Magic Sinewave Pulse Visualizations

Don Lancaster
Synergetics, Box 809, Thatcher, AZ 85552
copyright c2003 as GuruGram #24
http://www.tinaja.com
don@tinaja.com
(928) 428-4073

A new class of **math functions** called **Magic Sinewaves** lets you efficiently produce power sinewaves that can have **any** chosen number of low harmonics forced very near **zero**. And do so using the fewest possible switching events for the highest possible energy efficiency. Two new intros appear **here** and **here**, along with a development proposal **here**, a tutorial **here**, and calculators **here**, and seminars and workshops **here**.

In this **Gurugram**, we'll take a closer look at how the magic sinewave pulses vary with position and amplitude. These visualizations can give us some new insight on **Magic Sinewave** synthesis. With reasonable care, we can find that...

8-bit storage words are adequate for most magic sinewaves.

and...

As few as FIVE bytes of storage are needed per amplitude.

It turns out that the quantization options in our **Magic Sinewave Calculators** are extremely conservative as they were based upon a 20 degree "full scale" pulse or delay width. By viewing the actual needed pulses and applying some **Smoke and Mirror** techniques, considerably better performance can sometimes result.

The two most useful magic sinewaves I've found so far are the **Best Efficiency 28** and the **Delta Friendly 28**. Let's see what their pulse plots look like...

