**Part 1**

*Method*

To create a process that will approximate a sine wave with minimal SFDR at an arbitrary, I first consider the phase values which are input to the sinusoidal function over the first 50 periods of the signal. This value is expressed by , where is the fractional frequency. If we let so that the phase value begins at 0, we can iterate over the rest of the phases and find the point at which the phase best approximates 0 again. This will tell us the point at which the signal can be ‘re-read’ with very little distortion because the transition from ‘0’ back to the beginning of the digital readout (technically, back to the second sample, after zero) will be the most seamless. I make the assumption that a minimized distortion of phase over will result in a minimal distortion of SFDR.

*Time Domain Results*

Figure 1 shows the result using this method to find the best index at which to repeat for the case of fractional frequency

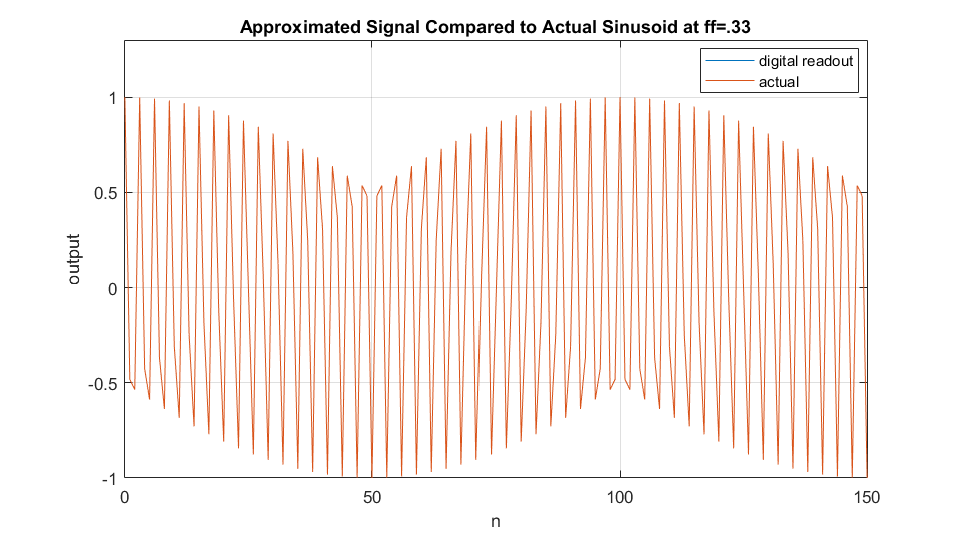


Figure - Time-domain readout compared to the sinusoid it approximates, ff=.33

From Figure 1, we see that the algorithm has successfully selected a repetition point which succeeds in imitating a true sinusoidal signal at least for a short duration of 150 samples. Figure 2 shows the error of the signal compared to a true sinusoid for the first 500 periods of that sinusoid.

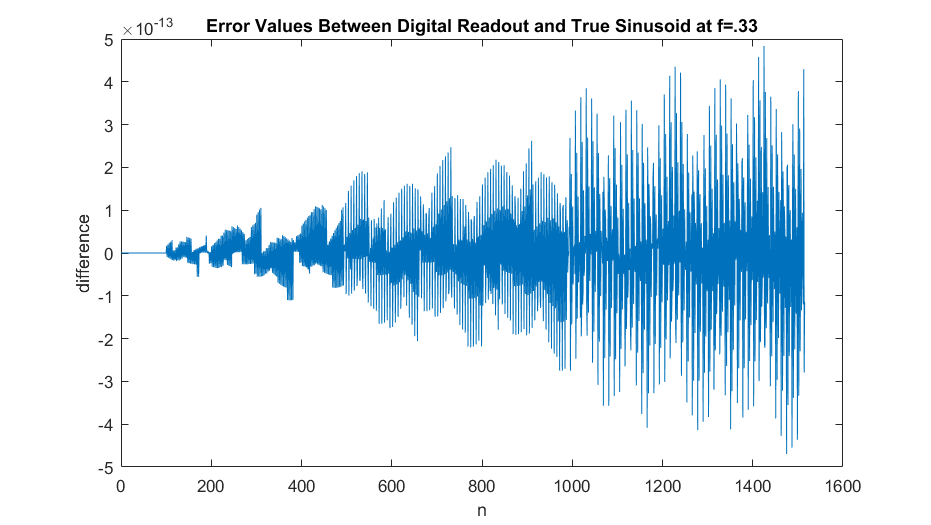


Figure - Digital readout error

Figure 2 shows that the error of the digital readout system is quite small in magnitude, although it accumulates over time. This makes sense considering that the phase of the generated signal gradually drifts from that of the target signal every time the readout is repeated.

