

April 6, 2014

## **CHANGING THE MOVING AVERAGE LENGTH – A RECIPE**

### INTRODUCTION

A number of previous app notes [1-4] have dealt with “Un-Filtering”, particularly in the form of inverting of the previously implemented smoothing of sequential data points, usually with the Moving Average (MA) as the prime example. A variety of issues were dealt with, including a direct inversion (IIR) approach vs. a least-squares (FIR) approach; a variety of smoother types; and the influences of noise. For this note, we do not concern ourselves with these additional issues – just the simplest MA inversion.

Thus in this note we shall, with appropriate cautions (meaning - test with known data!), concentrate on the direct inversion, with the additional feature of changing the numbers in the moving average length. Further, a “recipe” presentation will be made for the most part.

The recipe includes the previously noted fact that the direct inversion of a length-N MA was a convolution of a non-decaying sequence formed by starting with a sequence +1, -1, and then adding on to that length-2 sequence enough 0’s to fill out the length of the MA, and then repeating this length-N sequence for enough times to “outlast” the sequence to be inverted, and then watching out for the “echo”. Since we find this completely inverts the averaging, it is clear that we could follow this inversion with a new averaging of any different length. Combining the inversion with a re-averaging is an obvious modification of the inverting sequence, convolving it with the new MA.

### THE RECIPE

Form the basic non-decaying inversion sequence. As in the references, we form this as follows:

- (a) Begin with a 1 and then a -1.
- (b) Add enough zeros to make a length-N sequence.
- (c) Repeat this length-N sequence enough times so that it is longer than the smoothed sequence to be inverted.
- (d) If you wish to only invert back to a shorter MA length, convolve the sequence of step (c) with the shorter length MA rectangle.
- (e) Convolve (c) or (d) with the input.

**Example:** Change from a length-7 MA to a length-5 MA. Assume the sequence (x) to be inverted is length-18.

(a)  $h = [1 \ -1 \ ]$

(b)  $h = [1 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0]$

(c)  $h = [1 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0]$  (>18)

(d)  $h_5 = h$  convolved with  $[1 \ 1 \ 1 \ 1 \ 1]$   
 $= [1 \ 0 \ 0 \ 0 \ 0 \ -1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ -1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0]$

Notes: Use (c) to just invert to no smoothing, (d) to invert to shorter smoothing. Observe that in (d) while we convolve with a long sequence MA of 1's, the [1 -1] groups in (c) zero out major portions. Always test the procedure on well-defined test data.

### CASE 1 FULL REVERSAL

At this point, we will take a look at three examples of program runs. The Matlab code used to generate the graphs is given at the end of the note for specificity, and perhaps as a guide to programming. So this is a review of the previous two app notes. In terms of the current program, we have N1 (starting length) = 7, and N2 (final length) = 1.

