New real-time algorithms for arbitrary, high precision function generation with applications to acoustic transducer excitation

Patrick Gaydecki
School of Electrical and Electronic Engineering, University of Manchester, PO Box 88, Manchester M60 1QD, United Kingdom
Email: patrick.gaydecki@manchester.ac.uk

Abstract. A system is described for the design, downloading and execution of arbitrary functions, intended for use with acoustic and low-frequency ultrasonic transducers in condition monitoring and materials testing applications. The instrumentation comprises a software design tool and a powerful real-time digital signal processor unit, operating at 580 million multiplication-accumulations per second (MMACs). The embedded firmware employs both an established look-up table approach and a new function interpolation technique to generate the real-time signals with very high precision and flexibility. Using total harmonic distortion (THD) analysis, the purity of the waveforms have been compared with those generated using traditional analogue function generators; this analysis has confirmed that the new instrument has a consistently superior signal-to-noise ratio.

1. Introduction
The evolution of high speed DSP devices has resulted in a wide variety of products intended for signal and function generation, operations that have traditionally been performed by analogue circuitry. In fact, the emergence of digitally-based signal generators has enabled certain, specific operations to be realized in real-time that would have been extremely difficult to implement through analogue means. In particular, arbitrary functions are almost impossible to generate using analogue circuitry, precisely because the signals cannot be described by mathematical functions. When implemented using digital technology, arbitrary functions are as easy to produce as well-known standard waveforms such as sine, square, triangle and ramp (saw tooth) functions. Here, we describe a flexible DSP-based unit that is capable of generating real-time waveforms using both the traditional look-up table approach for arbitrary functions, and a new function interpolation technique for synthesis of sine and modulated sine waveforms. The interpolation enables such functions to be generated with very high frequency resolution and moreover, permits frequencies to be altered in real time via interaction with the user through appropriate control software.

2. Hardware description
The conceptual components of a basic digital function/signal generator are conceptually very straightforward, and are shown in Figure 1. A central DSP device, in this case a Freescale DSP56321, operating at 580 million multiplication-accumulations per second (MMACs), is
responsible for producing the signal in digital form; this signal is fed to the digital-to-analogue converter section of a Cirrus CS4271 codec. This is a dual-channel device with 24-bit resolution and user-selectable sample rates from 4 kHz up to 200 kHz. The output of the DAC is connected to an anti-aliasing filter and the final analogue signal is thus sent to the outside world. In addition to the DSP and DAC components, the system incorporates memory and a communications interface through which it receives instructions from purpose-written control software operating on a conventional PC running Microsoft Windows XP or Vista.

3. Direct digital synthesis (DDS) using the look-up table approach
The system exploits the look-up table approach for the playback of pre-computed mathematical functions and user-defined arbitrary signals [1 – 3]. These signals are generated by the PC, downloaded to the memory of the DSP unit and finally transmitted to the DAC in real time. The user interface is depicted in Figure 2. Any number of sine waves with different frequencies, gains, phases and applied window functions may be combined to synthesize signals of arbitrary complexity. Additionally, the wave shapes may be designed by third party software design tools and imported into the program as a text file. One of the restrictions of the look-up table approach is that the length of the function in time is limited by the sample rate of the DAC and the physical size of the memory which represents the look-up table. Hence the user must here also specify the duration of the signal, which in this case is equal to approximately 6000 values. The high level code simply takes the input from the interface and generates the appropriate function with the desired window function.

Once the values have been calculated by, or entered into, the computer program, they must be converted into 24-bit fixed point fractional format, since the DSP563xx family of processors employ fixed point 24-bit architectures. Only then can the values be downloaded into the on-board memory of the DSP system, i.e. the look-up table. To convert the numbers, the software multiplies each coefficient by $2^{31}$, and then stores the result in a long integer. This is because in high-level code, long integers are represented by four bytes (i.e. 32 bits). We multiply by $2^{31}$ since the numbers must span the range $\pm 1$. The three most-significant bytes are then used to form the 24-bit number, with the least-significant byte being discarded.
Figure 2. Real-time arbitrary function interface

Most of the low-level code required to produce an arbitrary function in real-time is actually concerned not with signal generation, but with communicating with the computer to store the values in local DSP memory (the look-up table) and set appropriate address counters. Once this has been completed, the code to send data to the DAC is very short and simple.

3.1. Limitations of the look-up table approach

Although the look-up table represents a simple and intuitive approach to real-time function generation, it has both merits and limitations. One clearly useful feature is its flexibility; a computer can be used to pre-compute very complex functions, the values of which may then be stored and played out in real-time as required. However, the length of the function is restricted by the amount of memory available. Additionally, there is a more fundamental constraint, which concerns the accuracy with which sine waves may be generated at specific frequencies [4]. Consider a sine wave that has been synthesized over a single cycle, with 8 equi-spaced points in the look-up table. If the DAC conversion rate is 1 kHz, then the sine wave generated will be 125 Hz. Now, in order to generate an accurate sine wave, the values held in the look-up table must be such that they end and start with no discontinuities, i.e. typically they must be lie on the zero-crossing points. Hence, if only a single cycle is used in the computation process (as in this example), the next highest frequency that may be generated, using 7 points, is 142.857 Hz. If better frequency resolution is needed, more cycles must be included in the look-up table – i.e. if two cycles are stored, then the next highest frequency that accurately can be generated is \((16/15) \times 125 = 133.33\) Hz.

This clearly means two things: first, the better the frequency resolution needed, the larger the look-up table must be; second; given that an entirely new look-up table must be computed for each frequency, the method is not fast enough to permit frequency alteration in real time.

