array} \quad (4)$$

We now have 14 numbers in total, and we immediately notice that the first 7 (top line of equation (4) are a recovered form of x. This was expected. Perhaps less expected is that the second line of equation (4) is just the negative of x. If you think of what is going on, this just means that the component amounts exit from the summer in the same order they entered.

Thus we see that the MA (here actually the moving sum – which gives the same illustrations) does retain the original information, unlike the fixed average. This is somewhat intuitively understood. What may be less clear is exactly what is going on, and how much care goes into looking in the right places to find the output numbers.

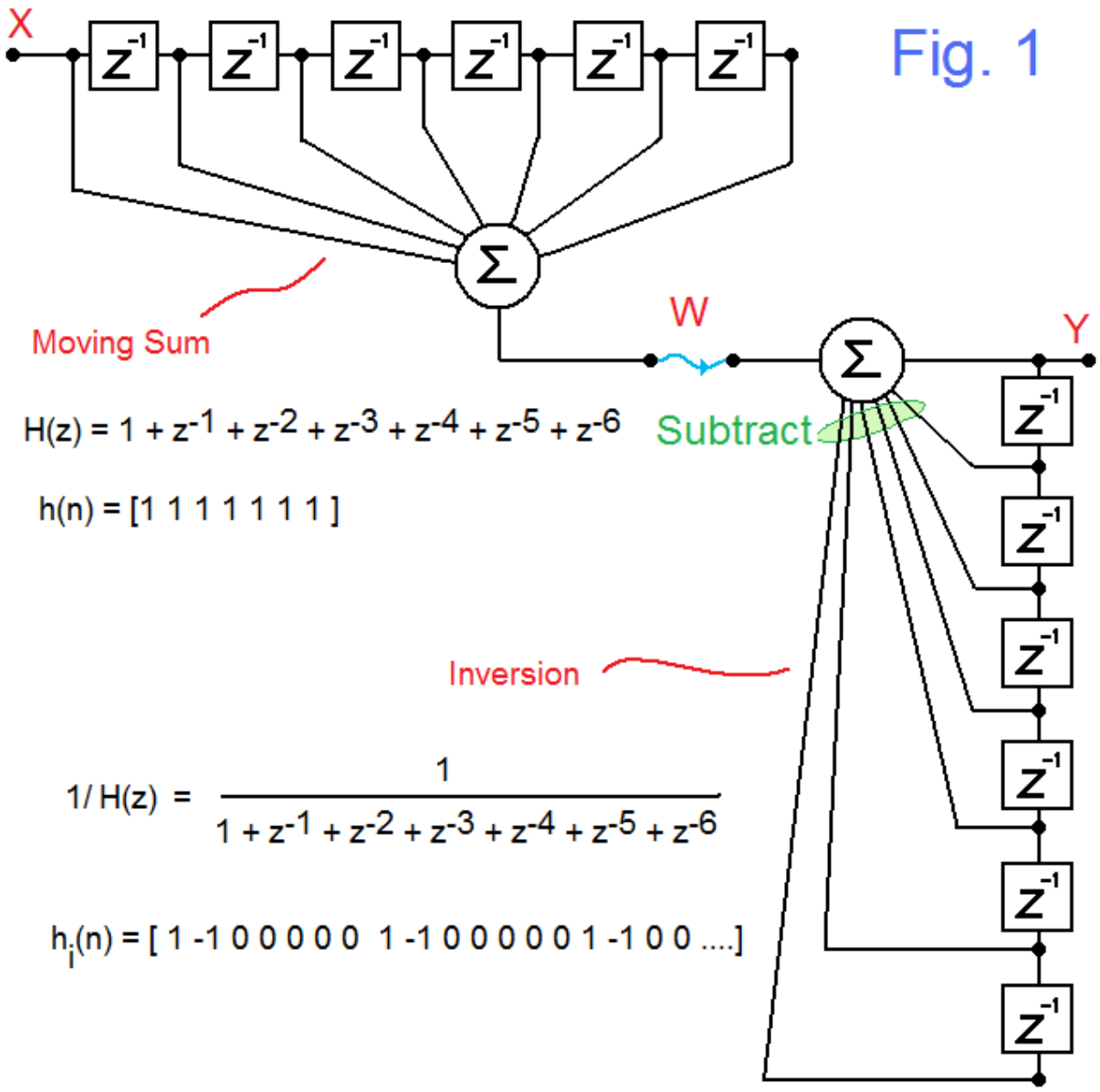
THE INVERSE FILTER

In a previous app note, AN-366, "Unfiltering – Equalization", May 2006 we looked at some aspects of this problem [1]:

http://electronotes.netfirms.com/AN366.pdf

There we considered the general problem and some specific least-squares based approaches to finding inverse filters that were at least approximations.

The approach was simple enough. We filter with a filter H(z) and convert an input signal X(z) to an output signal Y(z) = H(z)X(z). Clearly the way to un-filter this is to multiply by a filter 1/H(z). As shown in the reference at the link above, or in many texts, this is not always, or even often, possible. The reason is simple: lots of filters H(z) that we use have unit circle zeros (and/or zeros outside the unit circle). This is because lots of (most) useful filters have stopbands – they are trying to filter something out. Yet the frequency response of a filter, if it is of the common LTI (Linear Time-Invariant) type can be zero only at isolated points (nulls or notches – unit-circle zeros), and we rely on the fact that the response in the vicinity of these nulls is relatively small as well. This is how stopbands are achieved. Thus we immediately understand that some signal components will be blocked by a notch, and thus lost. As for the inverse filter 1/H(z), it will have unit-circle poles at the same unit circle positions, and will require infinite response there. Filter with zeros outside the unit circle will have inverses which additionally blow up. Even if we believe we have zeros coming first, blocking any nasty poles, any noise in the data can cause a response to blow up even where numerically it seems to sneak by.



INVERTING THE MOVING SUM

At this point, we will look explicitly at the “Moving Sum” which is just the MA without the confusing division (we will freely jump back and forth). Recently we have looked at MA and related smoothing [2-8]. We will continue to use length 7, and Fig. 1 shows H(z) and the inversion 1/H(z) for this case. Note the subtractions in the feedback (green oval). H(z) is simply a length-7 delay line with unity taps. Its impulse response is seven ones. It is easy to write down 1/H(z) as in Fig. 1. It is not obvious (to me at least) what the impulse response is, but it is easy to calculate: it is an infinite length sequence that repeats the length-seven sequence [1 -1 0 0 0 0 0].

