

Magnitude Scaling for Increased SFDR in DDFS

Petter Källström and Oscar Gustafsson

Linköping University Post Print

N.B.: When citing this work, cite the original article.

©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Petter Källström and Oscar Gustafsson, Magnitude Scaling for Increased SFDR in DDFS, 2011, 29th Norchip Conference, Lund, Sweden, 14-15 November 2011.

<http://dx.doi.org/>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-72128>

Magnitude Scaling for Increased SFDR in DDFS

Petter Källström and Oscar Gustafsson
 Department of Electrical Engineering
 Linköping University
 SE-581 83 Linköping, Sweden
 Email: {petterk, oscarg}@isy.liu.se

Abstract—When generating a sine table to be used in, e.g., frequency synthesis circuits, a widely used way to assign the table content is to simply take a sine wave with the desired amplitude and quantize it using rounding. This results in uncontrolled rounding of up to 0.5 LSB, causing some noise. In this paper we present a method for increasing the signal quality, simply by adjust the amplitude within a ± 0.5 range from the intended. This will not affect the maximum value of the sinusoid, but can increase the spurious free dynamic range with some dB.

I. INTRODUCTION

A direct digital frequency synthesis (DDFS) is used to generate sinusoids with high spurious free dynamic range (SFDR) and good frequency control. The simplest version of a DDFS consists of a phase accumulator and a look-up-table (LUT) containing the sine table. In order to save ROM area, the sine symmetry can be used to store only one quarter of the table.

Figure 1 illustrates an W bits DDFS, quantized to D bits output resolution (including the sign bit), and an L bits phase accumulator, where $L \geq W$. The architecture uses $L - W$ bits phase truncation (PT) between signal s_1 and s_2 , where the $L - W$ least significant bits (LSBs) have been discarded. The data on the signals s_1 to s_8 are illustrated in the graphs both by the specification $W = 4$, $D = 3$ (the square curves), and without truncation/quantization (the thinner curves).

Quadrants 1 to 4 are marked Q1 to Q4 over graph s_3 . The signals are as follows:

- s_1 The phase accumulator values.
- s_2 The phase after truncation of the $L - W$ (LSBs).
- s_3 The truncated phase when the two most significant bits (MSBs) have been removed.
- s_4 The second most significant bit (MSB) of the phase, indicating Q2 and Q4.
- s_5 The phase, when Q2 and Q4 have been “mirrored”.
- s_6 The output of the look up table (LUT).
- s_7 The most significant bit of the phase, indicating Q3 and Q4.
- s_8 The phase, when Q3 and Q4 have been inverted.

Each clock cycle a frequency control word (FCW) is added to the phase accumulator, modulo 2^L . When $FCW = 1$, the accumulator will make one rotation in 2^L clock cycles.

If $FCW > 1$, the accumulator will finish exactly FCW rotations per 2^L clock cycles, so the frequency resolution, f_{res} ,

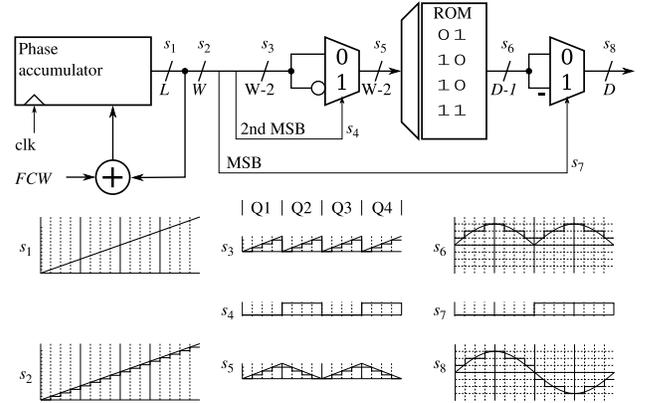


Fig. 1. DDFS block schematic with signal indications for the signals s_1 to s_8 .

and the output frequency, f_o , will be

$$f_{res} = \frac{f_{clk}}{2^L},$$

$$f_o = FCW \cdot f_{res}.$$

The ROM coefficients are typically calculated using a 0.5 LSB phase offset, which make the sine symmetry more efficiently implemented. This is well described by, e.g., Vankka *et al.* [1]. This can be seen as a phase rounding toward the closest “.5”, e.g., phase 3.125 is rounded to 3.5 rather than truncated to 3.0, as the truncation would mean.