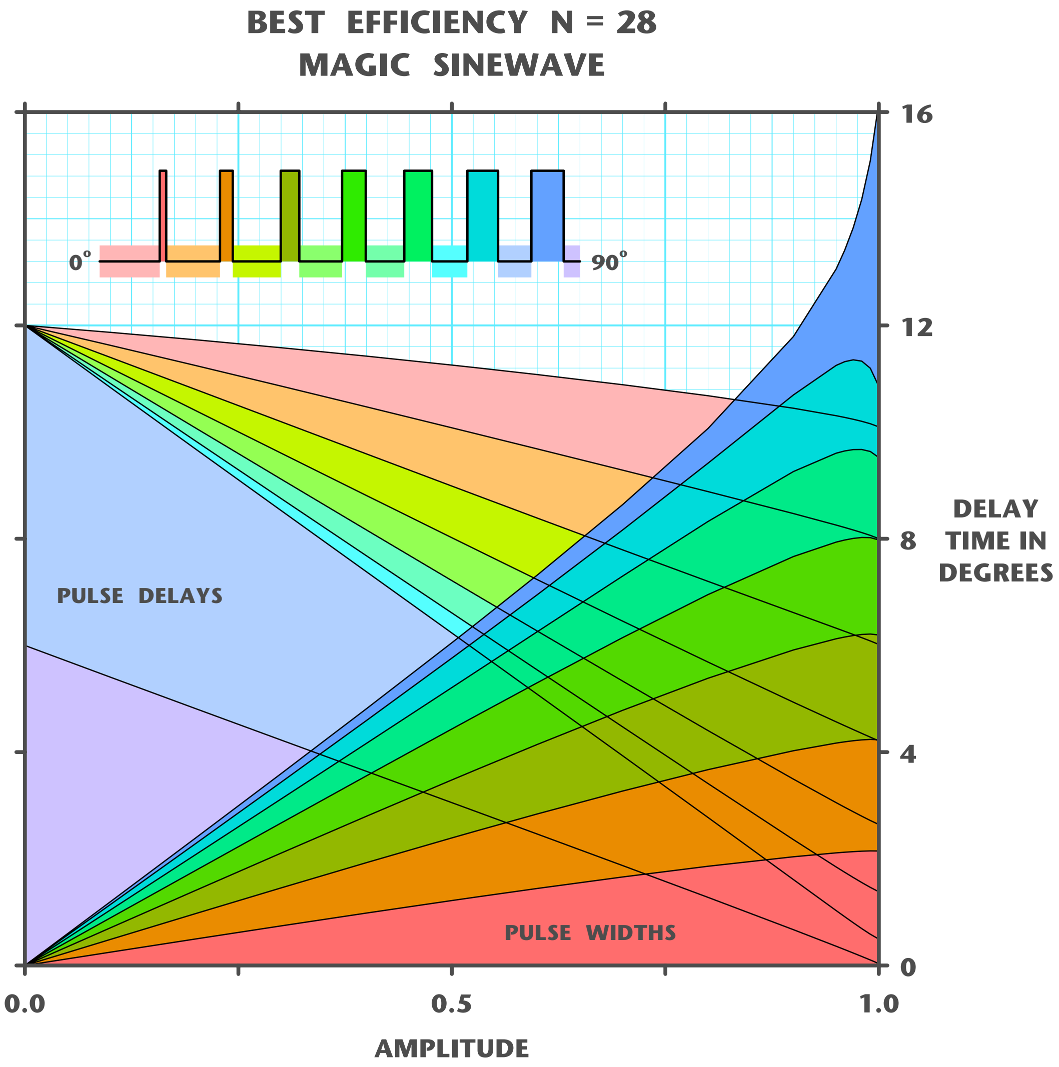
Best Efficiency 28

The best efficiency 28 magic sinewave has 28 pulses per cycle or 7 pulses per quadrant. It zeros out the first 28 harmonics but is suitable only for single phase use. Here is a plot of how its first quadrant pulses change their position and width versus amplitude...

The pulse widths and positions are gotten from the calculators by subtracting their starting and ending angles. Our plot shows the interpulse delays to the **left** and the pulse widths to the **right**. Red values are **early** while blue values are **late**.

We see that the delay values are somewhat linear. Some values are possibly directly generated rather than using table lookup. Delay values can be built as a deviation from linearity or as the **sum** of all lesser delays. Either method should substantially improve table lookup accuracy. To well beyond what is needed.

Note that the pulses are well behaved up to amplitude 0.95 and a little strange above that. Which might tempt us to not use the highest possible amplitudes.



We can quickly tell that high amplitudes of **p7** will be the most critical. Owing to the "double whammy" of being both very large and very bent. You can use the **Magic Sinewave Calculators** to find how much worse the distortion gets with a pulse position or size error. Entering half a bit at 8-bit full scale 16 degree accuracy tells us that such a **p7** error here will only increase total distortion by a fraction of a percent.

But even this acceptably small error can be substantially reduced by generating **p7** either as a **deviation from linearity** or as the **sum of lesser pulse widths**.

Thus, use of 8-bit table lookup should work just fine here. And that at least a few of the table lookup delay values can possibly be directly calculated.

Delta Friendly 28

The **delta friendly 28** magic sinewave is fully three phase compatible and needs far fewer table lookup storage values. But it only zeros out the first 22 harmonics, compared to the 28 that got zeroed by the **best efficiency 28**. Its visualization plot appears on the next page.

Understanding how to get from the **Magic Sinewave Calculator** pulse edges to three phase port patterns can be tricky. A good way is to start with the **wrap map** of our **Delta Friendly Magic Sinewave** intro tutorial and then going on to these **special phasing plots**.

In general, any 30 degree interval has a repeating pattern of three delay values and five pulse width values. This pattern mirrors over a 60 degree interval and then repeats six times for a full cycle. In each interval, pulse presence and polarity is suitably routed to each needed port line. For instance, in the 0-30 interval, **p1** might be sent to phase **a** with positive polarity, **p5** goes to phase **b** with positive polarity, and **p7** goes to phase **c** with negative polarity.