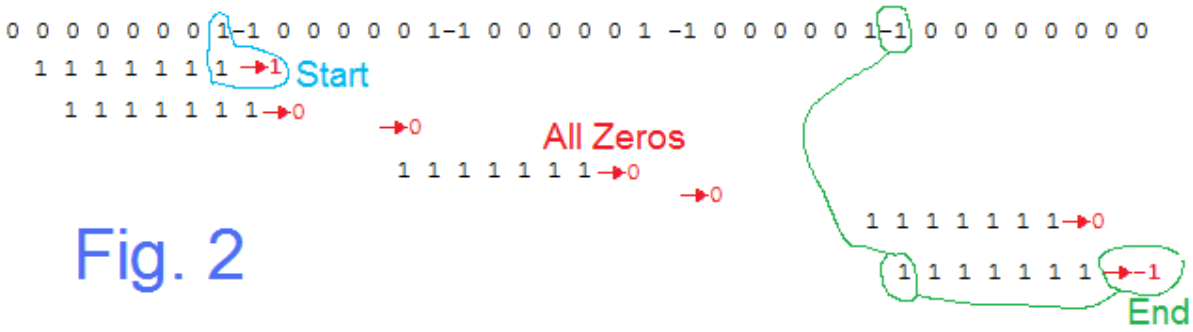


Fig. 2

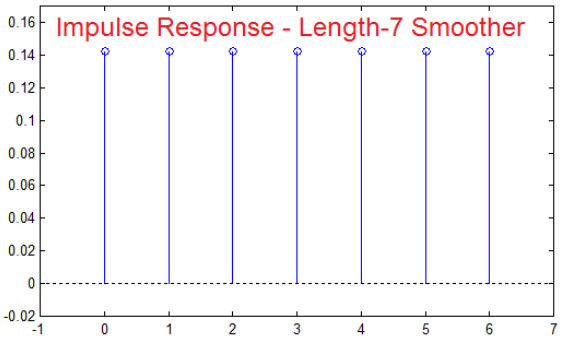
Now we have H(z) and 1/H(z) and their impulse responses should cancel. If we analyze Fig. 1, we see that H(z) loads 1/H(z) with an impulse, which comes immediately out. After the shift, another "1" is loaded to 1/H(z), but the original impulse at Y has been shifted down, subtracted, so that the current Y is zero. It is less clear that a string of zeros is all that happens – in theory. Fig. 2 shows the convolution situation. Note the initial "1" at the light blue Start. Then there is a region of red All Zeros, and this would seem to be ongoing indefinitely. As long as we don't stop. Note that if the response stops, we get a departing green "-1" at the End. It is extremely entertaining to watch numbers move through the networks, cancelling, but it had to happen!

Note that the impulse response of 1/H(z) is IIR (Infinite Impulse Response). Because of the unit circle zeros, this IIR response with its corresponding unit-circle poles does not decay. In many cases, AN IIR impulse response does decay and may get negligibly small and end "quietly" (see below).

CONTINUING THE EXAMPLE

Next we continue this example in some detail. Fig. 3a shows the impulse response, frequency response, and zero-plot for the length-7 Moving Average [2]. Note that the frequency response has notches at frequencies 1/7, 2/7, and 3/7. Our "test signal" here will be a length-10 rectangle, not to be confused with the smoother's length-7 impulse response. The unit circle zeros shown correspond to the notches in the frequency response, and are perfectly acceptable for a filter as far as stability goes. When we put H(z) in the denominator to form 1/H(z), we put poles at the exact same position as the unit circle zeros (Fig. 3b) and this has two important effects on the filter's responses. First, the impulse response not only is no longer finite duration, but does not decay at all. Here we plot only 111 values in Fig. 3b, (upper left – see also Fig. 2), although the actual program that we use here kept 300 samples (which will be important for Fig. 3c). Secondly, because the filter 1/H(z) has unit circle poles, the response blows up to infinity at frequencies 1/7, 2/7, and 3/7 while our plot (Fig. 3b, upper right) is of course clipped.

Now it is curious that the filter 1/H(z) being unstable (conditionally stable?) allows us to see anything useful. When we bring noise into the picture, we will see obvious problems. The fact that we see an interesting result (Fig. 3c) is a consequence of the same curious special loading of samples that we described above as extremely entertaining.



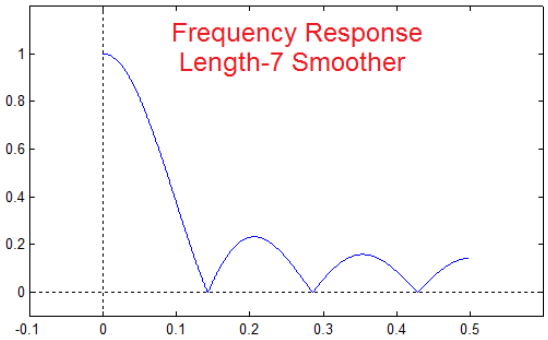
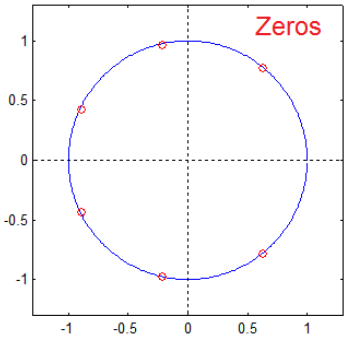
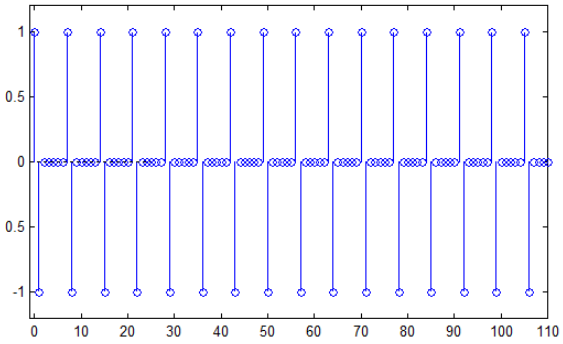


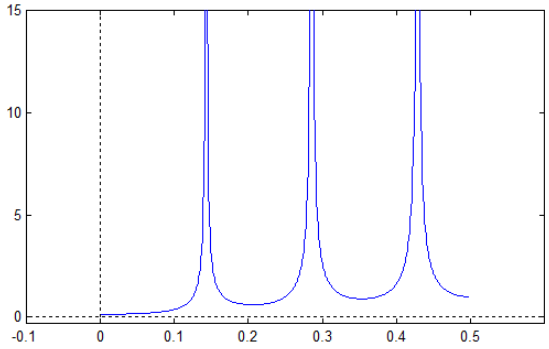
Fig. 3a



H(z)



Impulse Response - Inverter
(actually continues forever on right)



Frequency Response - Inverter
(peaks infinitely high)

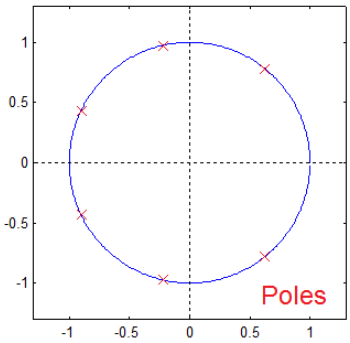


Fig. 3b

1 / H(z)

