ELECTRONOTES
1016 Hanshaw Rd
Ithaca, NY 14850

APPLICATION NOTE NO. 374

August 2010

A VIEW OF LINEAR FREQUENCY INVARIANT SYSTEMS

While on the one hand, we think we are free to jump between the dual variables (time and frequency) of the various Fourier Transform relationships, and we generally are, it sometimes proves difficult to actually throw the switch in the brain, especially as we are accustomed to using principles and procedures in mainly only one of the two directions. For example, previously we explored the notion of “Time Aliasing” [Application Note 324] where a spectrum was sampled resulting in overlapped copies of a time waveform. Here we will want to twist our notion of a Linear Time Invariant (LTI) system into that of a Linear Frequency Invariant (LFI) system. Everything here is trivial.

Virtually all “signals and systems” courses begin with the notion of LTI systems and their properties. First of all, they are linear. Secondly, the time-invariance means that when we shift a particular signal in time, and present it to the input, the output is just shifted in time by the same amount. We also learn that the eigenfunctions of an LTI system are exponentials, and the eigenvalues are a corresponding “frequency response”. Further, an LTI system does not have any frequencies present at the output that were not present at the input. So, what would an LFI system be like?

First of all, it would be linear. Secondly, the output would be invariant under a shift in frequency. Real world devices such as frequency shifters and single-sideband modulators suggest the sort of operations we have in mind. With some risk of confusion, we nonetheless will choose a simple frequency shift operation as a type case of a LFI system to study. We will rely on examples using the FFT. At this point, an initial example is overdue.

Fig. 1 (top) shows the case of a length 20 signal that has a length 20 FFT such that the values of the FFT for k = 3, 4, and 5, are set to 1, 2, and 3. Note that we are not choosing the time domain signal directly, but rather, its FFT. Having chosen the FFT as real numbers, and without symmetry (for simplicity), we know that the time signal will be complex, but we don’t propose to even look at this. We want to shift this spectrum upward by 5 positions, and double the values, thereby obtaining the result in the bottom portion of Fig. 1. How do we do this?

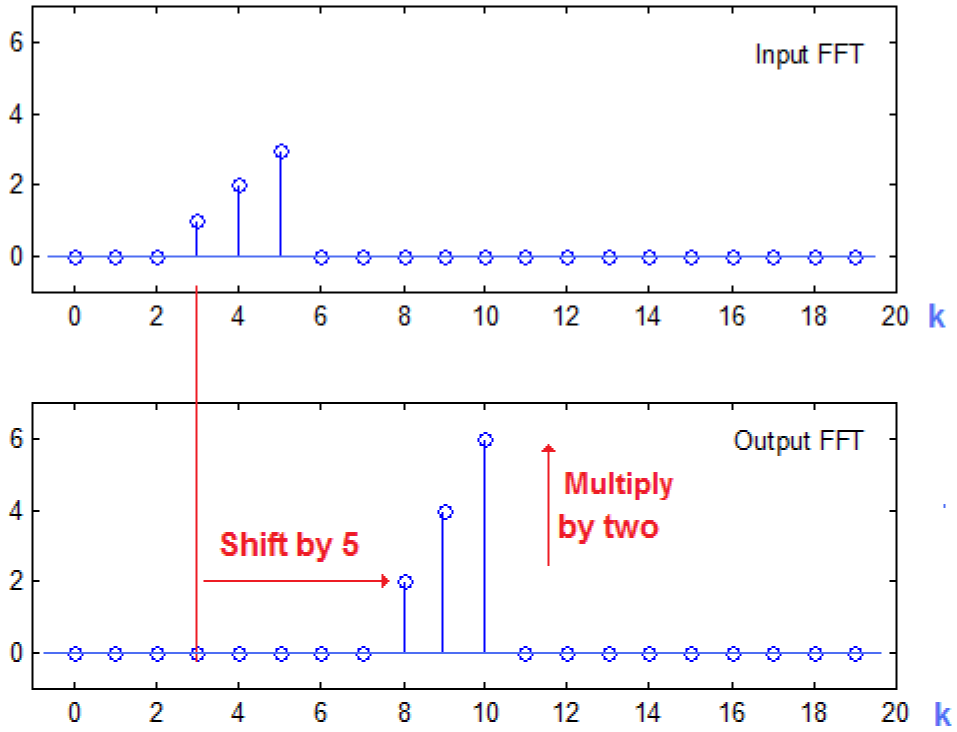


Fig. 1 A spectrum (FFT) is shifted up 5 positions and multiplied by 2

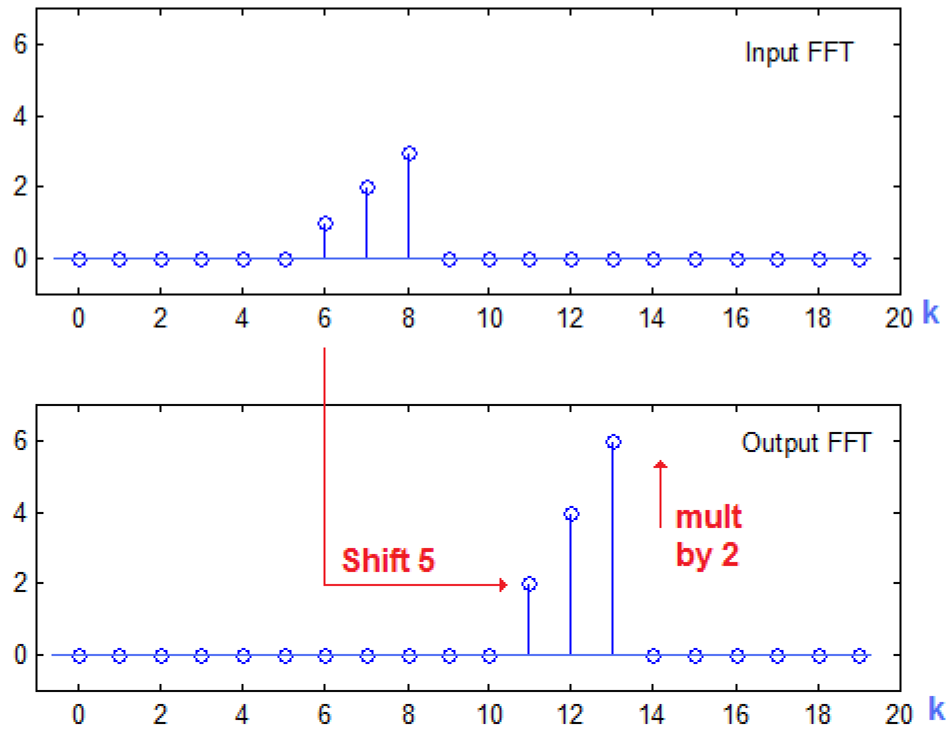
We can see that we want to convolve this with a delta-function at k=5 (with weight 2 actually). Alternatively, we know that we can shift a frequency by multiplying by a time exponential of frequency k=5. These are equivalent operations of course, and the "Code Snippet 1" shows how this is written in Matlab. Thus we choose the specified values for the desired FFT for X(k) and use the inverse FFT to get x(n) (not shown). We compute a length 20 shifting sequence s(n) with frequency 5/20. (The frequency corresponding to k=1 is of course 1/20.) Then we multiply (point-by-point) the time sequences x(n) and s(n) giving xs(n), and take its FFT XS(k). That's it.