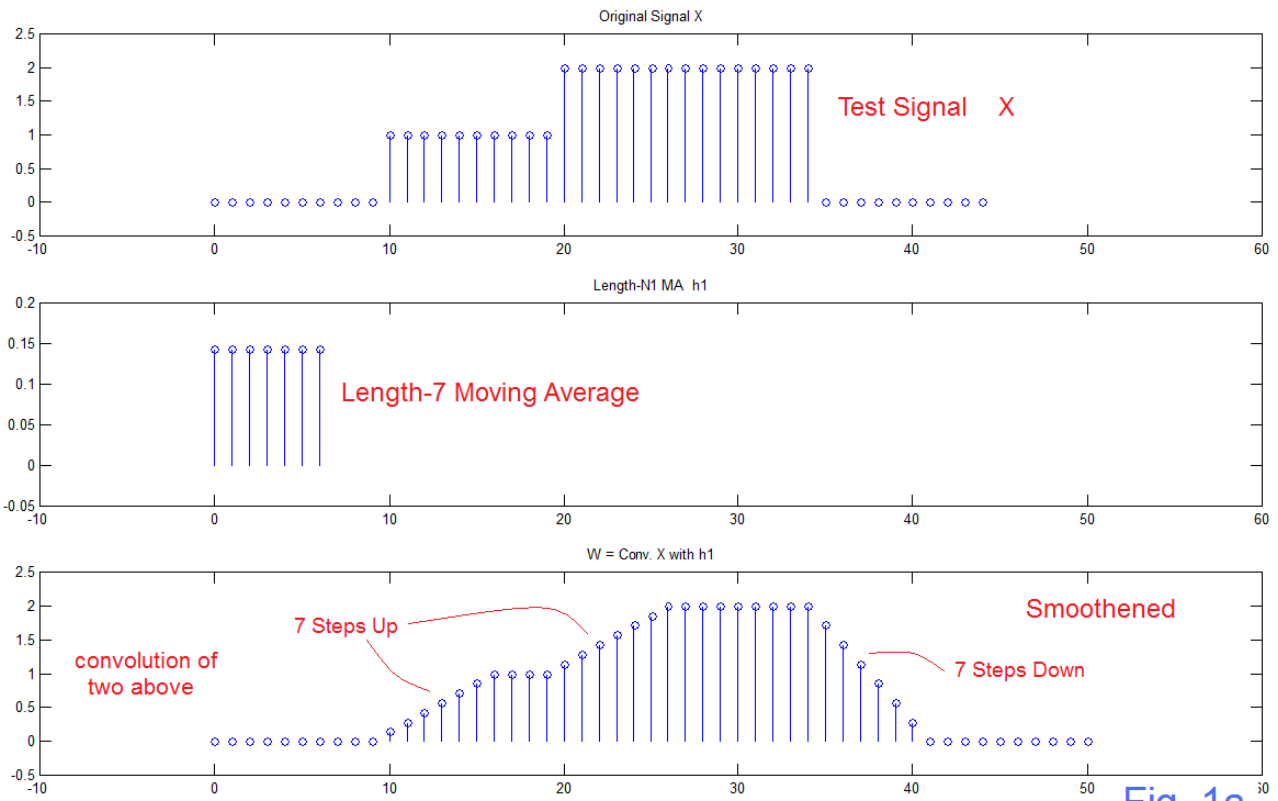


Fig. 1a

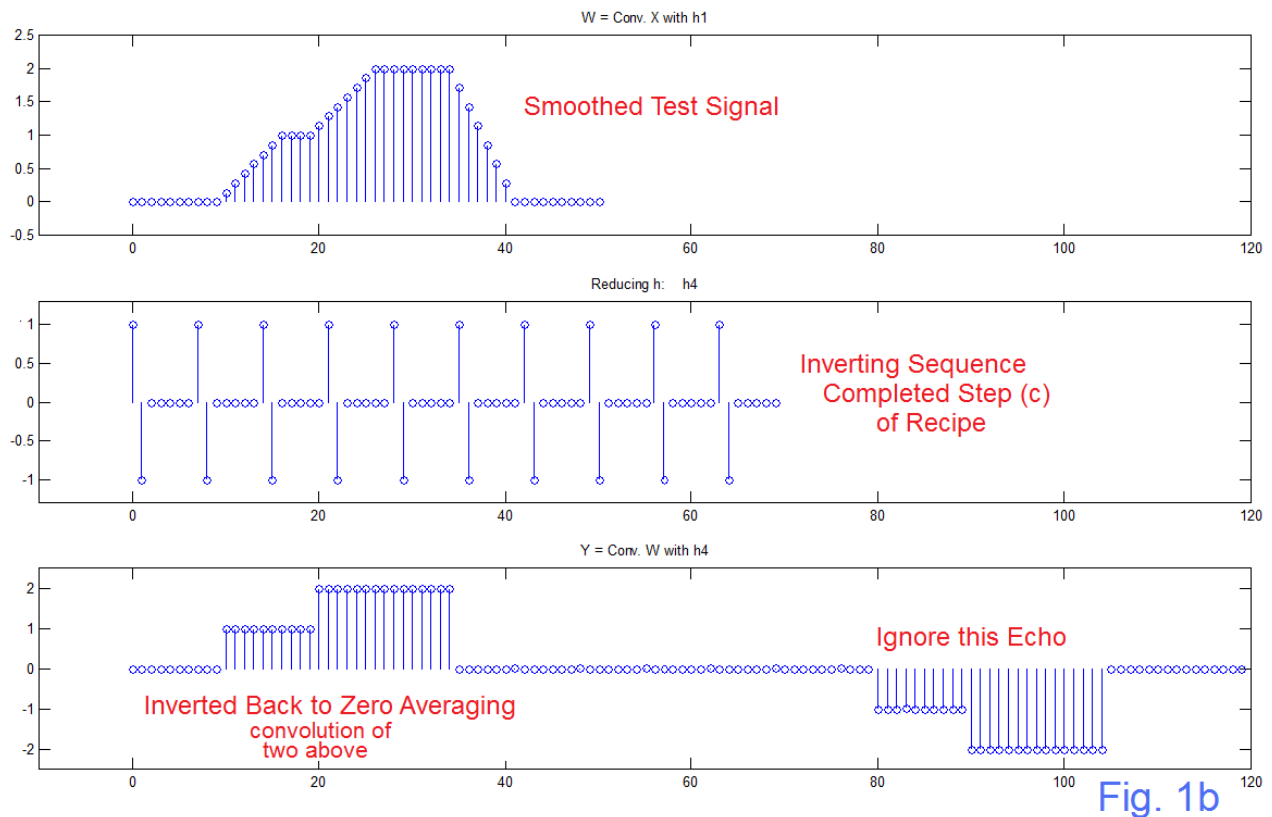


Fig. 1b

Fig. 1a (top) shows a test signal consisting of a step of height 1 and length 10 followed by a step of height 2 of length 15. This is smoothed by a MA of length 7 (middle). Appropriately the bottom panel of Fig. 1a shows ramps of 7 steps between level changes. The top of Fig. 1b shows the same smoothed signal, with the inverting sequence shown in the middle. This inverting sequence is formed according to steps (a), (b), and (c) of the “recipe”. Note that the inverting sequence overlaps and extends beyond the signal to be smoothed. This is done in order first of all to get the right answer (bottom panel). Secondly, the “inverted echo” begins after the sequence is truncated (at about sample 70), so the right side extension moves the echo out of the way.

## CASE 2 PARTIAL REVERSAL

What is new now is to show a case where the length of the MA is changed. One obvious way to do this would be to take the original length-7 average and invert it, as in Fig. 1b. Then we can simply do a new average of whatever length we choose (longer or shorter) since we have recovered the original data (assuming we discarded the echo). That means we convolve the smoothed sequence with first the inverting sequence, and then convolve the recovered original again, this time with the new averaging sequence. Because convolution commutes, we can do this in another order: convolve the inverting sequence with the new MA, and then convolve this with the data as smoothed by the original MA. Indeed this second approach appears to be more computationally efficient, although likely this speed/computation issue is seldom important. We look at it both ways to see if we gain insight. So for the moment, we believe the method of complete inversion, followed by a new averaging, needs no further elaboration.