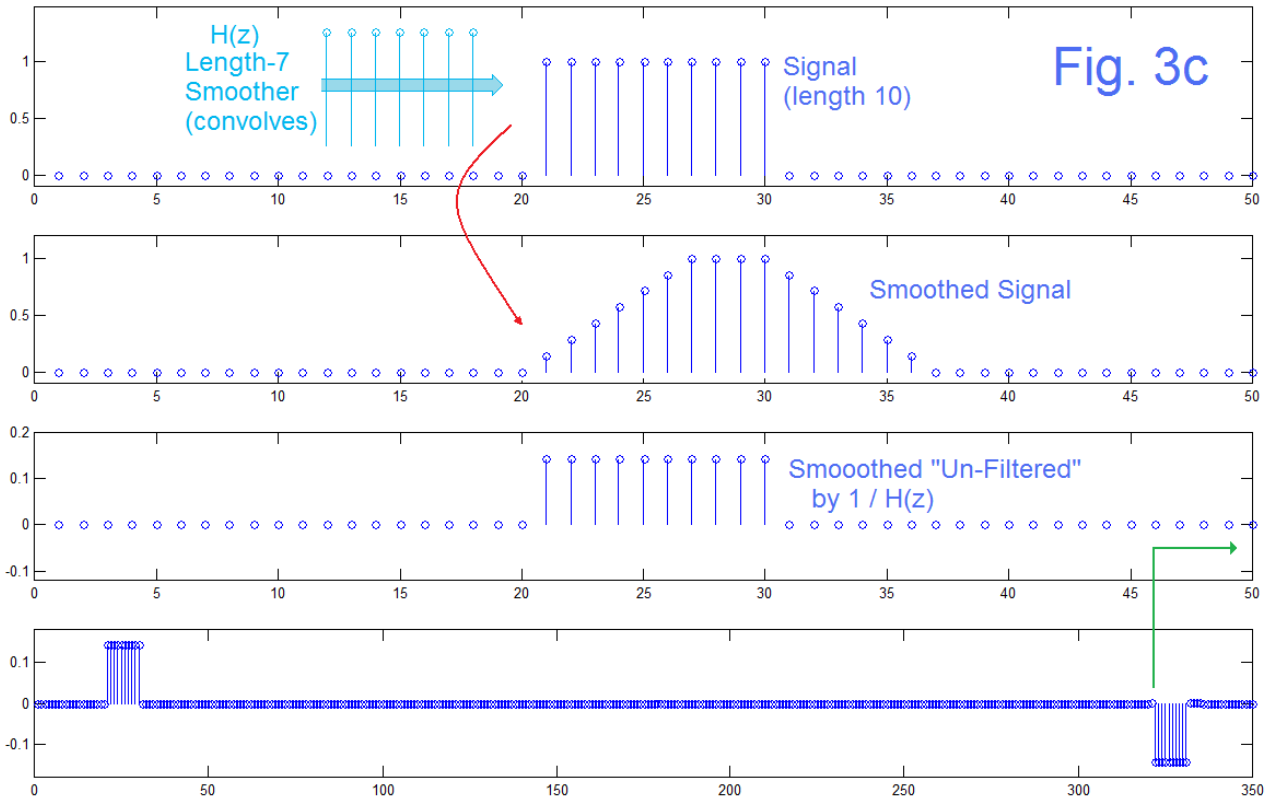


Fig. 3c shows a time-domain view of what is happening. In the top line we see the test signal (length-10 – darker blue) and the impulse response of H(z) (length-7 – lighter blue) as they are about to be convolved, resulting in the trapezoid of the second line. Accordingly the step edges are made gradual (smoothed). The question is whether we can remove this smoothing, and the unlikely answer – is Yes. In Fig. 2, we saw that H(z) convolved with 1/H(z) gave back an impulse. So we should not be surprised (because of superposition) that we get the length-10 rectangle back (third line of Fig. 3c). Now, from Fig. 2 we also saw that if we DID truncate the IIR response, we got a late inverted “echo” of the impulse response. Here (bottom line of Fig. 3c) we see the rectangular signal is likewise echoed because, as we said, we truncated the IIR response at length 300, where it was still going strong (no decay at all). The green line from the bottom line to the one above reminds us where this bump really is.

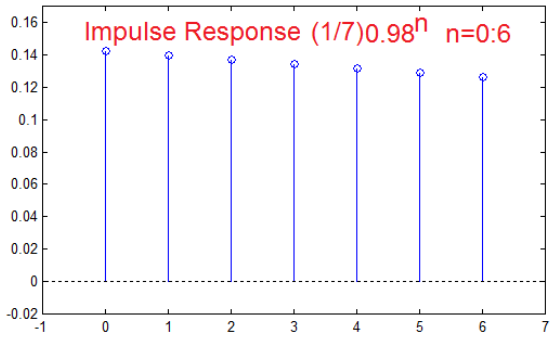
All this is amusing, and perhaps misleading. We have three more things to look at: (1) what happens if the poles move inside (or outside) the circle; (2) what happens for a frequency of exactly 1/7 and is this information lost (and what does that mean); and (3) what happens with noise added prior to reconstruction?

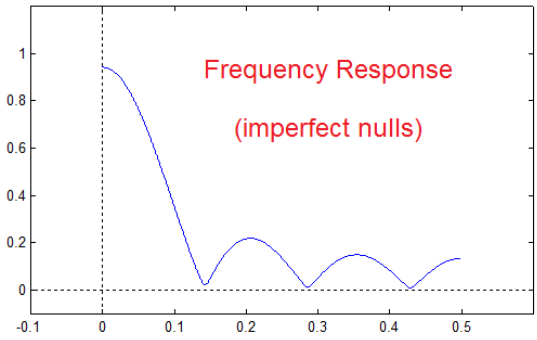
DECAYING POLES

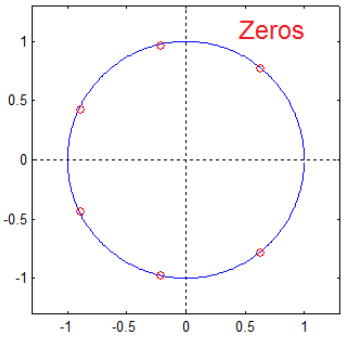
Here we really don't need to do anything except rerun Fig. 3a, Fig. 3b, and Fig. 3c with the zeros slightly inside the unit circle. Actually, we started with a rectangular smoothing function and calculated the zeros as being on the unit circle. At this point, how do we choose the smoothing taps? In full disclosure, I guessed, guessed correctly, and lost interest in the derivation. The verification was just a matter of starting by setting zeros at the same angles as the original, moving the radius of the zeros from 1.00 to 0.98, and using Matlab's ***poly*** on these. This in fact gives h(n) as $(0.98)^n$ for n=0 to 6. In fact, doing the same thing by hand convinces us that we understand the necessary derivation steps if we were so inclined. Thus, instead of h(n) = [1 1 1 1 1 1 1] we would have:

h(n) = [1 0.98 0.9604 0.941192 0.92236816 0.9039207968 0.885842380864] (5)

This is the only change. Note that because we started with the z-plane, an overall gain factor is arbitrary. (Any transfer function can be multiplied by any constant factor and still has the same poles/zeros.) From Fig. 4a, upper left, we see the impulse response is slightly decaying instead of being exactly rectangular. In consequence, the frequency response nulls (Fig. 4a, upper right) are not exactly to zero. This is important because it means nothing is completely lost – not even at isolated frequencies. On the scale plotted, it is difficult to see, but the zeros are slightly inside the unit circle (Fig. 4a, lower left).