With the same procedures, we can demonstrate the frequency invariance and linearity, as in Fig. 2, Fig. 3, and Fig. 4. Fig. 2 looks a lot like Fig. 1 except here we have started the input FFT at k=6 instead of at k=3. The result is that the output is likewise shifted up by three relative to Fig. 1. This illustrates the frequency invariance.

To illustrate the linearity, we will use a different input FFT, the one shown in Fig. 3, which has values of 1 at k=5 and k=6. We see that this is shifted up by 5 and multiplied by 2 as in the first two test cases. Adding the two FFT's we get Fig. 4, values of 1, 2, 4, and 1 at k=3, 4, 5, and 6. These are shifted and multiplied by 2 as shown. Thus superposition works and linearity is illustrated.

<u>CODE SNIPPET 1 (FOR FIG. 1)</u>

```
X=[0 0 0  1 2 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0];  % original spectrum
x=ifft(X);                                      % time signal
s=exp(j*2*pi*(5/20)*[0:19])                     % shifting signal
xs=s.*x;
XS=2*fft(xs);                                   % shifted spectrum
figure(1)
subplot(211)
stem([0:19],real(X))
axis([-1 20 -1 7])
title('Input FFT')
subplot(212)
stem([0:19],real(XS))
axis([-1 20 -1 7])
title('Output FFT')
```



<u>Fig. 2</u> Moving the test input spectrum to start at k=6 (instead of at k=3) shifts the output to start at k=11 (instead of k=8) illustrating frequency invariance