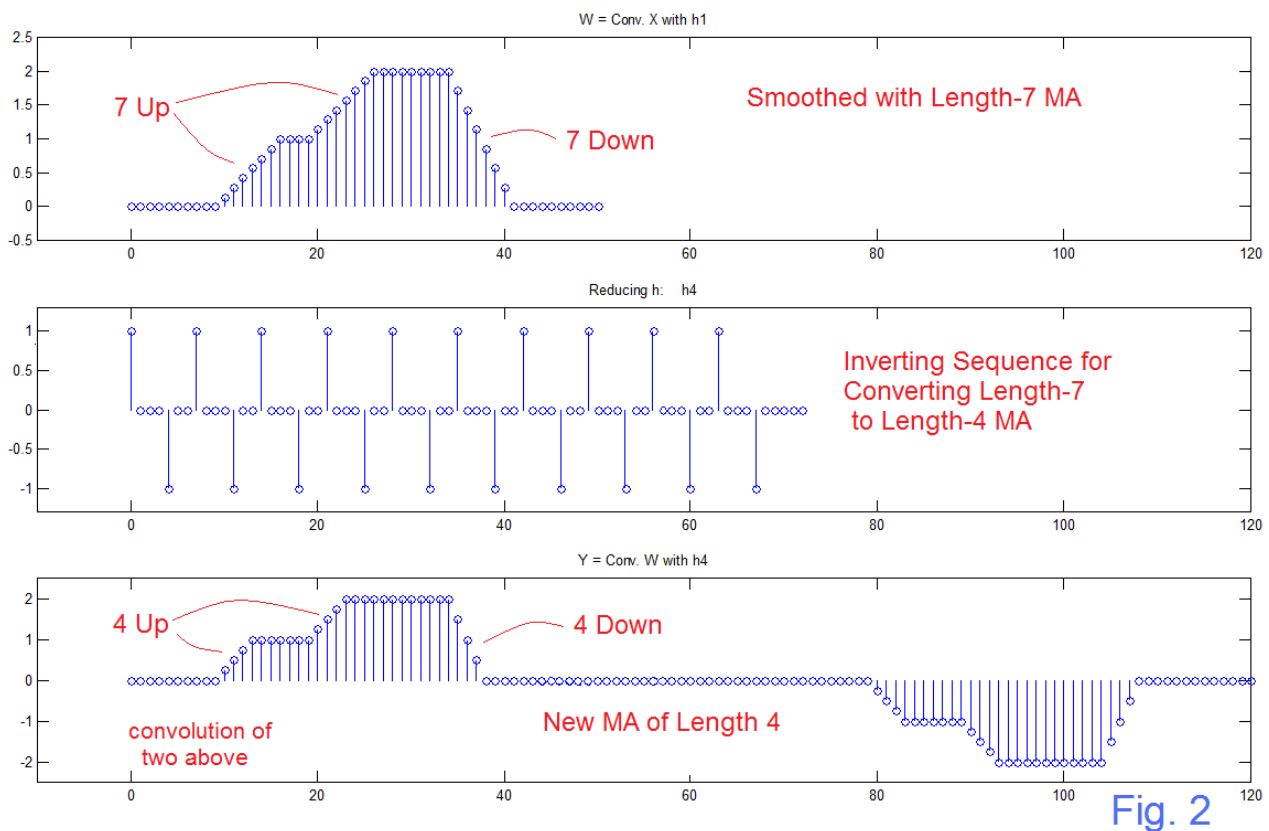


Fig. 2 shows the result where our signal as smoothed by a length-7 MA is converted to the case that would have been obtained if the original data (as top panel of Fig. 1a) had been smoothed with a length-4 MA. Observe first that things worked out fine. In the top panel of Fig. 2, we have the three changes of level being accomplished with 7 steps each. In the bottom panel of Fig. 2, these same transitions are accomplished with just four steps each. The transition slopes are of course steeper with the shorter number of steps. So this works, and may be handy.

So the real curiosity is in examining what the conversion sequence looks like, and the answer to this is immediately available in the middle panel of Fig. 2. It is perhaps simpler than we might have been supposing. Like the inverting sequence of Fig. 1b (middle) this involves reputed segments of length 7, with one +1, one -1, and five zeros. They are just moved. And it is simple enough to see that the results are correct from the standpoint of convolution. Note well the “conspiracy” of the +1 and the -1 in cancelling out so many terms that might have supposed would all be non-zero. It does seem too simple, but it is not unusual either to find solutions that seem to be more “clever” than we ourselves suppose.

### CASE 3 IMPULSE - HOW TO GET FOOLED – HOW IT WORKS

In my teaching career I always suggested that it was perfectly okay, if not indeed recommended, to say in a report what sort of false assumptions (pitfalls) were uncovered during a study. This was because another researcher, indeed even the incidental reader

of a report, could logically be found to be following in similar footsteps – patterns of thinking in this case.

In this instance of thinking about reducing the length of a MA I was thinking in terms of the number of steps in a ramp. Since a length-N MA changes a step into a ramp of N steps, isn't it the case that changing the length of a MA changes the number of steps? Certainly that is true – see examples above. But is that the fundamental thing it does. NO!