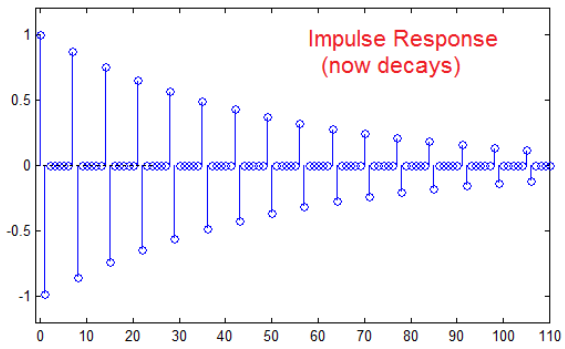




zeros slightly inside circle at 0.98

Fig. 4a

H(z)



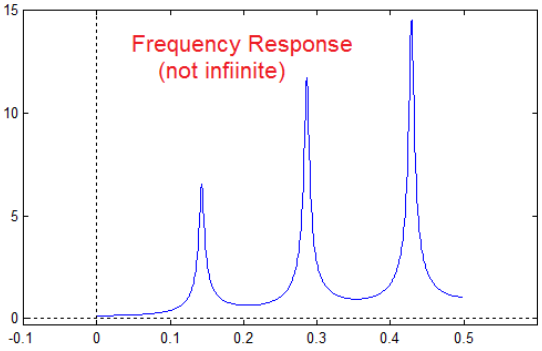
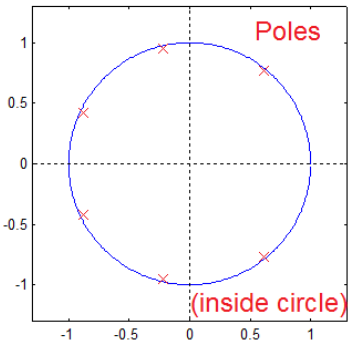


Fig. 4b



Impulse response is $(0.98)^n$ multiplied by [1 -1 0 0 0 0 0] repeated for all $n \geq 0$.

1 / H(z)

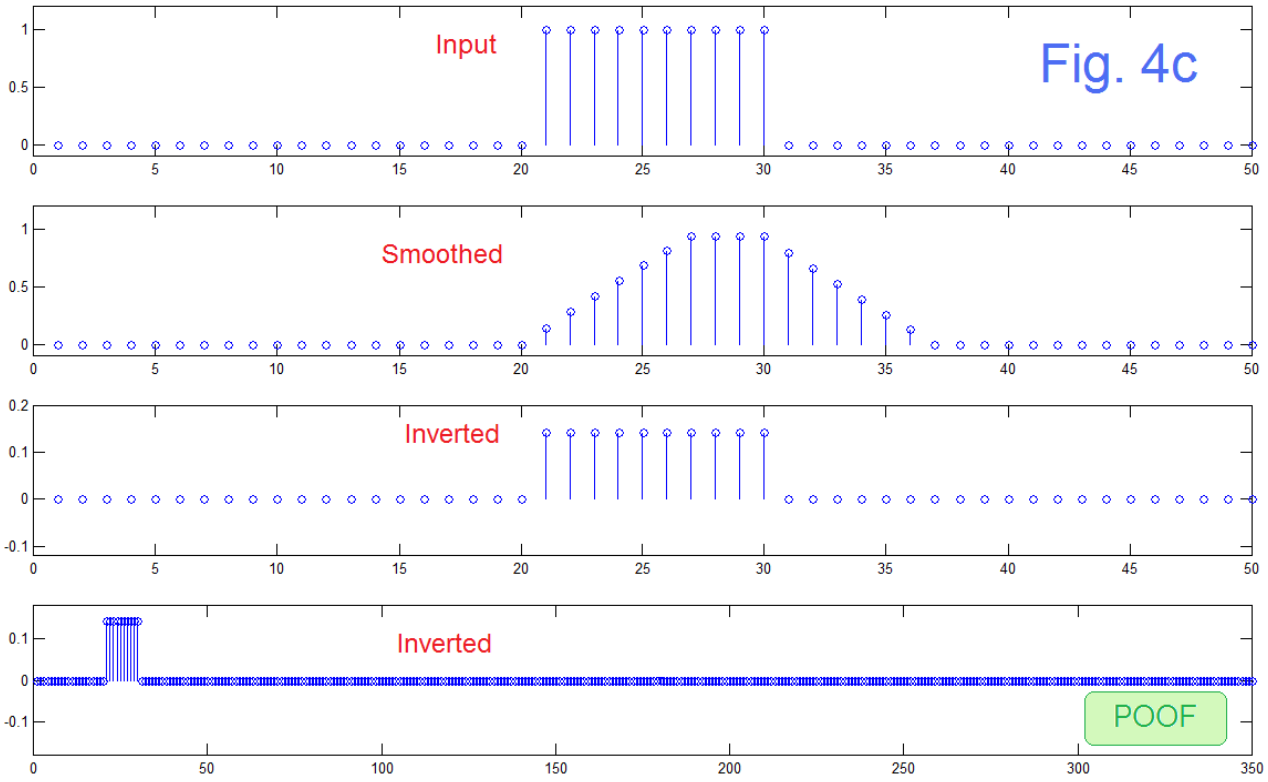


Fig. 4c

In Fig. 4b, we have the corresponding 1/H(z). The impulse response now decays, as is obvious in the upper left. There we plot 111 values of the impulse response, but in the actual calculation we keep 300 values. The frequency response (upper right of Fig. 4b) now shows peaks at 1/7, 2/7, and 3/7 that do not blow up to infinity, and the poles are inside the circle exactly as the zeros were. We might expect the reconstruction to be apparently better, and it is seen in Fig. 4c. In fact, it looks pretty much the same as Fig. 3c, with the notable exception that the "late echo" at the right side of the bottom line is gone (POOF). This is because instead of being a truncated version of a non-decaying impulse response, it went away exponentially and gradually. Yet the cancellations beyond the original recovery are complete because we have the same values in the impulse responses of H(z) and 1/H(z). Something a bit magical still is going on it seems. Well, it's just math.

LOSS OF INFORMATION?