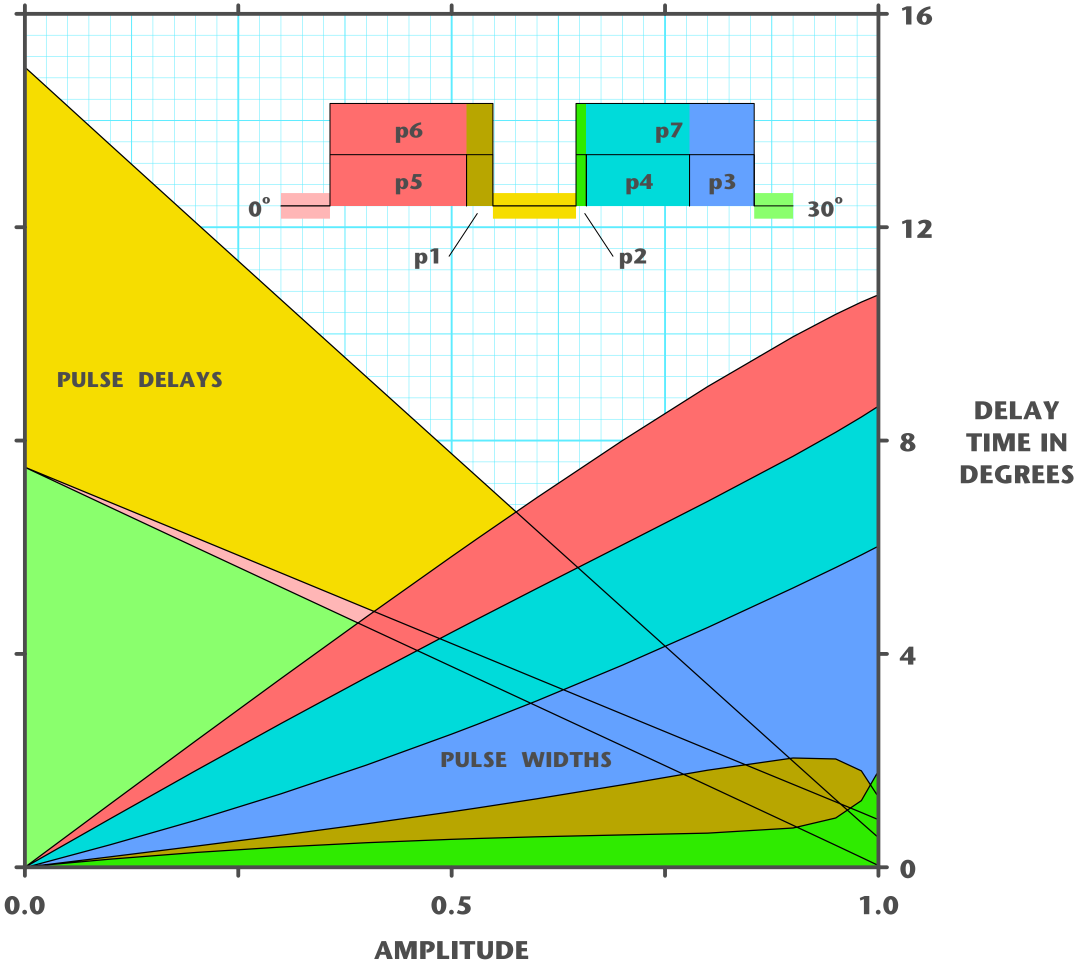
We can see that the plotted three delay values are very nearly linear. And thus can possibly be eliminated as table lookup storage values. The slight nonlinearity that remains can still cause moderately worse distortion, so it is best eliminated by adjustment or by recalculating optimal Magic Sinewave positions. Adding forced true linearity as a constraint.

Repeating ourselves...

As few as FIVE bytes of storage are needed per amplitude!

Except for that disconcerting "width crossover" of p1 and p2 at high amplitudes, the pulses seem well behaved and easily synthesized. Their accuracy is further improvable by again using deviation from linearity or smaller pulse summing techniques. And, again, 8-bit table lookup data used with care should suffice.

DELTA FRIENDLY N = 28
MAGIC SINEWAVE



What clock Frequency?

The minimum clock frequency needed for magic sinewaves may seem surprisingly high at first glance. This comes from the need to accurately resolve small portions of each cycle. A quick answer is "**Use 4 Megahertz**".

Consider a 60 Hertz magic sinewave. Its time period will be 16.667 milliseconds. A two degree interval of the sinewave will be 92.5 microseconds. Chopping this up to eight bit (256 level) resolution will give you a timing interval of 0.36 microseconds, or somewhat under four Megahertz. Details will, of course, vary with your required generation accuracy and the number of clock cycles needed per instruction on your chosen microcontroller.

Since the microcontroller is basically generating much longer time delays, any needed setup or overhead can usually be included **within** those delays.

Naturally, the **highest feasible clock frequency** may give advantages in coding flexibility and data access. Microcontroller power needs are seldom of concern in most power sinewave applications. A reminder that trying to generate both a magic sinewave and its reference frequency at the same time may need excessive clock frequencies. It is thus best to use one system to generate a Megahertz region variable reference clock and a second for actual sinewave synthesis.

Lower clock frequencies can be gotten by allowing higher distortion levels. Note that the calculated distortion values are **before** the lowpass filtering needed to reject the first two uncontrolled harmonics. If the low harmonic distortions are more or less acceptable, chances are the higher ones will be filtered out almost completely.

Transient Considerations

Sudden changes in a magic sinewave synthesizer can create **transients** that may have undesirable effects on the load being driven. The details are highly system specific. But two first principles are...

Slew rate limit both amplitude and frequency changes.

And...

Avoid unneeded transients by synchronous switching.

While it is safest to change frequency or amplitude only on the positive zero crossing of each cycle, faster response may be gained by switching more often. The key rules are **switch zero to zero** or, if really needed, **switch one to one**.

On a delta-28 magic sinewave, it is feasible to zero switch every 15 degrees or to zero and one switch approximately every 7.5 degrees. Your average switching latency times would end up as a few degrees or several hundred microseconds.

For More Help

Additional **Magic Sinewave** help is available per the previously shown web links. Detailed analysis of system specifics are available at our standard consulting rates. Per our **Magic Sinewave Development Proposal** and our **Consulting Services**.

Additional **GuruGrams** columns await your ongoing support as a **Synergetics Partner**.