In trying to get a step-by-step grasp on the calculations here (why DID that sequence of Fig. 2, middle, work!), I thought in terms of manipulations of samples, and was unable to even write down what seemed to be the correct equations. Facing such difficulties, you generally instinctively go to the simplest case – which would be an impulse. **NEW FLASH** – an impulse is not a step! But this needed to be revealed by the test. We just plug a different test signal (just an impulse) into the program and see what comes out. OUCH!

This is exactly right (Fig. 3) – the Length-N MA of an impulse is just a repetition of the impulse N times scaled by 1/N. And here we have done a slightly simpler 5 to 3 length change, so we have another example of the converting sequence (middle of Fig. 3). Note that Fig. 3 shows the conversion of a width-5 rectangle to a width-3 rectangle. It is easy to see how the impulse response sequence in the middle panel, when convolved with the top, gives the bottom, at least for the parts that yield a zero result, as the 5 input samples always overlap one +1 and one -1. So the three samples in the bottom panel at the left are the result of the 5 samples moving (from the left - **green arrow**) into the impulse response, and encountering the first +1 for three shifts, prior to reaching the first -1.

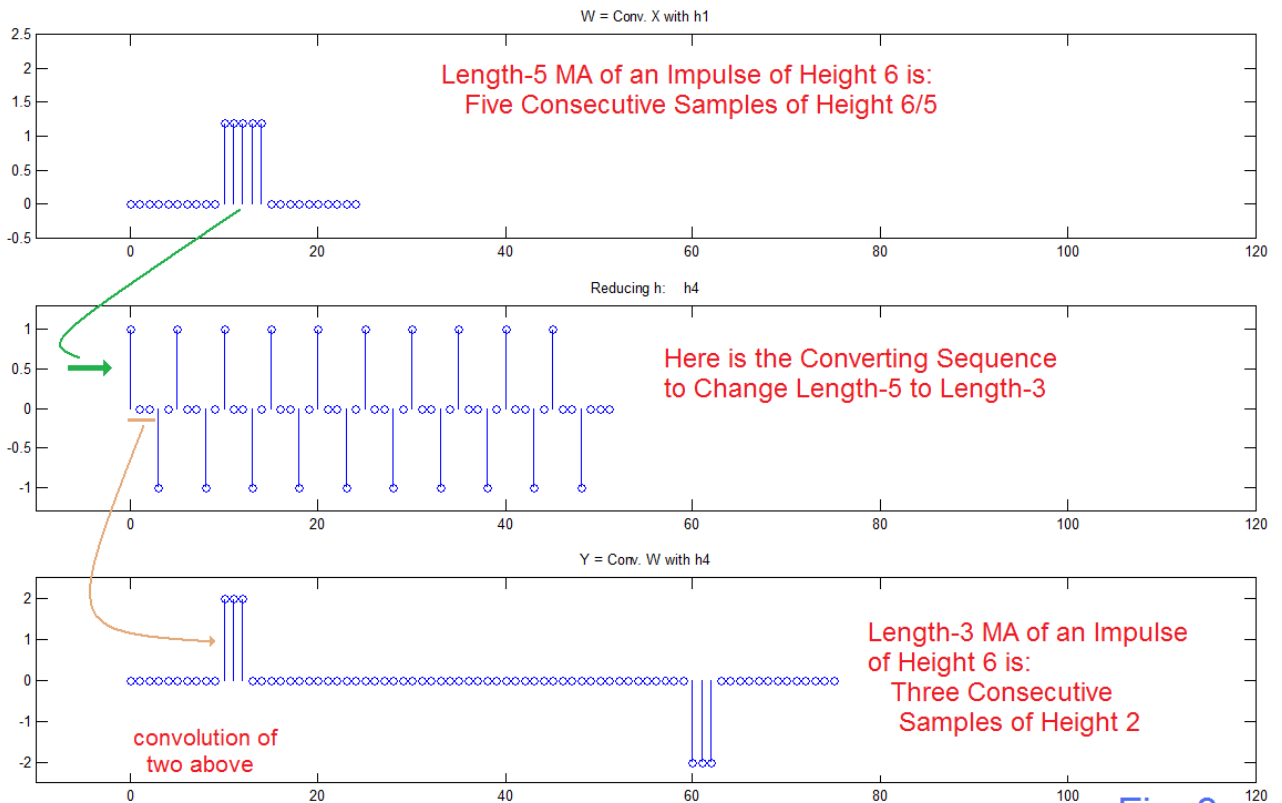


Fig. 3

We have envisioned the length conversion as part 2 of a process – the first part being the original MA. Hence the impulse response is subject to a cascade, of which the second part is seen as the convolution in Fig. 3. The first part was easy enough. If we reverse the order of the operations however, the picture is harder to draw although the