Recently I took a part in an online discussion of whether or not filtering means that information is lost. A good part of the disagreement was semantics, one of the most difficult things to settle with blog comments. To me, clearly, when you filter something information is lost (missing from the filter output), regardless of whether or not you can fully recover the input signal, or if you have stored away a copy of the input in reserve!

In the present discussion, we are talking about "un-filtering" so we have full recovery in mind. This requires, at the very least, a full knowledge of the filtering involved and an absence of random noise, as we have done so far. Our immediate task now is to show what happens when we clearly remove information from the signal.

The MA filter, first of all, has a specific purpose of reporting out a moving mean, which is clearly low frequency and thus something for which we would expect to see a low-pass filtering. It is also true that the MA, in addition to passing DC, has specific notches, as we have seen at 1/7, 2/7, and 3/7 in our example. Suppose we want to use the MA to block signal components of frequency 1/7. A sine wave of frequency 1/7 (period 7) is thus exactly what we would hope to block, and this looks easy to show. In Fig. 5a (X) we form an input signal as 28 zeros, 6 full cycles of frequency 1/7, and 28 more zeros. This is filtered with a length-7 moving sum (impulse response [1 1 1 1 1 1 1]) to give W, which does seem to put a big hole in the middle of the sinewave cycles. We might have expected everything to disappear, but there are the end transients to consider, and these remain.

Looking ahead we see that the bottom line of Fig. 5a shows the recovery of X as Y (along with the echo as we truncated 1/H(z) at length 154). So while we can do this, we rely on our knowledge of the specific filter to get the right inversion. For the moment, looking at W, we see that information has been lost. Anything we expected to use or infer from looking at the W is confounded at best. One good way of looking at this situation is to consider, as we often do, the FFT of the output, choosing as we would generally do, the region of the most energy. From Fig. 5a, we would use samples 29-70 of X and samples 29-77 of W (W having become longer, like any convolution). Conveniently we take FFT of length N=42 for X, and of length N=49 for W, and the results (magnitude FFTs) are shown in Fig. 5b.

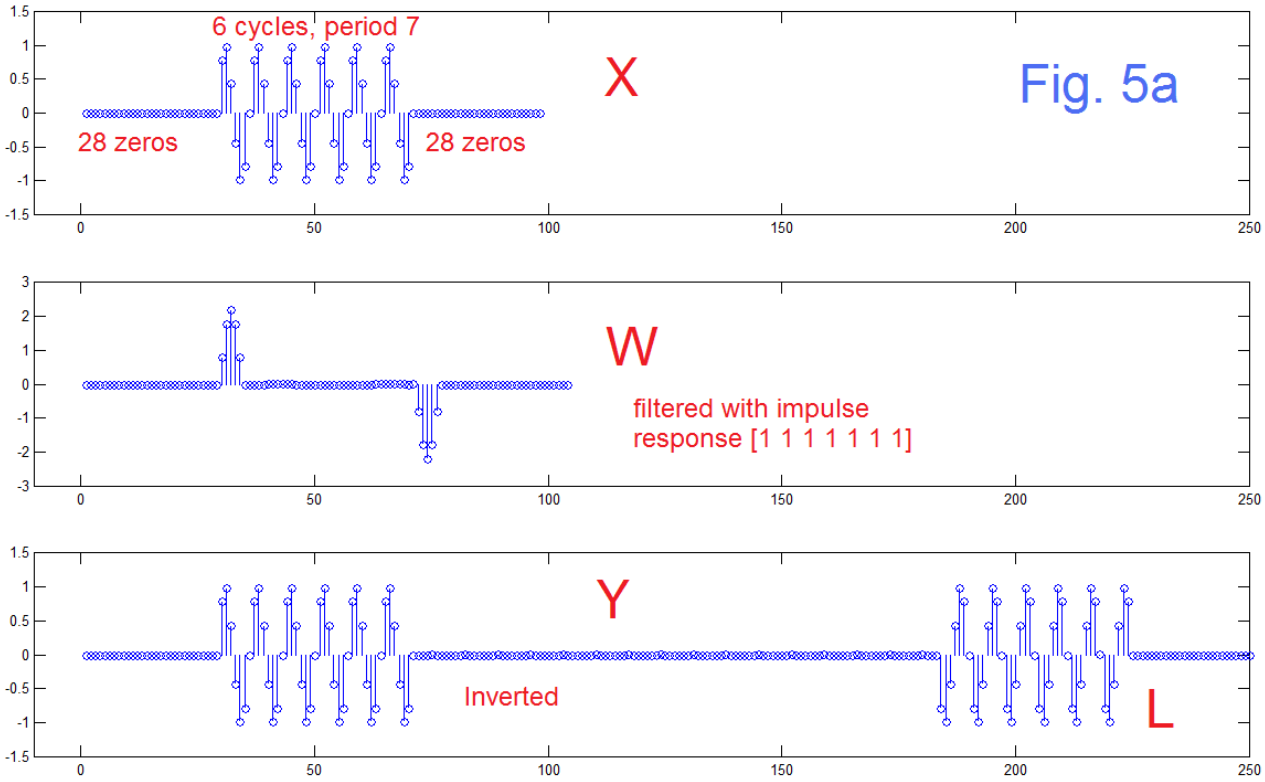

6 cycles, period 7
X
28 zeros
28 zeros
W
filtered with impulse
response [1 1 1 1 1 1 1]
Y
Inverted
L

Fig. 5a

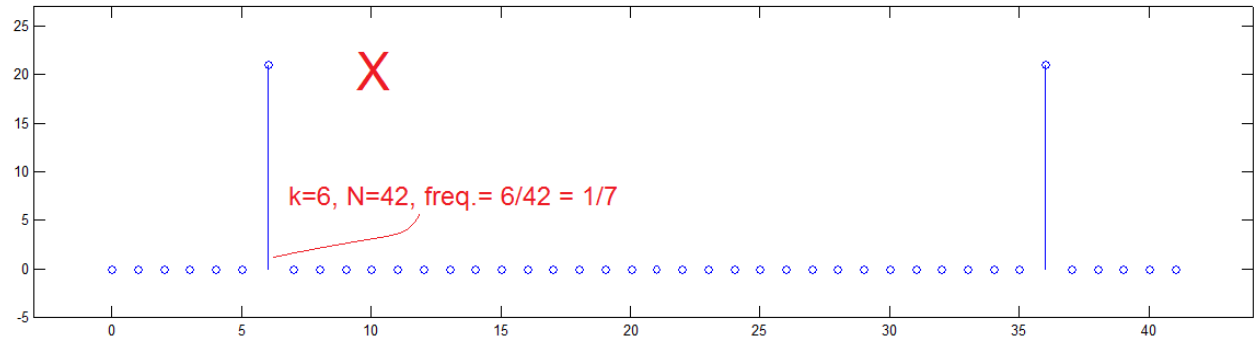
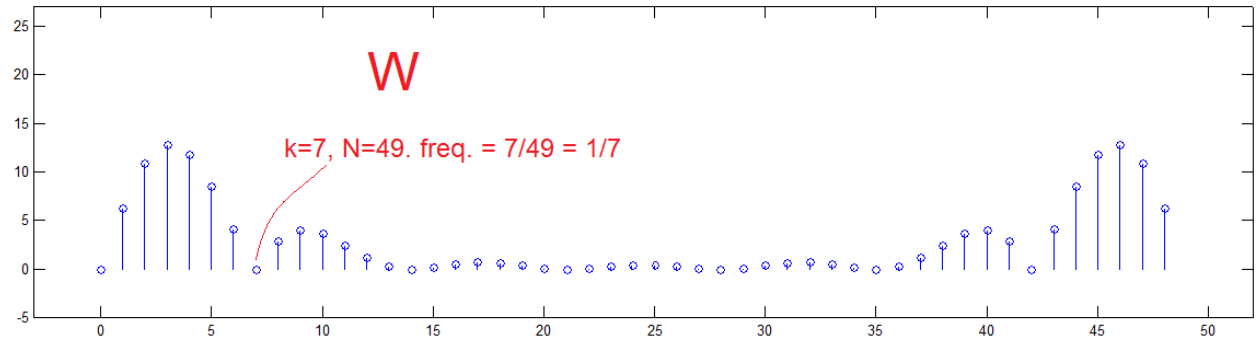