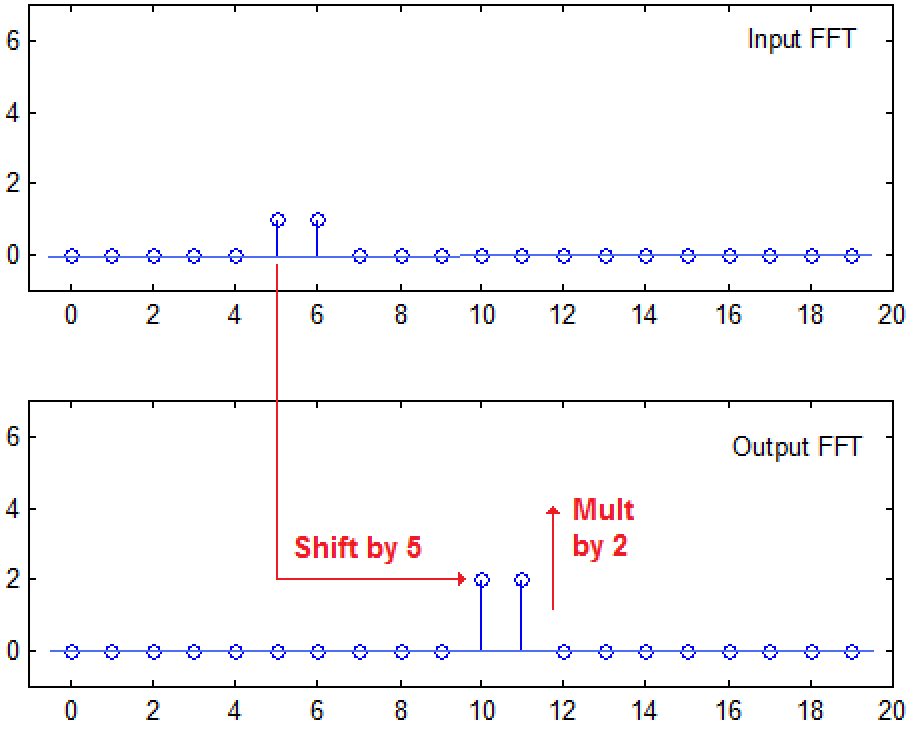


Fig. 3 A second test sequence will be used to show linearity

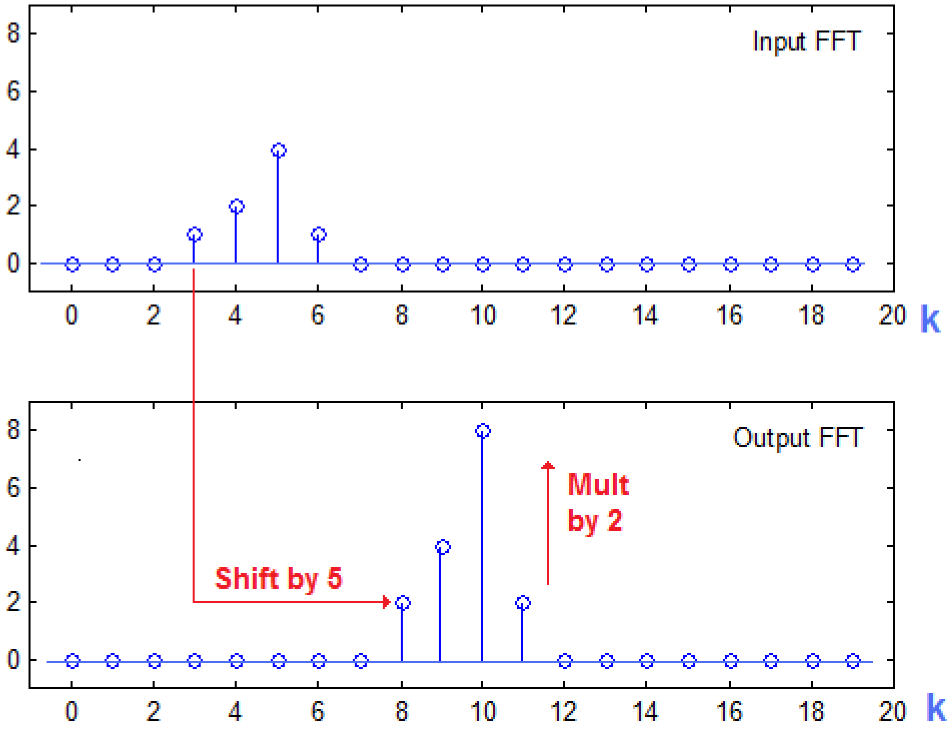


Fig. 4 Linearity illustrated by summing Fig. 1 and Fig. 3

What happens in the time domain when we employ an LFI system?

With an LTI system, no new frequency components appeared at the output, while with the LFI system, we certainly DO have new frequencies. However, by analogy, it should be true that with a LFI system, no new time components should appear at the output. It will turn out that this is obvious, but for the moment, we will just play with Matlab code and learn from it.

So, instead of starting with a known FFT, we want to begin with a chosen (real) time sequence. Once chosen, its FFT can be computed, and this is familiar and known to be generally complex, generally non-zero, and with symmetries for a real time signal. Now taking the FFT, we can shift the spectrum as in the Code Snippet. In our illustration here, we take a length 20 time signal that is 1 at n=4 and 2 at n=5 and zero elsewhere. Fig. 5 (top two parts) shows the real and imaginary parts of the FFT of this signal. Multiplying by the shifting sequence as in the first examples, we find a shifted FFT as in the bottom two portions of Fig. 5. We see that we do achieve the shift by 5 positions, exactly as we should.

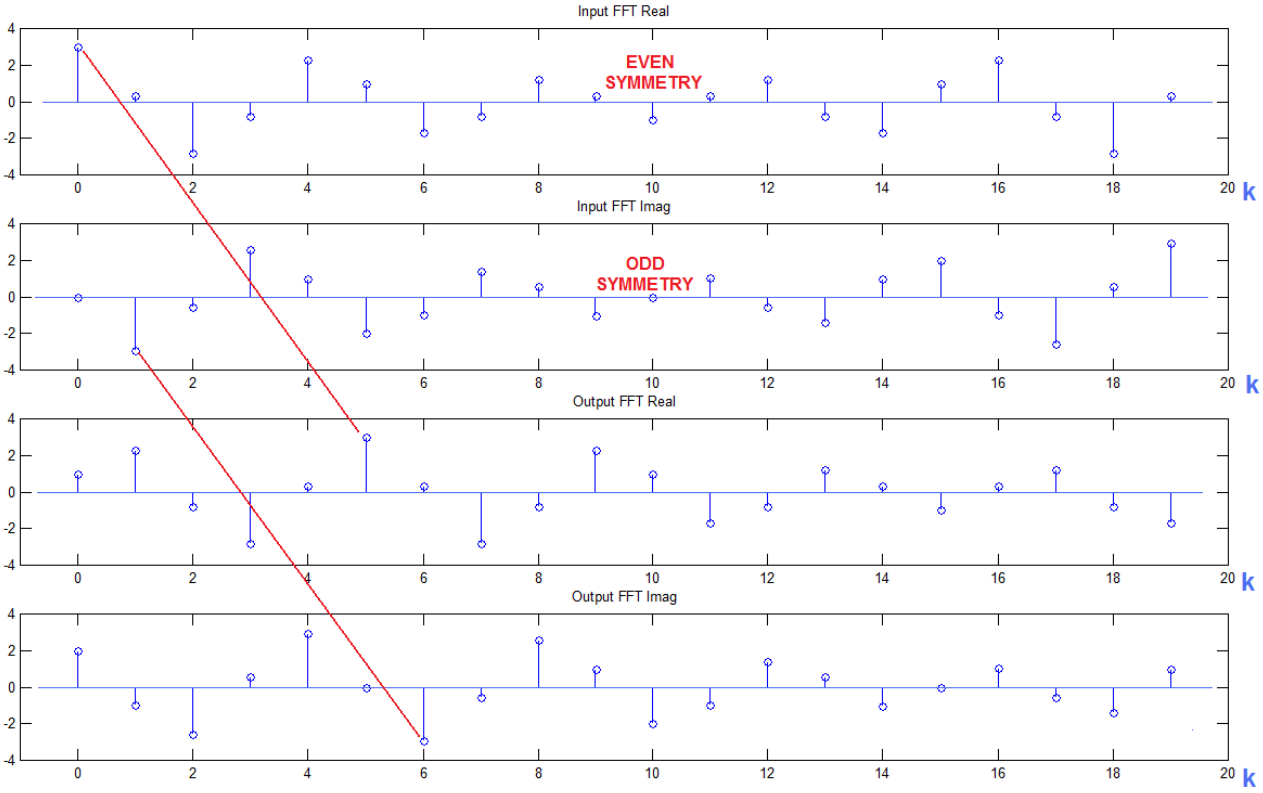


Fig. 5 Choosing a particular time sequence (not shown here), we compute its FFT (two top portions – real and imag parts) and shift it up giving the bottom two portions. Note: symmetry is gone. See note at end about wrap-around.