The phase truncation will give alias problems, well described by, e.g., Ashrafi *et al* [2]. Those are related only to the input wordlength, W , and weakly affected by the accumulator size L , according to the relation [3, eq. (10)]

$$SFDR_{alias} = 20 \log_{10} \left(\frac{\sin \frac{\pi(2^W - 1)}{2^L}}{\sin \frac{\pi}{2^L}} \right) \text{ dB.} \quad (1)$$

The phase truncation has a signal and DFT response effect illustrated with an example in Fig. 2, where $W = 5$ and the output quantization has been omitted.

The output quantization gives an error that, in some sense, is close to random, and so gives a noise spread over the entire spectrum of odd harmonics. Figure 3 illustrates the noise with 4 output bits ($D = 4$) and many input bits ($W = 10$, no phase truncation). The phase is given in radians, because the phase resolution is not very relevant here. As can be seen, there

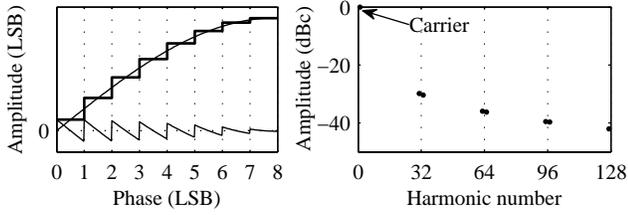


Fig. 2. The phase truncation, applied on a signal and its frequency response (only the first quadrant of the sinusoid is shown). Here $W = 5$ and quantization is omitted (so the amplitude scale is irrelevant). Six bits truncation shows the main aliases around the 2^W harmonic.

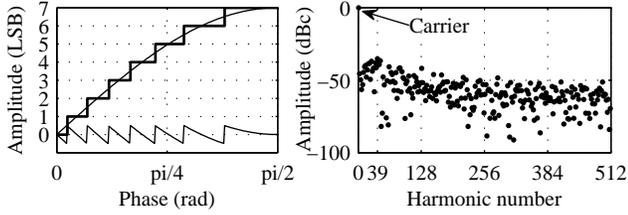


Fig. 3. Quantization of a signal and its frequency response. Here with $D = 4$ bits of amplitude resolution (excl. sign bit) are used. $W = 10$ phase bits are used to give a good phase resolution. The highest spur is clearly visible as the 39:th harmonic.

is a spur peak around harmonic $h_{spur} = 39$. The sinusoid corresponding to $h_{spur} = 39$ and the error are depicted in Fig. 4. A spur at harmonic h_{spur} will make h_{spur} rotations per 2^L samples, and so the sine component will have a period of $2\pi/h_{spur}$ rad (where 2π rad corresponds to phase 2^L).

This peak is caused by the triangular error shape, with a period starting at $1/(2^{D-1}-1)$ radians, and slowly increasing. The peak will, more general, have a harmonic number $h_{spur} \lesssim 2\pi \cdot (2^{D-1}-1)$.

Figure 5 illustrates a combination of phase truncation and output quantization, in a good balanced relation between $W = 5$ and $D = 4$; it differs ≈ 1 dB between the biggest quantization noise spur and the alias spur.

The phase to sine amplitude converter (PSAC) can be implemented in many ways. In this paper we are considering ROM based methods, where each phase can be controlled individually (in difference from, e.g., polynomial approximation algorithms). Those methods includes pure look up tables (with or without memory compression) [4], sum of bit products [5] or thermometer coded implementations [6].