X
k=6, N=42, freq.= 6/42 = 1/7

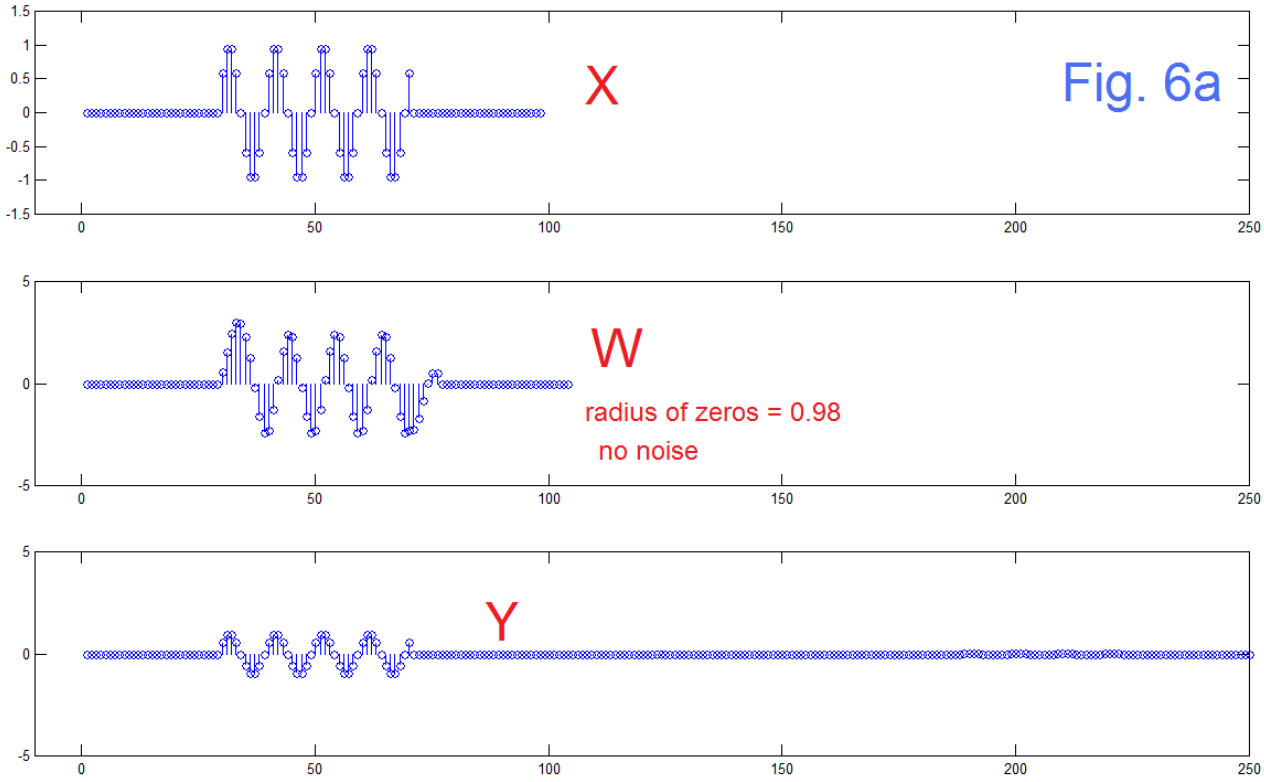
Fig. 5b

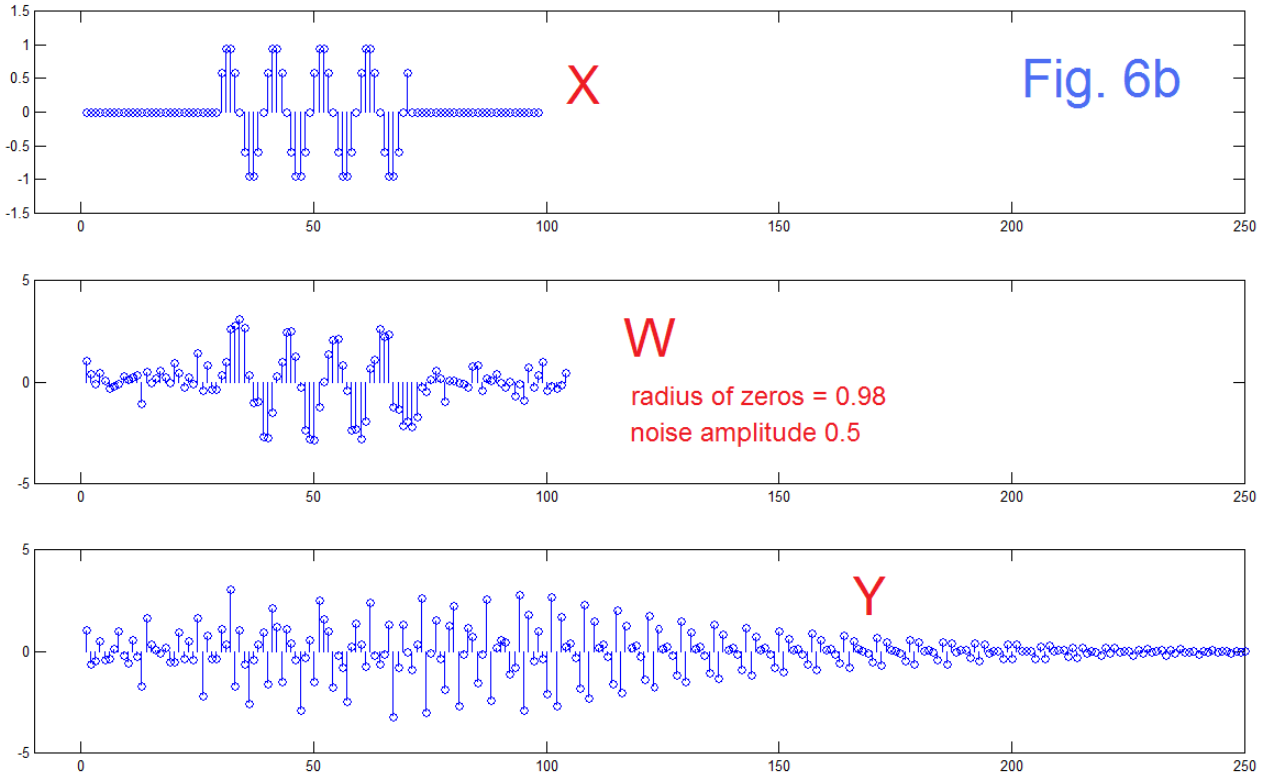

W
k=7, N=49. freq. = 7/49 = 1/7

Here we find an agreeable result. In Fig. 5b top, for X we see that all the energy is concentrated in the FFT bin for k=6 (and k=43-6 = 37) indicating that all the signal is periodic at frequency 6/42 = 1/7. Yes, this involved a special setup for the clear interpretation that results here. Similarly, W had absolutely no energy at the frequency for the k=7 FFT bin, or 7/49 = 1/7. So, at this point, we have a reasonable interpretation of the result that the presence of a periodicity of frequency 1/7 is lost. The notion that Y can be recovered from W, requiring us to specify the exact filter, is equivalent to storing the original signal. But this is an assumption. In general, we just will be observing that **we have only W**, and make of it what you can, which is less than getting X back.

AND – NOISE DOES WHAT?

Here when we speak of added noise we are asking about noise that is added to the filtered output before we try the inversion. This is typically something like what we try to achieve by "equalizing" a communications channel (reducing its amplitude and phase distortions) while at the same time recognizing that the channel has added random nose as well. Any noise that is in the input signal will just be handled exactly as the signal itself. Here we will start with a signal that has just over four full cycles or a frequency 1/10, and will use a length-7 MA filter that has zeros at a radius 0.98, so that the inverse will be stable (although it will "ring" considerably) . Fig. 4a shows that this MA filter had no null at a frequency 1/10 so that means that the signal is not blocked outright.





In Fig. 6a we show the noiseless case and this is a pretty good example of what we might chose as a typical application. Although very hard to see, the small “echo” up around 200 in Fig. 6a for Y is pretty much the only indication that anything is going on.