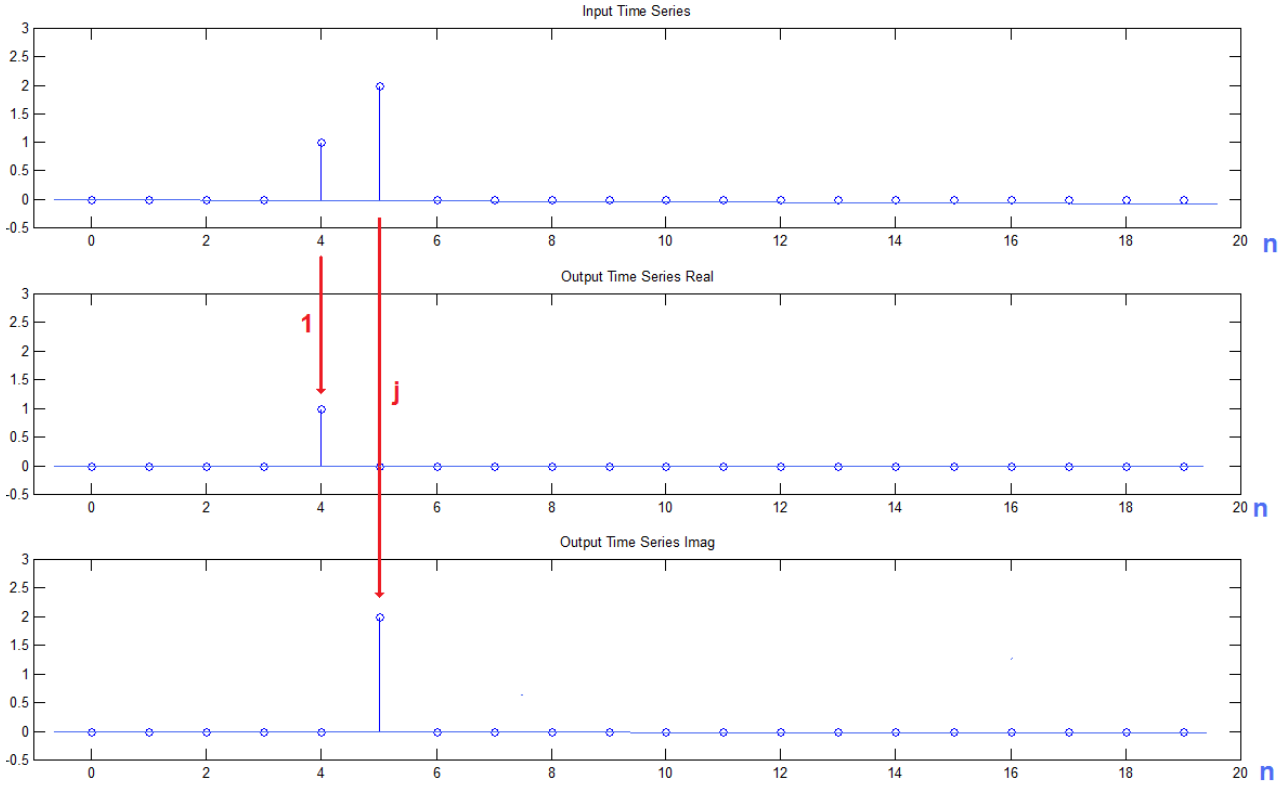


Fig. 6 The top portion shows the input time sequence. The bottom two portions show the real and imaginary parts of the output time sequence. No "new time" positions are occupied.

The remaining step is to take the inverse FFT of the shifted spectrum. Fig. 6 shows the result. The top portion of Fig. 6 shows the real time sequence we chose – simply two non-zero real values. The bottom two portions show the real and imaginary parts of the output time sequence. Note that the output sequence is complex, which we might have predicted because the shifting of the original FFT of course upsets the symmetry characteristic of a real signal. (The magnitudes of the time samples are unchanged.)

However, perhaps the most significant finding is that the output is non-zero only for the two time positions (n=4 and n=5) that were non-zero at the input. This might seem to have been a tall-order considering the multiple operations that led to this output. But of course, the "no new times" was what we expected of a LFI system, corresponding to "no new frequencies" in an LTI system. It was supposed to come out that way.

Of course, there is a larger swindle going on here. Recall that we started with a time sequence, and took the FFT. This FFT was then the input to a procedure (the Code Snippet 1) of which the first step was to take the inverse FFT (IFFT). Accordingly, we just got the time signal back. This was multiplied, point-by-point with the shifting exponential, s(n).

While we did not say so, this frequency 5, length 20 time sequence is just the values:

[1 j -1 -j   1   j -1 -j   1   j -1 -j 1 j -1 -j 1 j -1 -j]

which as multiplied by the input time sequence:

[0 0 0 0 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

of course giving:

[0 0 0 0   1 2j 0 0   0 0 0 0   0 0 0 0   0 0 0 0].

Unless there is a non-zero value in the input, the output has to be zero (no new times).

*   *   *   *   *   *   *   *

From the example of Fig. 6, and the linearity of the LFI system, we can achieve any set of frequency shifts (and associated weights for each shift) by multiplying one overall input time sequence by one overall shifting sequence, called g(n) below. The second codes snippet gives the example Matlab code for this idea.

CODE SNIPPET 2

```
function y = lfi(x,w)
% x=input time sequence
% y=output time sequence
% w=weight sequence
%     all sequences length 20 here
g=zeros(1,20);
for k=0:19
   s=exp(j*2*pi*(k/20)*[0:19]);
   g=g+s*w(k+1);
end
y=x.*g;
Y=fft(y);
G=fft(g);
```

With this coded function (plus appropriate display code) we can easily replicate the result of Fig. 6 with x = [0 0 0 0 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0] and w=[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]. Note that the overall frequency shifting sequence is only dependent on the weights w, just as a frequency response of a LTI system (commonly denoted H) depends only on the impulse response (commonly denoted h). In both the LTI system and the LFI system, the "system function" is independent of the actual input x.

Consider an additional example. We start with x being an exponential of frequency k=1 and propose to shift it by 2. Thus x=exp(2*pi*j*[0:19]*1/20) and w = [0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]. The results are shown as Fig. 7.