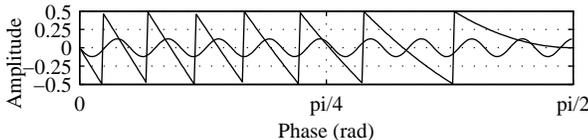


Fig. 4. The quantization error from Fig. 3, and the greatest spur sinusoid, caused by the error. Again $D = 4$ and $W = 10$. The spur has harmonic number 39.

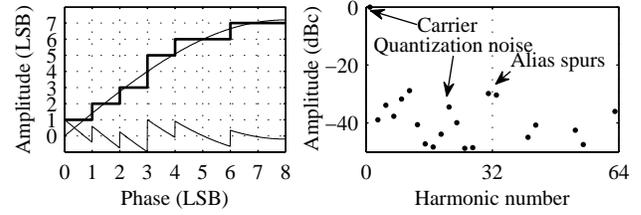


Fig. 5. A signal and its frequency response, where $W = 5, D = 4$ bits of phase and amplitude respectively, illustrates the noise sources. The intended signal and the errors are also shown. The quantization error causes a floor of spurs, while the phase truncation causes aliases around the $2^W \cdot f_{res}$ frequencies.

A. Input vs Output Resolution

When $W \ll D$, the aliases spurs are dominating, and so the quantization has no impact at all on the SFDR. This “range” in the (D, W) design plane is denoted PT range in this paper (because the phase truncation causes the aliases).

When $D \ll W$, the amplitude quantization (AQ) is the limiting factor on the signal, so the SFDR can be optimized with the proposed scaling algorithm. This range is here denoted the AQ range.

The selected ratio between D and W depends on the application, but typically $D \approx 0.75(W - 1)$ gives a good balance, as supported by the results. A greater D (lower quantization noise) can be motivated if a low pass filter can be used to suppress the alias spurs, or if the total noise should be minimized. A greater phase resolution (W) will not increase the SFDR very much (it may even decrease, as will be shown).

B. The SFDR Measurement

The spurious free dynamic range (SFDR) is a measure of how much “louder” the carrier is than the highest spur for a sinusoid. The SFDR for DDFS’ is typically measured with an odd FCW during 2^L samples, because this will “test” all 2^L phases. The spurs will be rearranged, but not “collide” with each others when comparing different (odd) FCW s. Typical, the measurement is performed with $FCW = 1$, which gives the same SFDR as for any odd FCW , as supported by Torosyan *et al.* [3].

The SFDR of a certain integer vector \bar{s} for the first quadrant, using $L - W$ bits phase truncation, is calculated in the following steps:

- 1) Expand \bar{s} to the entire rotate from the first quarter.
- 2) Duplicate each point 2^{L-W} times.
- 3) Perform an FFT and keep first half of the vector.
- 4) Find the amplitudes for the carrier, c , and biggest noise spur, n .
- 5) $SFDR = 20 \log_{10}(\frac{c}{n})$

II. PROPOSED SCALING ALGORITHM

If the sine table, in position i , has the value $ampl \cdot \sin(\phi_i)$, the biggest value in the memory will be $\lceil ampl \rceil$ ($ampl$ rounded toward closest integer). This is typically desired to

be = $2^{D-1} - 1$, which implies that $ampl = 2^{D-1} - 1 + a_s$, where a_s , $-0.5 \lesssim a_s \leq 0.5$, denotes a “sub amplitude”.

The proposed algorithm starts at the smallest allowed sub amplitude (≈ -0.5), and gradually increase it up to 0.5. For each step, the SFDR is analyzed, and the best result is selected. The step size is set so that exactly one value in the ROM is affected.

The algorithm is presented in a pseudo code format in Algorithm 1.