In contrast, the noisy case of Fig. 6b is pretty much a disaster. Here the noise added is about 20% of the signal level at that point. In W of Fig. 6b, we see the noisy signal awaiting inversion. In Y of Fig. 6b there is very little to suggest that the input signal can be recovered. About all we can say is that the signal ended somewhere around sample 100 when the IIR reconstruction filter just rattles as it decays.

APPLICATION OF FINDINGS

Since the MA filter is so common we might have reason for attempting to invert data from an MA smoothed output. Most people who use it tell you the length, and whether or not anything else special was used. Often however there are unwarranted assumptions about the frequency response other than at DC. There is no reason not to investigate the entire range however, and this is very possibly the most important thing about using a MA. There are many alternatives.

There is also the possibility of an approximate inversion of a MA using an approximate reconstruction that is stable. That is, use an IIR that has its poles slightly inside the unit circle. So we use half our findings above. We saw that we could invert using a stable IIR if we started out with a MA-like smoother that had weights that decayed. The pairing was

still H(z) and 1/H(z). Given that so many (all?) use ordinary equal weigh MA, the alternative to do smoothing a new way may have passed. As such, particularly as the smoothed data may have become itself noisy (such as by roundoff), attempting an inversion with unit-circle poles begins as something worse than a bad idea! But - nothing prevents us from trying an inversion with poles inside the unit circle. It might work well enough.

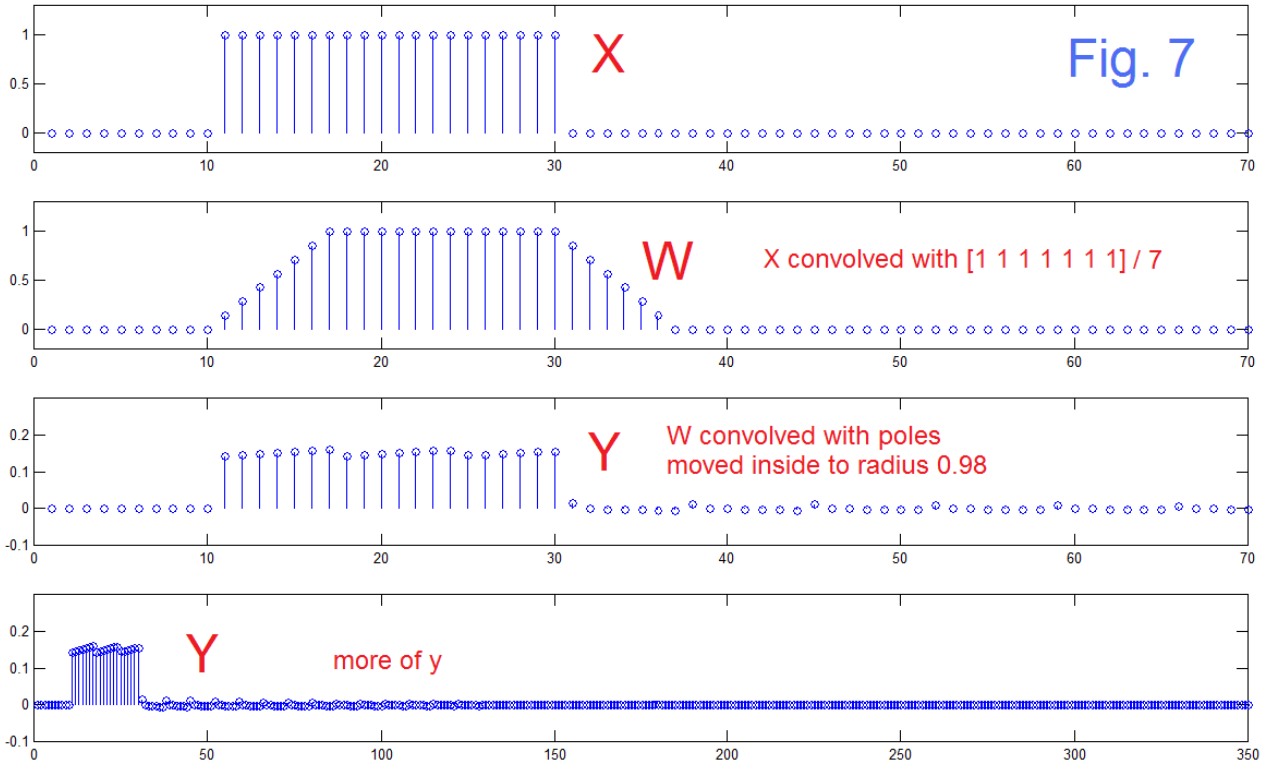


Fig. 7 shows such an example. The input is a length-20 rectangle X and it is smoothed with a length-7 MA, resulting in W. Then we do the inversion not with the usual 1/H(z) but with the IIR with poles moved to a radius of 0.98 rather than the original radius of 1. To be more specific, it is too late (we suppose) to fix the smoother, but we will try to fix the inverter. The hope is that it is good enough, and perhaps it will work in cases where the poles on the unit circle would have been just a disaster. Perhaps. Y is not all that bad.