*Frequency Domain Results*

Figure 3 presents the resulting signal in the time domain using both Hamming and Blackman windows.

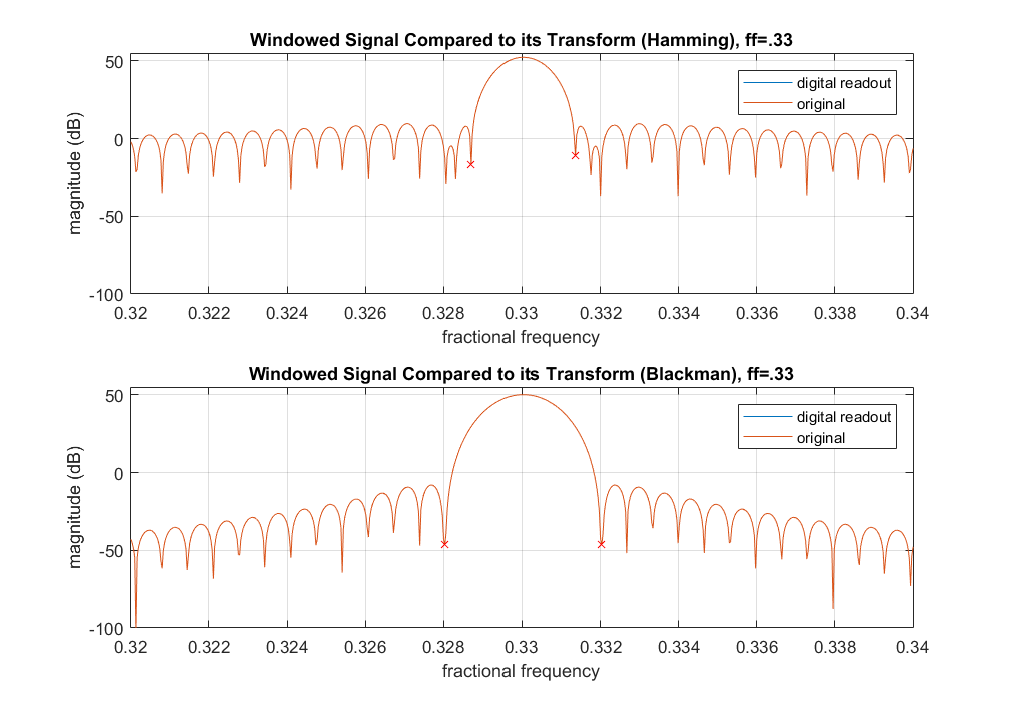


Figure - Comparing the frequency domains using different windows

First, we can make some observations about the window shape themselves. We know that the plots shown in Figure 3 are just scaled and shifted versions of the windows represented in the frequency-domain. This is because we multiplied the windows by a sinusoidal signal and then took the DFT of the result. This operation corresponds to convolving the transform of the window against the transform of a sinusoid which is a scaled and shifted pair of delta functions. The result of this convolution is a pair of scaled and shifted windows. Figure 3 focuses on the frequencies around , so the other transform of the window at is not visible. We can see that the Hamming window has a very narrow main lobe compared to the Blackman window, while the Blackman window drops off much faster and farther than the Hamming window does once outside of the main lobe.

It is worth noting that, for , the readout signal so closely approximates the original that the effects of the windows are more prominent than the signal’s distortion. By finding the SFDR of this signal, we are more or less finding a property that is inherent to the windowing function and not the signal of interest.

*Determining SFDR*

The two red ‘X’ marks on each subplot of Figure 3 denote the boundaries of the main lobe as determined by locate\_mainlobelims, a custom function I wrote (see accompanying files). This utility will assist in determining the SFDR by enabling me to make a distinction between the main lobe and the sidelobes.

My method for determining the SFDR is simply to take the maximum value within the main lobe and compare it against the maximum value among all frequencies outside the sidelobe. The output of my code provides the calculated SFDR of the generated signal for both the Hamming and Blackman windows.

SFDR\_HAMMING =

42.6582

SFDR\_BLACKMAN =

58.1099

The wildly varying results of this calculation dependent only on the window used is further evidence that in this case where the repeated signal is a very close approximation to a true sinusoid, the SFDR is a calculation that is far more dependent on the window than the signal that we are attempting to analyze.

**Part 2**