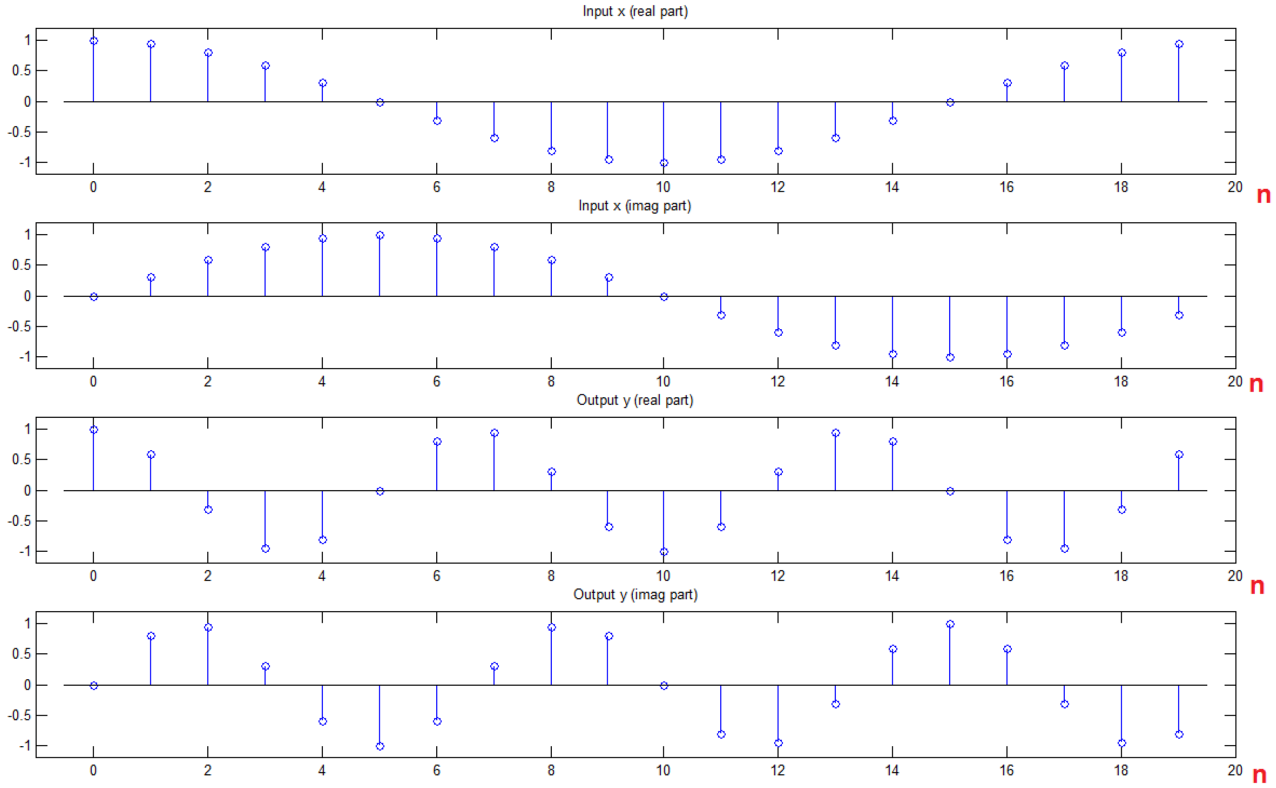


Fig. 7a Complex exponential (frequency k=1) shifted up by 2 to k=3. Time domain view.

Fig. 7a shows the result pretty much as we would expect. (Because we are using the FFT as our tool between the time and frequency domains, we need to work with complex exponentials – see more discussion below). The top two portions of Fig. 7a shows the input, clearly of frequency 1/20 (k=1) while the bottom two portions clearly show a frequency that is three times as high.

We need not show the corresponding frequency domain plots of X and Y as X is merely a real value of 20 at k=1 while Y is merely a real value of 20 at k=3, as we expect. It is much more informative to look at the system function g (and it's FFT G). These results are shown in Fig. 7b. The simple result is that this is only a complex exponential with frequency k=2. We could have guessed this, but it is comforting to see that Matlab agrees.

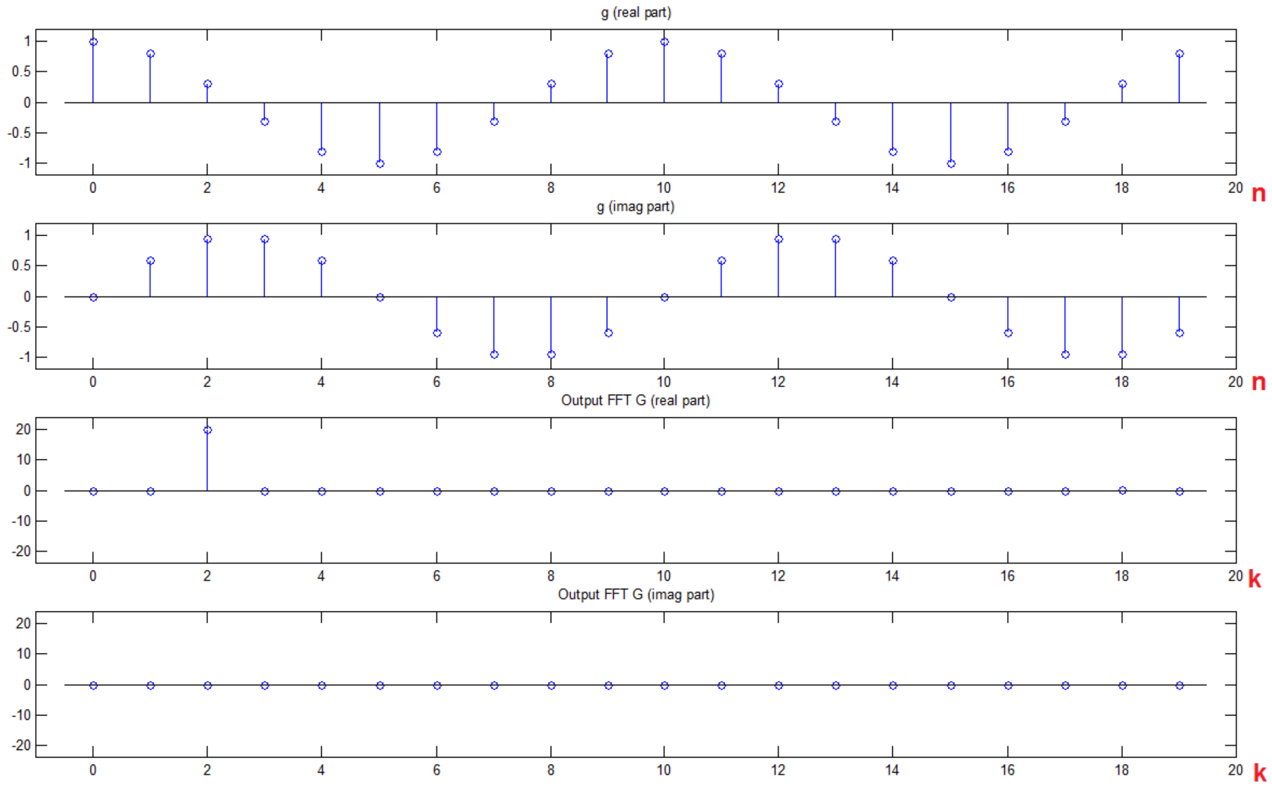
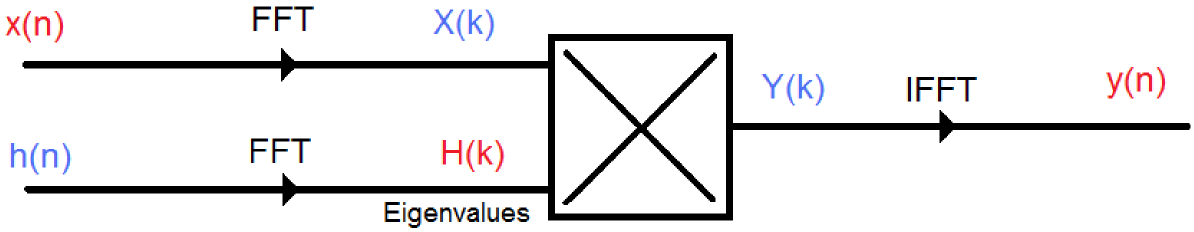