Algorithm 1 The scale algorithm. δ is a small number, used to avoid problems caused by floating point roundings.

```

 $ampl \leftarrow 2^{D-1} - 1$ ; // set the amplitude.
 $a_s \leftarrow \frac{ampl - 0.5}{\cos(\pi/2^W)} - ampl + \delta$ ;
// initialize the sub amplitude to the smallest allowed value.
 $\bar{\phi} \leftarrow ((1, \dots, 2^{W-2}) - 0.5) \frac{\pi}{2^{W-1}}$ ;
// a vector with all phases in the first quadrant.
repeat
   $\bar{s} \leftarrow (ampl + a_s) \cdot \sin(\bar{\phi})$ ; // table with the sinusoid.
  Analyze the SFDR for  $[\bar{s}]$ ;
   $\bar{inc} \leftarrow [\bar{s}] + 0.5 - \bar{s}$ ; // calculate how much each point can increase
  before it affects the rounding.
   $\bar{rinc} \leftarrow \bar{inc} / \bar{s}$ ; // same, but relative the amplitude.
   $minc \leftarrow \min(\bar{rinc})$ ; // the smallest change that affects one integer.
   $a_s \leftarrow a_s + (ampl + a_s) \cdot minc + \delta$ ; // Update the sub amplitude.
until  $a_s \geq 0.5$ 
Select the best SFDR found and the  $a_s$  that generated it.

```

A. Time complexity

During the entire amplitude scan, the amplitude is increased with one. The unrounded value in phase ϕ will increase $s \approx \sin(\phi)$ during the scan, and so the probability is roughly s that this phase will change value in the table during the scan. There is 2^{W-2} values in the table, and a ratio $2/\pi$ of them will change. Only one will change per iteration, so in average this will require $2^{W-1}/\pi$ iterations.

Figure 6 compares the real number of iterations (markers) with the approximation $2^{W-1}/\pi$ (line).

The carrier amplitude will differ with different amplitudes, this is however typically a very small change on the dB scale. With, e.g., $D = 7$, the amplitude will change less than $\pm 0.8\%$, which corresponds to ≈ 0.07 dB, or ≈ 0.004 dB when $D = 11$.

III. RESULTS

In all analysis presented here, five bits of phase truncation have been used, so $L = W + 5$ in (1), gives $SFDR_{alias} \approx 20 \log_{10}(2^W - 1) \approx 6W$ dB, as a higher bound on the SFDR from the alias spurs. In the AQ range ($W \gg D$), the SFDR can be increased up to allmost 3 dB by changing the sub amplitude (without any cost in the implementation hardware).

Figure 7 illustrates briefly, on scales from 0 to 3 dB, how much SFDR that is possible to gain for different W 's and D 's

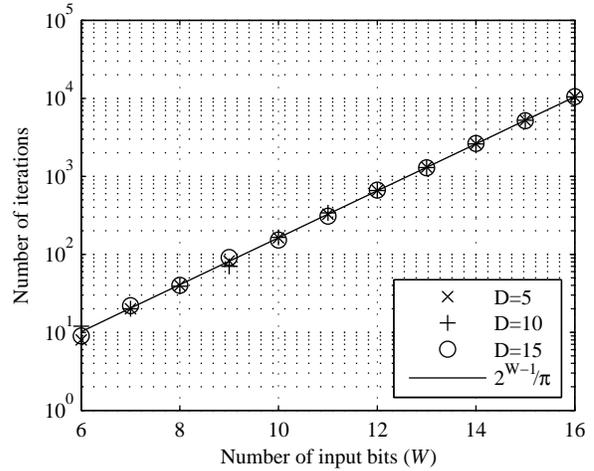


Fig. 6. Number of iterations as a function of W .

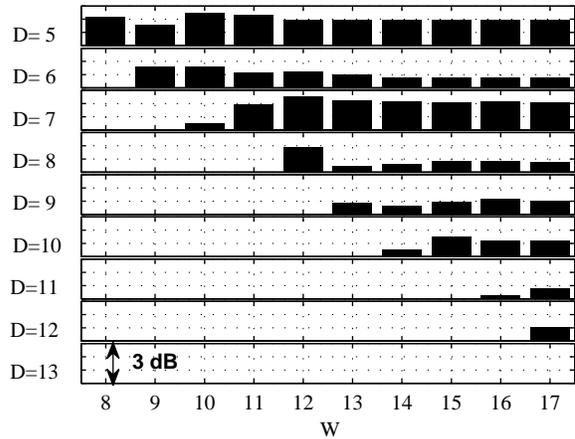


Fig. 7. The SFDR gain from scaling, for different W and D . The dB axes are 3 dB high.

by scaling the DDFS. The plot clearly shows the difference between the PT range in left bottom half, and the AQ range in the top right half.

