

An Expanded and Improved Ultra Fast Magic Sinewave Calculator

Don Lancaster

Synergetics, Box 809, Thatcher, AZ 85552

copyright c2008 pub 7/08 as **GuruGram #94**

<http://www.tinaja.com>

don@tinaja.com

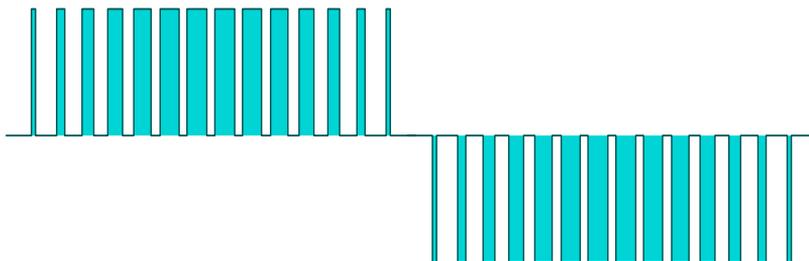
(928) 428-4073

Magic Sinewaves are a newly discovered class of mathematical functions that hold significant potential to dramatically improve the efficiency and power quality of solar energy synchronous inverters, electric hybrid automobiles, and industrial motor controls, among many others. An executive summary can be found [here](#), a slideshow type intro presentation [here](#), a development proposal [here](#), the latest calculator [here](#), demo chips [here](#), and additional design info [here](#).

Major goals of such digital sinewave generation including offering the **maximum possible efficiency** by using the fewest of simplest possible switching transitions; offering the **lowest possible distortion** by zeroing out a maximum number of low harmonics that impact power quality, whine, vibration, and circulating currents; and by using **all digital techniques** that are extremely low end microprocessor and/or microcontroller friendly.

Magic sinewaves have two remarkable properties: **Any number of desired low harmonics can be forced exactly to zero** in theory, and to astonishingly low levels when quantized to 8-bit compatible levels. And magic sinewaves use the **absolute minimum possible and simplest energy-robbing transitions** to achieve such harmonic suppression.

A typical **magic sinewave** might look something like this...



We see that this waveform is a variation on **PWM** or **pulse width modulation**.

Its highly unique characteristics are that it has far fewer energy robbing transitions than conventional PWM, that it is always exactly phase- and frequency locked to a fundamental, and uses half bridge rather than full bridge switching events for further efficiency improvement. Additional advantages include a 100 percent modulation depth allowing the carrier to **never** exceed the fundamental.

Plus, of course, zeroing out any chosen number of low harmonics and doing so with an absolute minimum of switching events.

There are several different types of Magic Sinewaves possible. Three of emerging interest are called **Best Efficiency**, **Bridged Best Efficiency**, and **Delta Friendly**. A **Best Efficiency Magic Sinewave** zeros out an additional two harmonics. When compared to conventional earlier solutions. Brought about by an invisible and zero integrated width pulse at zero degrees.

A bridged best efficiency is similar but is continuous at 90 degrees. And fills in with alternate values. A **delta friendly** magic sinewave meets the exacting special needs of three phase power systems. There are fewer of these at present, limited to **3, 7, 11, 15, ...** or more pulses per quadrant. They zero out somewhat fewer low harmonics but have a major advantage of needing only **one-half the storage** for amplitude data values.

Magic sinewaves are extremely exacting in their solutions. A typical equation set for a seven pulse per quadrant best efficiency **magic sinewave** might be...

$$\begin{aligned} \cos(1*p1s) - \cos(1*p1e) + \dots + \cos(1*p7s) - \cos(1*p7e) &= \text{ampl} * \pi / 4 \\ \cos(3*p1s) - \cos(3*p1e) + \dots + \cos(3*p7s) - \cos(3*p7e) &= 0 \\ \cos(5*p1s) - \cos(5*p1e) + \dots + \cos(5*p7s) - \cos(5*p7e) &= 0 \\ \cos(7*p1s) - \cos(7*p1e) + \dots + \cos(7*p7s) - \cos(7*p7e) &= 0 \\ \cos(9*p1s) - \cos(9*p1e) + \dots + \cos(9*p7s) - \cos(9*p7e) &= 0 \\ \cos(11*p1s) - \cos(11*p1e) + \dots + \cos(11*p7s) - \cos(11*p7e) &= 0 \\ \cos(13*p1s) - \cos(13*p1e) + \dots + \cos(13*p7s) - \cos(13*p7e) &= 0 \\ \cos(15*p1s) - \cos(15*p1e) + \dots + \cos(15*p7s) - \cos(15*p7e) &= 0 \\ \cos(17*p1s) - \cos(17*p1e) + \dots + \cos(17*p7s) - \cos(17*p7e) &= 0 \\ \cos(19*p1s) - \cos(19*p1e) + \dots + \cos(19*p7s) - \cos(19*p7e) &= 0 \\ \cos(21*p1s) - \cos(21*p1e) + \dots + \cos(21*p7s) - \cos(21*p7e) &= 0 \\ \cos(23*p1s) - \cos(23*p1e) + \dots + \cos(23*p7s) - \cos(23*p7e) &= 0 \\ \cos(25*p1s) - \cos(25*p1e) + \dots + \cos(25*p7s) - \cos(25*p7e) &= 0 \\ \cos(27*p1s) - \cos(27*p1e) + \dots + \cos(27*p7s) - \cos(27*p7e) &= 0 \end{aligned}$$

Power polynomials of this complexity are unlikely to have a direct solution. Instead, **Newton's Method**, otherwise known as "**shake the box**" has proven to be an effective solution route. In which a **good guess** is made based on a previously useful result or a nearby amplitude. This is followed by one or more iterations of **improvement** to the good guess.