Fig. 8 shows a second test example. Here we start with a Signal consisting of four sinusoidal cycles, and add noise. The Noise has a “marker” artificially put in – six ones from samples 60-65 just to help identify. The Signal + Noise constitutes X, which is then subjected to a length-10 MA rectangular smoother, producing W. Normally we think of W as the presentation of the filter output – enhancing the signal and rejecting the noise.

Here we are interested in seeing if we can recover the variability – the noise. We know the smoother was rectangular, but we want to avoid unit circle poles, so we construct an impulse response for the Proposed Inverter as suggested above by repeating the sequence [1 -1 0 0 0 0 0 0 0 0] ten times and then multiplying by $(0.98)^n$ where n runs from 0 to 99. This we convolve with W to produce Y. While we did not expect nor achieve a perfect reconstruction, we have little doubt that we are recovering much of the variability.

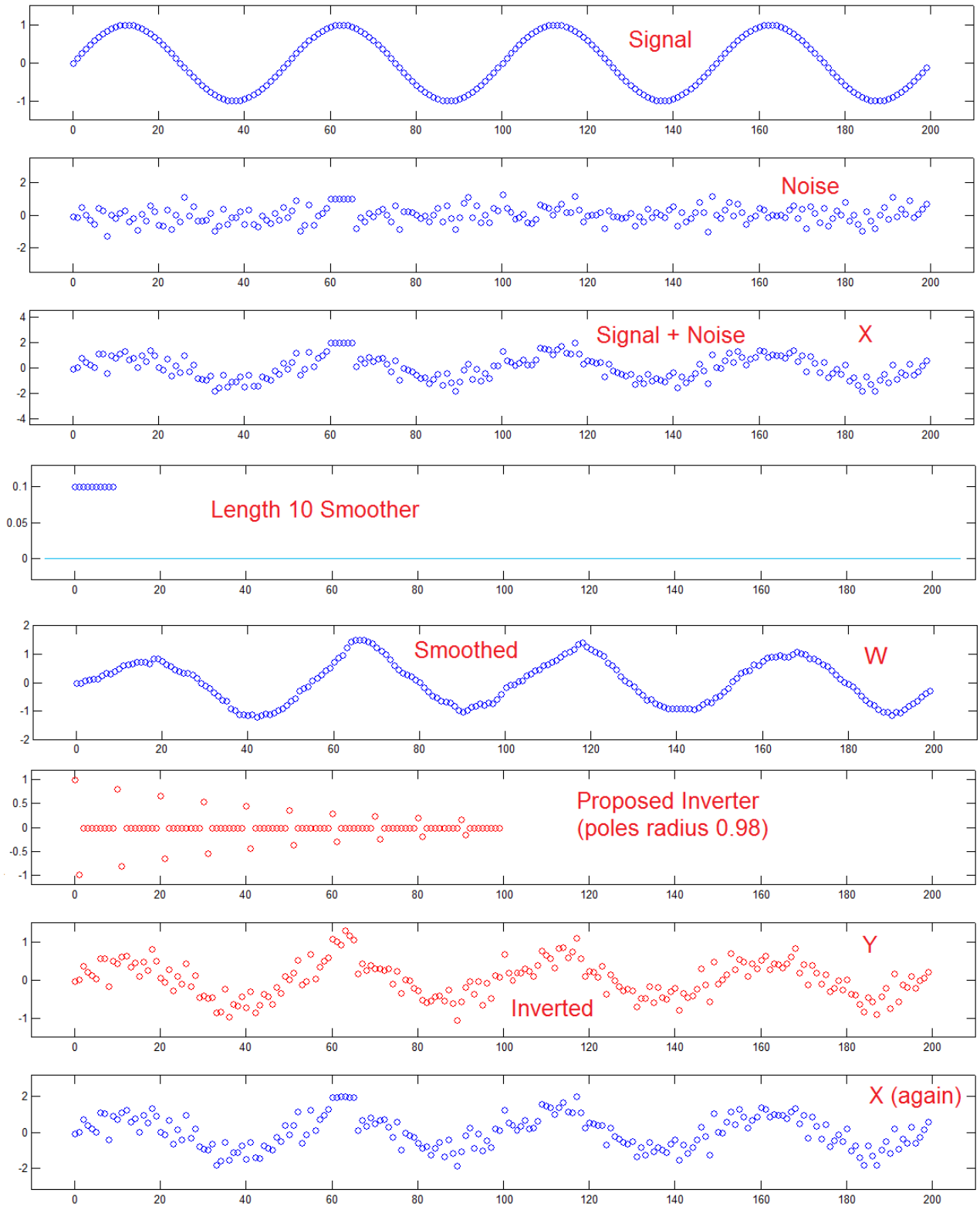

Signal
Noise
Signal + Noise
X
Length 10 Smoother
Smoothed
W
Proposed Inverter
(poles radius 0.98)
Y
Inverted
X (again)

Fig. 8

REFERENCES

All here written by B. Hutchins for Electronotes

[1] "Unfiltering – Equalization", Electronotes Application Note AN-366, May 2006
http://electronotes.netfirms.com/AN366.pdf

[2] Analysis of Moving Average is available in just about any DSP text, or here:
http://electronotes.netfirms.com/EN197.pdf

[3] "Moving Averages and Single Sinewave Cycles", Electronotes Application Note AN-375, November 2011
http://electronotes.netfirms.com/AN375.pdf

[4] "Averaging - and Endpoint Garbage", Electronotes Application Note AN-395, March 30, 2013
http://electronotes.netfirms.com/AN395.pdf

[5] "Yearly Moving Averages as FIR Filters", Electronotes Application Note AN-401, Dec 22, 2013
http://electronotes.netfirms.com/AN401.pdf

[6] "Spurious Correlations Due to Filtering (of Noise)", Electronotes Application Note AN-403, Jan 27, 2014
http://electronotes.netfirms.com/AN403.pdf

[7] "Savitzky-Golay Smoothing", Electronotes Application Note AN-404, Feb 13, 2014
http://electronotes.netfirms.com/AN404.pdf

[8] "Removing "Outliers" from Averaging", Electronotes Application Note AN-405, Feb 22, 2014
http://electronotes.netfirms.com/AN405.pdf