Fig. 7b The system function g (and G) is just a complex exponential of k=2

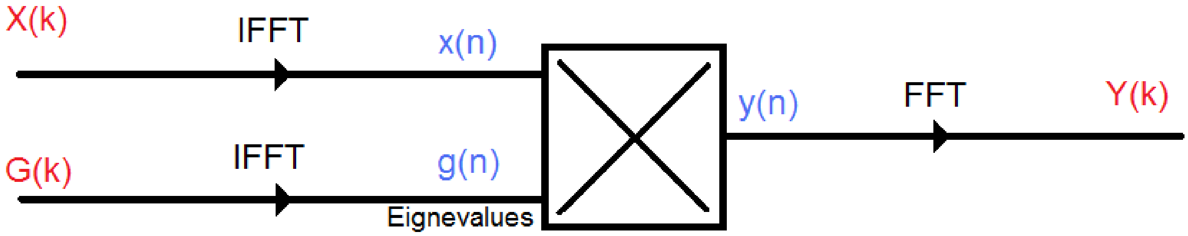
We have been using the FFT in our study. In order to understand what is happening in an LFI system, we need to be clear what happens in an LTI system, vis-à-vis the FFT and Matlab. In Fig. 8, we compare LFI systems to LTI in the context of the FFT. In an LTI system, if we want to filter, we start with a length N time domain waveform, x(n) and find its spectrum (length N FFT). Then we have a notion of what we want our filter to do, and we represent this as N points in the frequency domain, H(k), and we think of this as a frequency response (in the sense of an FFT). The inverse FFT of the frequency response is the impulse response (again in the sense of the FFT). We do a multiply in the

Fig 8a LTI



$$y(n) = h(n) \circledast x(n)$$

Fig 8b LFI



$$Y(k) = G(k) \circledast X(k)$$

$\circledast$ =circular convolution

frequency domain (point-by-point multiply of the FFT of the input by the FFT of the filter). Taking the inverse FFT of this product gives us the time-domain output sequence, y(n). In Fig. 8a, we have shown the scheme with our values of primary interest in red, with the corresponding (FFT) values in blue. Note that the multiplication shown, Y(k)=H(k)X(k), is indicated as well by the corresponding circular convolution. The "eigenvalues" of the LTI system are the frequency response values, and the eigenfunctions are the "DFT Harmonics" $e^{-j(2\pi/N)nk}$.

As we would expect, the dual, the LFI system, has a similar diagram as in Fig. 8b. In as much as we have been looking at this is terms of a frequency shifter, the emphasis is slightly different, such that the spectra are the values of the most interest. Also, unlike the LTI case, it is not the lower input to the multiplier g(n) but rather its FFT G(k) that is primary. That is, we think primarily of the shifts in frequency. The shifting functions g(n) are the implementation of our specifications. Hence the red and blue emphasis. The time-domain multiply has the corresponding frequency domain circular convolution, which is exactly how we envision our frequency shifting (and find it realized in Matlab). The eigenvalues here are the g(n), and the eigenfunctions are the FFT of the DFT harmonics, i.e., δ(k).

There is some unfinished business here before we come to what is really an unremarkable conclusion. First we have been thinking of g(n) as being composed of a sum of shifting sequences (as in Code Snippet 2 where we construct such a sum). Fig. 8b on the other hand suggests g(n) is merely the IFFT of G(k). Indeed, if we choose G(k) to be the FFT of the desired frequency shifts (with weights), g(n) can be easily obtained