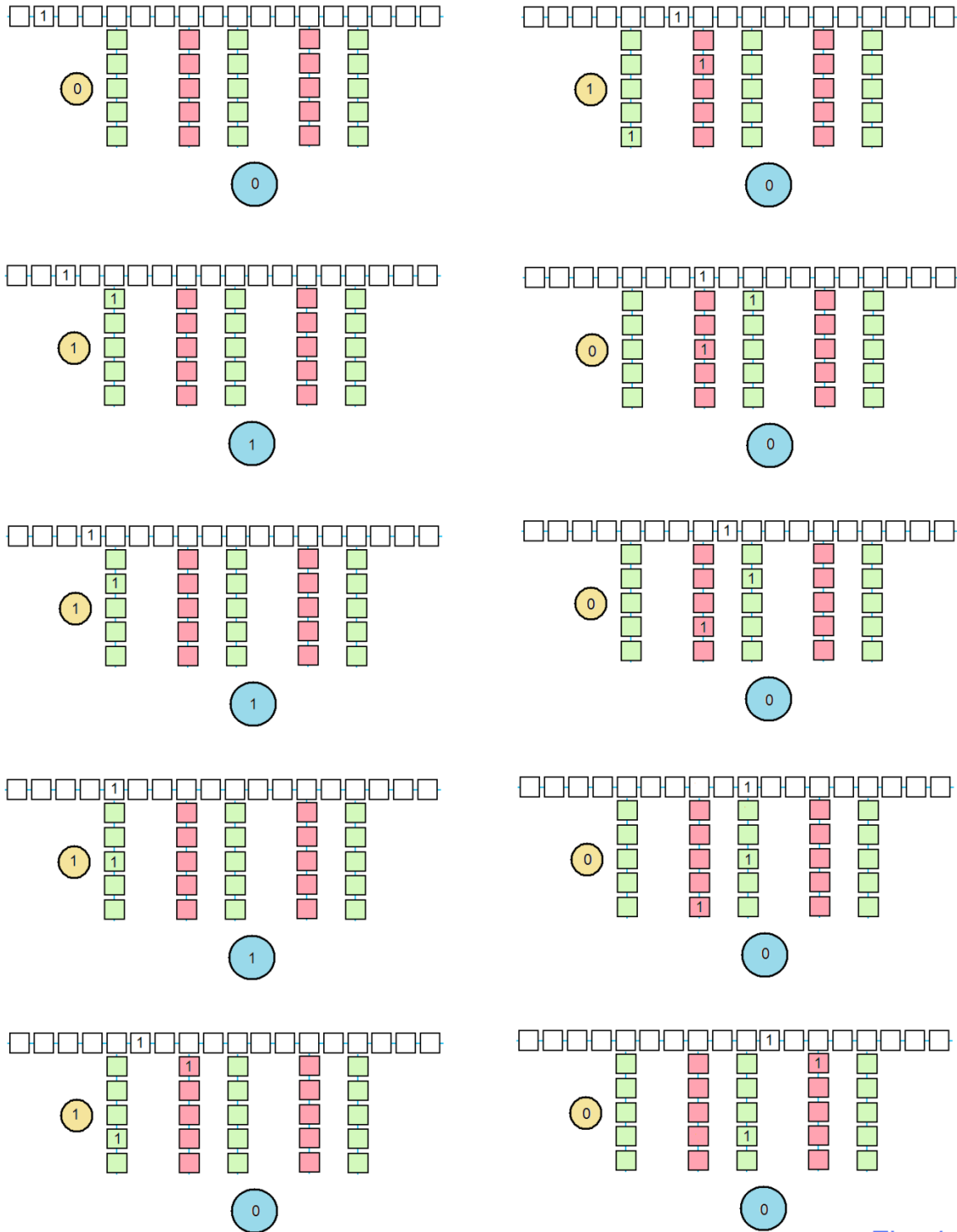


Fig. 4

results will be the same. In Fig. 4 we show, as a sequence of detailed steps, how an impulse in will end up width three by the scheme. Here note the 10 instances (in two rows) where an impulse moves along the horizontal delay line (white boxes) one step at a time. All the empty boxes are zeros. Here, unlike the first ordering of operations, the length-5 MA's follows, each as a length-5 vertical delay line (green to be added, pink to be subtracted in a grand total reported in the blue circles). The MA's are taken as below the center of the five samples being monitored, and no MA's are provided for taps that would be zero according to Fig. 3 (middle panel). So the impulse moves down the line, left to right, and will pass through (get stuck temporarily in) the different length-5 MA's. The "game" here is clear enough. Note the essential findings. The first MA shows five consecutive 1's and that's it forever (yellow circles). In keeping track of the total population on the whole set of MA's, there are three consecutive 1's and that's it (blue circles). Mostly, there will be a 1 in a green box and a cancelling 1 in a pink box. Yes - there will eventually be the echo.

The scheme of Fig. 4 is perhaps more complicated, while at the same time perhaps showing in more detail how a conspiracy has been implemented. The examples here and in previous notes [1,2] will likely convince the reader that this is simple and that it works. We recommend that in addition to trying any actual application with well-defined data, that some sketches like we have done here also be done – for the entertainment value if for nothing else.

