Generate a swept sine in LabView

Test audio devices by producing a signal on a data-acquisition card.

By Sean McPeak, University of California, San Diego

Swept sine waves let you test a device over a wide frequency range. To implement a swept sine wave with a multifunction data-acquisition card, you need to generate the data points and send them to the card. You can create a swept sine function in National Instruments' LabView with just one VI (virtual instrument) that can control start and stop frequencies, sample rate, and sweep duration. Using the signal, I tested an acoustic transducer used in a research project for measuring wave propagation in the open ocean.

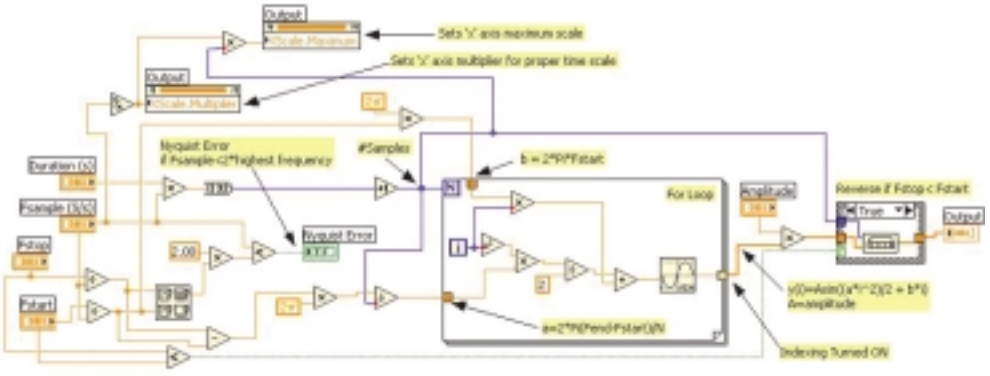

Figure 1 A LabView VI uses an array to calculate the points in a swept sine wave.

The LabView VI (**Figure 1**) calculates an array of numbers that represent the swept sine wave at each sample point as the frequency increases or decreases. To implement a swept sine wave, you must change frequency on a point-by-point basis (Ref. 1) using this equation:

$$y(i) = A \bullet \mathrm{Sin}((a \bullet i^2)/2 + b \bullet i))$$

where $y(i)$ is the amplitude of the swept sine wave as a function of sample point, i is the integer that steps through the time series, and A is the signal's peak voltage.

Variables a and b are defined as:

$$a = 2\pi \bullet (\mathrm{Fstop} - \mathrm{Fstart}) / n$$

$$b = 2\pi \bullet \mathrm{Fstart}$$

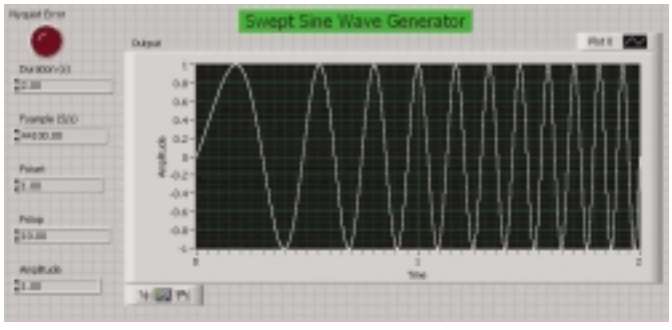

Figure 2 The user panel shows the swept sine waveform, start frequency, stop frequency, duration, amplitude, and Nyquist error indicator.

where n is the number of samples, Fstart is normalized start frequency, and Fstop is normalized stop frequency.

To normalize the start and stop frequencies, you must change the unit to cycles per sample. Do that by dividing the Fstart and Fstop frequencies in Hertz by the sample rate. A good rule of thumb is to use a sample rate of 10 samples/cycle at the highest frequency.

My LabView VI uses array manipulation and a For loop. The inputs are Duration (s), Fsample (samples/s), Fstop (Hz), and Fstart (Hz). The VI converts the start and stop frequencies to cycles/sample by dividing them by the sample rate. A "Max Min" block takes the normalized Fstop and Fstart and determines the maximum frequency input by the user. The VI doubles that value and compares it to the defined sample rate. From the comparison, the VI determines if the sample rate meets the Nyquist criteria for the highest frequency in the signal.

The block's output is a simple Boolean function that tells the user if the sample rate meets the Nyquist criteria. The For loop in the center of the VI diagram runs through the total number of samples that it must calculate, which it finds by multiplying the duration in seconds times the sample rate in samples/s.

To guarantee that the For loop processes all of the samples, you must add 1, because the For loop will stop at N – 1. The For loop implements the output function with algebraic operators and a sine block. The output is an array that goes to the right side perimeter of the For loop. You must enable "indexing" at this node, which lets each element in the array be acted upon individually at the output.

A gain stage after the loop sets the signal's peak-to-peak value. Finally, a case structure uses a "Rotate 1D Array" block to flip

the array around if Fstop is less than Fstart, which lets the VI produce a swept sine wave of descending frequency.

I implemented two property nodes for the graph that set the "x" scale multiplier and the maximum value. These nodes let the plot of the resulting waveform display the proper time scale and maximum value (**Figure 2**). I set the program for a sweep from 1 Hz to 10 Hz over a 2-s duration with a sample rate of 44,100 Hz. The graph palette at the lower-left corner of the graph in Figure 2 lets me zoom in on portions of the sine wave and verify the proper frequency.

I tested the VI with a spectrum analyzer by comparing the signal generated with a multifunction data-acquisition card to that from an arbitrary waveform generator (a waveform generator has this function built in). I also listened to both signals through an audio amplifier and speaker. I found listening useful in determining sweep rate, duration, and stop and start frequencies in the audible range.

You could make several modifications to the VI for increased functionality. For example, you could use this VI with NI data-acquisition hardware to generate a looping up and down frequency sweep. You can also keep track of the output samples and, when finished sending them to the data-acquisition card, reverse the frequency sweep array and feed the data back into the data-acquisition system's output.

Depending on the max/min frequencies, sweep duration, sample rate, and available PC memory, you may not be able to flip the array and configure the data-acquisition system quickly enough to not miss a sample. In that case, you can fill a frequency sweep array for a set number of passes. These modifications would let the sweep continue up and down in frequency for a set period of time. You can also add a real-time FFT (fast Fourier transform), which lets the user see the sweep in the frequency domain. You may find this especially useful for verifying proper start and stop frequencies as well as sweep duration. T&MW

REFERENCE

1. Rowe, Martin, "Generate a swept sine test signal," *Test & Measurement World*, October 2000. www.tmworld.com/article/CA187440.html.