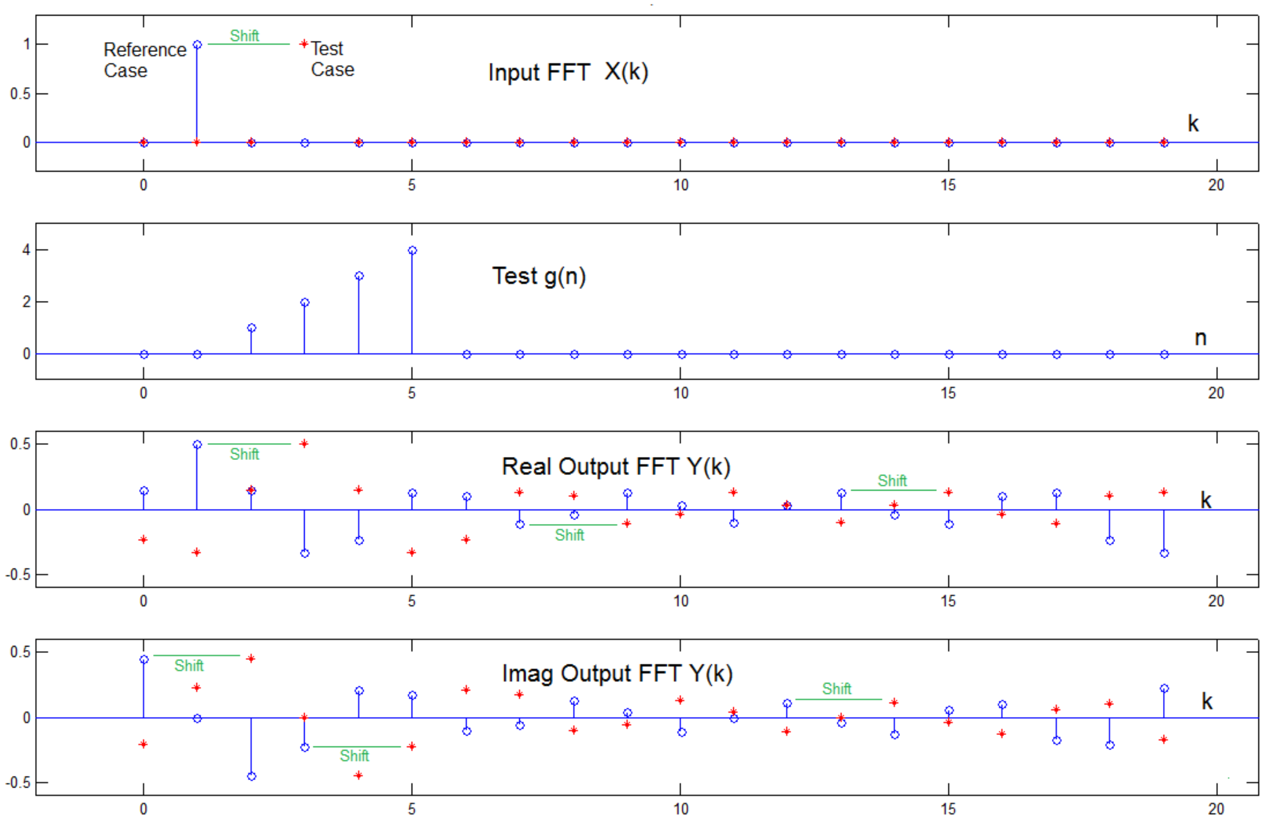


Fig. 9 Here we test the notion that g(n) is really always a sum of frequency shifting signals. An arbitrary test g(n) is on the second line. The first line shows two caese to be frequency (blue stems and red stars). The output FFTs are just shifted (bottom two lines) by the same amount. See note later on wrap around.

using an IFFT, exactly as in 8b. Turning this idea around, it is thus clear that any arbitrary g(n) has to correspond to some G(k), and thus any g(n) should work as LFI. This is what Fig. 9 shows. We start with two test frequencies, k=1 (blue stem) and k=3 (red star) in the top line of Fig. 9. For simplicity, these are taken as purely real. The test sequence g(n) is arbitrarily chosen as 1, 2, 3, 4 for n=2, 3, 4, and 5. Proceeding exactly as in Fig.

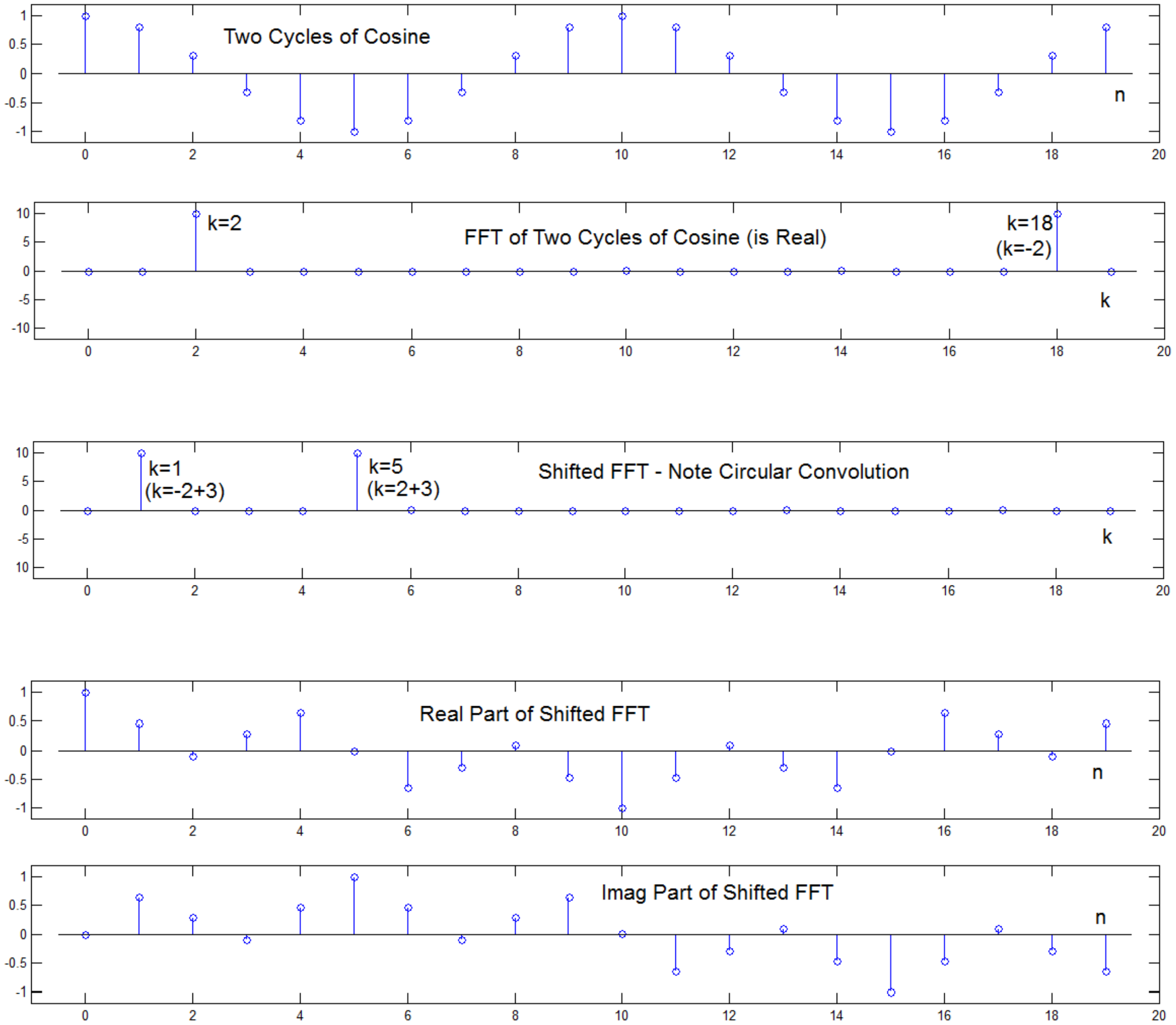


Fig. 10 The frequency shift here is circular, since the use of the FFT implies periodicity, and convolution is circular. Here a single cosine appears to have two different frequencies after being shifted (see text)

8b, we obtain (now complex) FFT's as outputs (bottom two lines of Fig. 9) corresponding to the two different inputs. Note that each blue stem has a red star two places to the right of it (a few indicated by green lines). We conclude a LFI result for an arbitrary g(n).