4. Real-time sine wave generation using interpolation

The system also exploits another means of generating real-time sine waves, which for simple functions, is far more accurate and provides much greater frequency resolution. It uses almost no memory resource, and since it calculates the output in real-time rather than using stored values, it is ideally suited to digital-function generators, in which the frequency may be changed in real-time by the user. This method we term frequency approximation through interpolation. It exploits a polynomial to approximate the sine wave function over a half a
cycle, using the coefficients of the polynomial to compute the function in real-time in the DSP code [5]. Since the sine wave is a smoothly changing analytic function, it can be approximated extremely accurately using, for example, a 4\textsuperscript{th} order fit. Figure 3, for example, actually shows two curves, although only one is apparent: a sine wave function and its approximation using a 4\textsuperscript{th} order polynomial fit. This method is so precise that the quality of the real-time function it produces is better than that generated by expensive analogue instruments.

![Figure 3. Sine wave and its approximation using a 4\textsuperscript{th} order polynomial fit](image)

The high-level software employs Gaussian elimination to obtain the polynomial coefficients. This works as follows: consider the solution for a 3\textsuperscript{rd} order polynomial thus:

\[
\begin{align*}
  y_0 &= a_0 x_0^3 + a_1 x_0^2 + a_2 x_0 + a_3 x_0^3 \\
  y_1 &= a_0 x_1^3 + a_1 x_1^2 + a_2 x_1 + a_3 x_1^3 \\
  y_2 &= a_0 x_2^3 + a_1 x_2^2 + a_2 x_2 + a_3 x_2^3 \\
  y_3 &= a_0 x_3^3 + a_1 x_3^2 + a_2 x_3 + a_3 x_3^3
\end{align*}
\]

(1)

Both the \(y\)-values and \(x\)-values are known, since these are the vertices. The objective is therefore to find the \(a\) coefficients which will allow the polynomial given by

\[
y = \sum_{m=0}^{3} a_m x^m
\]

(2)
to pass exactly and smoothly through the vertices. Gaussian elimination involves re-arranging the equation set and cancelling using back-substitution to allow the determination of the coefficients.

Having calculated the polynomial that describes the function, the coefficients are stored in the memory of the DSP device. Regardless of the frequency selected, the coefficients never change – all that will change is the \(x\)-axis (time) step of the system, which determines the output frequency. To select a different frequency, only a single value is downloaded to the system, i.e. the time step. It is clear that there are several advantages of this system over the look-up table method. First, the memory requirement is minimal, requiring 4 words for the coefficients and one word for the time increment. Second, to alter the frequency, only a single new time step need be downloaded, so the system is ideally suited for altering the frequency on-the-fly. Finally, this system is not constrained to using an integer number of samples to represent a complete cycle. In fact, the frequency resolution is limited only by the word length of the system. It is quite straightforward for this system to produce a digital sine wave with a frequency step of 1 millihertz.
Figure 4 depicts the real-time sine wave generator interface from the control software. It allows the user to generate two independent sine waves on channels 1 and 2, with a maximum frequency of 22 kHz and a frequency resolution of 0.01 Hz. The sliders may be adjusted in real-time to alter the sine wave on-the-fly. In addition, the system allows the user to modulate one channel by the other.

![Real-time sine wave generator interface](image)

**Figure 4.** Real-time sine wave generator interface from *Signal Wizard*

### 5. Results

#### 5.1. Look-up table system

Figure 5 (left) depicts an arbitrary waveform designed using the high-level software; Figure 5 (right) shows the same signal captured from a digital oscilloscope, having been downloaded to, and generated by, the hardware in real time. Quantitative analysis confirms that the real-time signal is a very precise replica of the design-time waveform.

![Design trace](image)

**Figure 5.** Design trace (left) and actual output (right) from system.

#### 5.2. Interpolation system

Figure 6(a) depicts a 1 kHz sine wave produced by a commercial laboratory analogue function generator (FG); its dB frequency spectrum is shown in Figure 6(b). In contrast, Figures 6(c) and 6(d) also show a sine wave at 1 kHz and its spectrum, produced by the DSP function generator operating in interpolation mode. It is evident that the DSP system has a considerably superior noise performance, and additionally, it maintains its spectral resolution over the entire range of its operation.

![Spectrum](image)

**Figure 6.** Spectrum for the different systems.
requires more memory resource and is constrained with respect to its frequency resolution. In contrast, the interpolation method is most suitable for the generation of simple sine waves. This technique is far less demanding on storage and furthermore permits real-time frequency alteration with very significant precision. Comparisons of this latter method with standard analogue instruments confirm superior performance respecting signal-to-noise ratio and total harmonic distortion.

![Figure 6. THD analysis. (a) 1 kHz sine wave produced by analogue function generator; (b) its dB spectrum. (c) 1 kHz sine wave produced by DSP function generator; (d) its dB spectrum](image)

References

[1] Humez J C 2009 Sine waves in programmable logic, *Elektor Electronics* 35 38-42
[2] Cordesses L 2004 Direct Digital Synthesis: A Tool for Periodic Wave Generation (Part 1), *IEEE Signal Processing Magazine* 21 50-54
[3] Cordesses L 2004 Direct Digital Synthesis: A Tool for Periodic Wave Generation (Part 2), *IEEE Signal Processing Magazine* 21 110-117
[4] Abeyasekera, T, Johnson C M, Atkinson D J and Armstrong M 2003 Elimination of sub harmonics in direct look-up table (DLT) sine wave reference generators for low-cost microprocessor-controlled inverters, *IEEE Trans. Power Electronics*, 18 1315-1321
[5] Ashrafi A, Adhami R, Joiner L and Kaveh, P 2004 Arbitrary waveform DDFS utilizing Chebyshev polynomials interpolation, *IEEE Trans. Circuits and Systems*, 51 1468-1475