## REFERENCES

All here written by B. Hutchins for Electronotes

- [1] "Examples of Inverting of Smoothers" Electronotes Application Note AN-408, April 5, 2014  
<http://electronotes.netfirms.com/AN408.pdf>
- [2] "Inverting a Moving Average Operation" Electronotes Application Note AN-407, March 30, 2014  
<http://electronotes.netfirms.com/AN407.pdf>
- [3] "Unfiltering – Equalization", Electronotes Application Note AN-366, May 2006  
<http://electronotes.netfirms.com/AN366.pdf>
- [4] "Yearly Moving Averages as FIR Filters", Electronotes Application Note AN-401, Dec 22, 2013  
<http://electronotes.netfirms.com/AN401.pdf>

# PROGRAM

```
% lsian409
N1=5
N2=3
x=[0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 0
0 0 0 0 0 0];
%x=[0 0 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0 0 0 0 0 0 0 ] % impulse test option

h1=ones(1,N1)/N1;
w=conv(x,h1);
LH1=length(h1);
LX=length(x);
LW=length(w);

figure(1)
subplot(311)
stem([0:LX-1],x(1:LX))
axis([-10 60 -0.5 2.5])
title('Original Signal X')
subplot(312)
stem([0:LH1-1],h1)
axis([-10 60 -0.05 0.2])
title('Length-6 MA h1')
subplot(313)
stem([0:LW-1],w(1:LW))
axis([-10 60 -0.5 2.5])
title('W = Conv. X with h1')
figure(1)

h2=[1 -1 zeros(1,LH1-2)];
h2=[h2 h2 h2 h2 h2 h2 h2 h2 h2 h2];
h3=ones(1,N2);
h4=conv(h2,h3);
LH4=length(h4)
y=conv(h4,w);
y=y*(N1/N2);
LY=length(y)
%
figure(2)
subplot(311)
stem([0:LW-1],w(1:LW))
axis([-10 120 -0.5 2.5])
title('W = Conv. X with h1')
subplot(312)
stem([0:LH4-1],h4)
axis([-10 120 -1.3 1.3])
title('Reducing h: h4')
subplot(313)
stem([0:LY-1],y(1:LY))
axis([-10 120 -2.5 2.5])
title('Y = Conv. W with h4')
figure(2)
```