Two examples of this gain is illustrated in Fig. 8 where the SFDR is plotted as a function of the sub amplitude, with $W = 12$.

Figures 7 and 8 also depict that with a $D \gtrsim 0.75(W - 1)$ the scaling will have no effect at all.

The frequency response has, as mentioned, a peak just below harmonic $h_{spur} \approx 2\pi \cdot (2^{D-1} - 1) \approx 396$ when $D = 7$. In Fig. 9 this is illustrated using $W = 12$ and five bits phase truncation, both before and after the scaling. There is a clear peak at harmonic 393, which has spread out slightly to the neighboring harmonics when the amplitude is scaled from 63 to 63.2423. 63.2423 is the amplitude that will give the length of the triangular wave (Fig. 3) the largest distortion (so two neighboring teeth will probably have different length).

Table I illustrates the SFDR before and after scaling, as well as the gained SFDR. In the PT range (left bottom corner), the

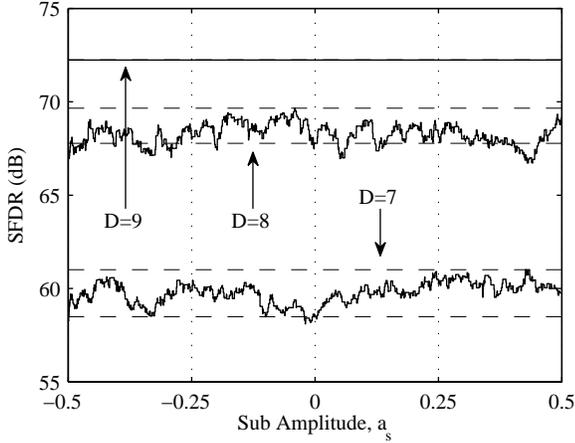


Fig. 8. The SFDR as a function of the sub amplitude for $W = 12$, $L = 17$ and $D = 7, 8, 9$. For each curve, the maximum and ($a_s = 0$) SFDR level is plotted. With five bits phase truncation the upper bound $SFDR_{alias} = 72.23$ dB, which causes the entire $D = 9$ plot to be constant and independent of the scaling.

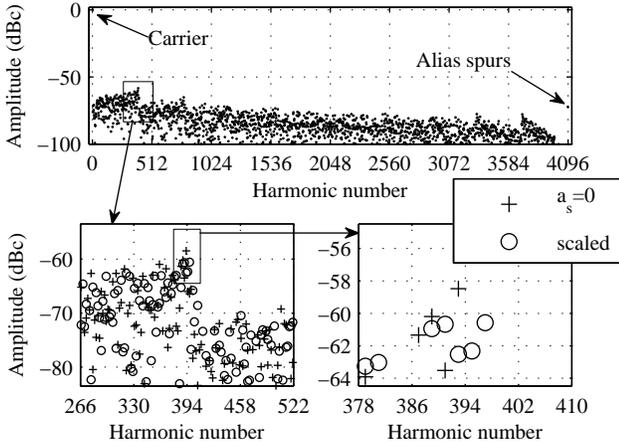


Fig. 9. The frequency response for $W = 12$, $L = 17$ and $D = 7$, showing the characteristic peak of spurs, both default ($a_s = 0$) and optimized amplitude ($a_s = 0.2423$ in this case).

SFDR is affected only by the number of input bits, W . In the AQ range (top right corner), the SFDR is mainly affected by the output resolution, D , and is therefore possible to increase with scaling.

One effect that is clear in table I is that the SFDR is increased with ≈ 16 dB per two output bits (8 dB per bit), in the AQ range. The noise floor is decreased by ≈ 6 dB per bit, so the SFDR should increase with 6 dB per bit, if it was not for the spur peak, illustrated in Fig. 9. The peak seems to grow as the D decreases, with roughly 2 dB per bit.