In this **GuruGram** update, we will describe some new expansions and upgrades to our earlier magic sinewave calculators whose details were originally shown to you [here](#). We have kept the same filename of **DEMO28A.ASP** for the latest ready-to-use version of this calculator.

Our main addition was to expand coverage to some fifty plus **Magic Sinewaves** whose pulse per quadrant counts are **23** or less. This should take care of nearly all sizes and values of primary engineering interest. In addition, some new debug tools are now included and future expandability to higher "n" values has been eased somewhat.

Coincidentally, the latest versions of JavaScript found in **FireFox 3** provide an additional significant speedup. All but the largest "n" values now calculate in less than one second.

The Algorithm

Details on the original ultra fast calculators appeared [here](#). Reviewing this file can ease understanding of what follows. The new calculator is found [here](#).

It is still an open question whether one-step fully deterministic solutions to magic sinewaves can be found. But the evidence keeps accumulating that approaches involving converging approximations may in fact be the **only** solution route.

The present algorithm makes a guess as to suitable angle values for a desired amplitude and zeroed harmonics of our above or similar equations. A "good guess" can be based on a previous nearby solution or directly from plots of angles versus amplitude. As detailed [here](#). That guess can be then improved by use of a powerful but underappreciated **trig identity**...

$$\cos(a + x) = \cos(a) \cos(x) - \sin(a) \sin(x)$$

This identity is true for all values of **a** and **x**. Useful simplifications can result if we are in the first quadrant and if **a** is much larger than **x**. If **x** is very nearly zero, its cosine will be close to one and its **radian** value will nearly equal its argument.

Which simplifies to...

$$\cos(a+x) \text{ can be approximated by } \cos(a) - x \sin(a) \text{ if } a \gg x$$

This expression **exactly** matches that used by **Newton's Method**! Where you make a better approximation to a solution by multiplying its present error by the **slope** of the function and add this to the present value.

Note that the slope of the cosine is minus the sine. And also that the slope of **cos(nx)** is **-n * sin(nx)**.

In the case of our above **Magic Sinewave** equations, "a" will be our previous angle guess and a known value. While "x" will be our remaining error between where we are and where we want to be.

The equations can then be rewritten with all the unknowns on the left and all the knowns on the right. Leaving us with a new set of **$n \times n$** linear equations. Such high order linear equations are fairly easy to solve using **Gauss-Jordan Reduction** per **this tutorial**.

Gauss-Jordan reduction works by dinking around with the equations ahead of time in such a manner to leave your solution obvious by inspection. This gets done by a series of normalizations and subtractions. The process is simple but highly repetitive. The sort of things computers do well.

The equation solution will then give you new angles to work with that are usually closer to your final goal of a set amplitude and zero low harmonics. Typically, four or five trips may be needed to get desired results accurate beyond ten decimal digits. The process usually converges quite rapidly.

A Rude Surprise

As I started expanding our original calculator to higher "n" values, some big time blowup problems were encountered above **$n = 19$** . It turns out that coefficients in **Gauss-Jordan Solution** can easily get out of hand. To the point where they can cause serious errors with 64-bit math.

Fortunately, **reversing the equation row sequence from high to low angles** and using a row sequence of **amplitude then alternating high-low harmonics** can dramatically reduce coefficient buildups. In some cases by four or more decades!

This reordering was all that was needed for useful solutions beyond **$n=23$** . In anticipating further problems with higher "n" values, the row sequence was made array programmable for further buildup optimization.

Further Work

On the **improved calculator** versions, a very few amplitude values near 1.00 may not converge for very high "n" values. The easiest way to deal with these is to "sneak up" on them. Ferinstance, get a valid solution to 0.97. Then change the amplitude to 0.98 and use **Improve** repeatedly instead of **Set Amplitude**. Use the 0.98 results to get 0.99 and the 0.99 to finally get 1.00.

Several Delta Friendly solutions for higher "n" have yet to be added. These are presently hand coded and remain difficult and time consuming to do. It is also not yet clear whether additional Delta Friendly solutions beyond 3 , 7, 11, 15, ... exist and are useful. The trivial **$n=1$** case appears identical to a single pulse best efficiency solution.

Several new debugging tools have been added to **the improved calculator**. There is a redefinable general debugger along with three "progress monitors" of the Gauss-Jordan solution process. These may prove useful when developing higher "n" solutions.

Adding a **squared approximation term** to our trig identity should in theory converge much faster. But this seems to convert the linear nxn equations into an ungodly mess of multiple high order quadratics. With no obvious parametric substitution. And incidentally giving credence to a deterministic solution in fact not being possible.

For More Help

Obvious next steps are expanding **the calculators** for other types and orders of magic sinewaves. Of particular interest should be suppressing **1000** or more zeros. Which now should be possible with the dramatic speedups. Also of interest is finding whether in fact a fully deterministic solution can be found. Or an accurate interpolation scheme.

These further explorations require your participation as a Synergetics **partner** or **associate**.

To proceed, view the many **Magic Sinewave** tutorial files and JavaScript calculators you'll find at <http://www.tinaja.com/magsn01.asp>.

Full **consulting services**, custom designs, **seminars**, and workshops are available. Or you can email don@tinaja.com. Or call **(928) 428-4073**.