We note that the shifting of the spectrum here is circular. So far we have been careful that no spectrum or shifted replica ran past the end, and we opted for single rather than two sided spectra (and complex signals where necessary), or have mentioned the "wrap-around" as a matter to treat later – HERE. Fig. 9 was a case where all frequencies "lit up" and we had to deal with the circular wrap around. There is nothing new or unusual about this. But it can be confusing. In particular, Fig. 10 shows the case of a cosine (rather than a single complex exponential). Recall that Euler's formulas tell us:

$$\cos(\theta) = [\ e^{j\theta} + e^{-j\theta}\ ] /2$$

so the top two lines in Fig. 10 show us two cycles of a cosine in the time domain and two non-zero values (purely real for the cosine) at k=2 and k =18. The k=18 is equivalent to k = -2. A frequency shift of 3 rotates these (middle line of Fig. 10). The k=2 rotates (shifts up) to k=5, as expected. But, the k = -2 also rotates in the same sense, and so we add 3 and ends up at k=1. Taking the inverse FFT here, we end up with a complex signal (bottom two lines of Fig. 10). We can easily see the k=1 and k=5 components here. This is correct, and a caution when using circular convolution. [Practical frequency shifters involve Hilbert transformer phase splitters, and two multipliers, etc.].

The final thing we need to tie up is the idea that superposition applies, and we suggested this with our generation of weightings, but did not give an actual example. Here at the same time we can specify the shifts by giving the FFT G(k). A Code Snippet 3 is given below:

CODE SNIPPET 3

```
function Y=lfig(X,G)
x=ifft(X);
g=20*ifft(G);
xg=x.*g;
Y=fft(xg);
```

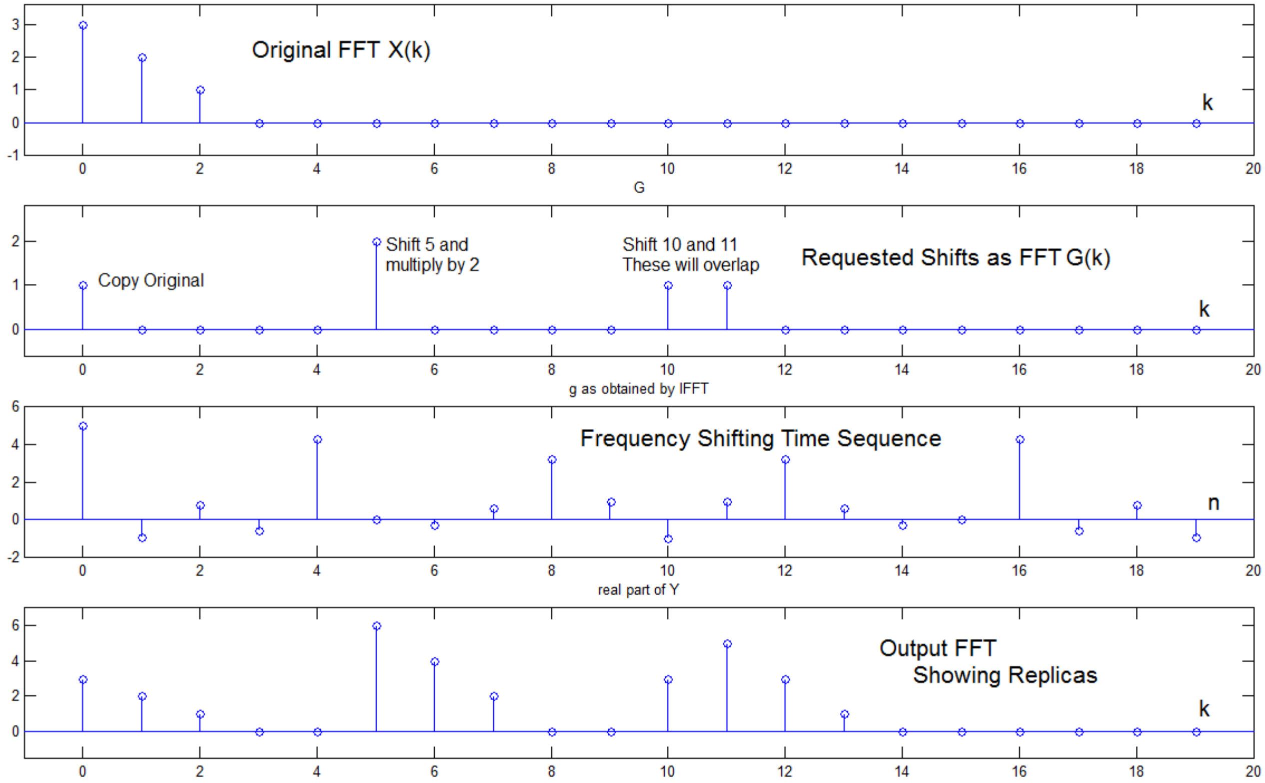


Fig. 11 This shows a shifting series of four replicas set by the FFT G(k).

Fig. 11 shows the case where an original spectrum is replicated about k=0 (the original is retained), about k=5 (multiplied by 2), and as an overlapping pair at k=10 and k=11.

This leads us to: WHY IS THIS ALL OBVIOUS:

BECAUSE IT IS JUST THE MULTIPLICATION PROPERTY OF THE FOURIER TRANSFORM. Multiplying in time shifts in frequency, and it is LFI.