Because the SFDR grows with 8 dB per output bit (D) in the AQ range, and with 6 dB per input bit (W) in the PT range, the balanced cut between the ranges is placed around the line $W = D \cdot \frac{4}{3} + 1$. A designer of a DDFS should mainly be interested in this range, \pm a few bits.

TABLE I
THE SFDR ACHIEVED BEFORE AND AFTER THE SCALING, ON THE FORM "before+gain=after" (dB).

Output bit width (D)	Input bit width (W)					
	6	8	10	12	14	16
5	35.97	42.23	42.63	42.94	42.92	42.91
	+0.00	+2.19	+2.47	+1.93	+1.92	+1.91
	=35.97	=44.42	=45.10	=44.87	=44.84	=44.81
7	35.97	48.12	59.70	58.48	58.73	58.76
	+0.00	+0.00	+0.48	+2.52	+2.14	+2.10
	=35.97	=48.12	=60.18	=61.00	=60.87	=60.86
9	35.97	48.12	60.18	72.23	76.31	75.85
	+0.00	+0.00	+0.00	+0.00	+0.66	+1.20
	=35.97	=48.12	=60.18	=72.23	=76.97	=77.05
11	35.97	48.12	60.18	72.23	84.27	92.75
	+0.00	+0.00	+0.00	+0.00	+0.00	+0.28
	=35.97	=48.12	=60.18	=72.23	=84.27	=93.03
13	35.97	48.12	60.18	72.23	84.27	96.32
	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00
	=35.97	=48.12	=60.18	=72.23	=84.27	=96.32

Note that the total noise will be reduced when D grows, even when the aliases are the dominating spurs, so the SNR can be increased slightly.

One other effect that can be seen in Table I is that the SFDR might be reduced when W increases, in some special cases. When, for instance, $D = 7$, $W : 10 \rightarrow 12$, the (unscaled) SFDR will shrink from 59.7 to 58.48. This may be caused by the fact that a lower W will introduce some noise to the length of the triangular waves, so the amplitude from the top spur may be divided into neighboring spurs, in a similar way as the scaling acts.

IV. CONCLUSIONS

A method to increase the SFDR for a LUT based DDFS without any hardware changes is proposed in this paper. The method is based on a small scaling of the amplitude, in order to archive the best SFDR. The SFDR can in this way be increased with more than 2 dB in some cases. In the case that the phase to sine amplitude converter has 12 input bits and 7 output bits (incl. sign bit), the SFDR can be increased from 58.48 to 61 dBc.

REFERENCES

- [1] J. Vankka, M. Waltari, M. Kosunen, and K. A. I. Halonen, "A direct digital synthesizer with an on-chip D/A-converter," *IEEE J. Solid-State Circuits*, vol. 33, no. 2, pp. 218–227, 1998.
- [2] A. Ashrafi and R. Adhams, "Theoretical upperbound of the spurious-free dynamic range in direct digital frequency synthesizers realized by polynomial interpolation methods," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 10, pp. 2252–2261, 2007.
- [3] A. Torosyan and A. N. Willson, "Exact analysis of DDS spurs and SNR due to phase truncation and arbitrary phase-to-amplitude errors," in *Proc. IEEE Int. Frequency Control Symp. and Exposition*, 2005.
- [4] J. M. P. Langlois and D. Al-Khalili, "Phase to sinusoid amplitude conversion techniques for direct digital frequency synthesis," *IEE Circuits, Devices, Syst.*, vol. 151, no. 6, pp. 519–528, 2004.
- [5] K. Johansson, O. Gustafsson, and L. Wanhammar, "Approximation of elementary functions using a weighted sum of bit-products," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2006, pp. 795–798.
- [6] H. C. Yeoh and K.-H. Baek, "A 4GHz direct digital frequency synthesizer utilizing a nonlinear sine-weighted DAC in 90nm CMOS," in *Proc. IEEE Asia Pacific Conf. Circuits Syst.*, 2008, pp. 1700–1703.