Reducing PWM distortion phase noise in fractional-N synthesizers

Andrew Holme

http://www.aholme.co.uk

***Abstract* — An entirely-digital linearization technique is described which reduces phase noise due to the intrinsic distortion of uniformly-sampled pulse-width modulation (PWM) at the phase detector output in fractional-N synthesizers.**

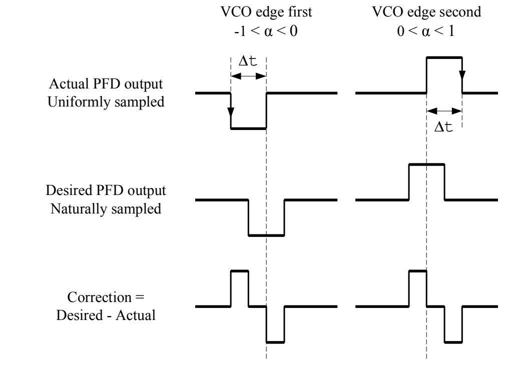
Introduction

Integer-N PLL synthesizers track with constant (zero) phase error. In contrast, the fractional-N principle involves a constantly changing output from the phase-frequency detector. The PFD output waveform is uniformly-sampled PWM (UPWM) which is inherently non-linear. Despite being concentrated at high frequencies by noise shaping, quantization noise folds down creating in-band distortion which cannot be removed by the loop filter. This adds phase noise on the synthesizer output at offsets close to the carrier. Ironically, the steeper the noise is shaped, the worse the effect. It might be noticeable if 4th or higher order noise shaping was used with a quiet VCO.

Anders Eklof [1] patented a design which converts UPWM to PAM (pulse amplitude modulation) by inserting an integrator followed by a sample-and-hold between the PFD and the loop filter. Unlike UPWM, PAM is linear in-band.

Presented here is a digital solution inspired by the work of Peter Craven [2, 3] on digital audio. A practical embodiment based on a 4th order MASH structure is described, although the principle could also be applied to sigma-delta noise shapers.

This solution can partially or substantially cancel the intrinsic distortion by converting UPWM to naturally-sampled NPWM, which is linear in-band. NPWM pulses have regularly spaced centres of gravity, whilst those of UPWM effectively introduce jitter in the time sampling points, as shown in figure 1.



Pulse width $\Delta t = \alpha \cdot \tau$ where α is PWM modulation index and τ is sampling interval.

Figure 1

Methodology

The structure of a fractional-N synthesizer constrains how and where correction can be applied. Ideally, the high-resolution of the 0.F input should be exploited [figure 2].

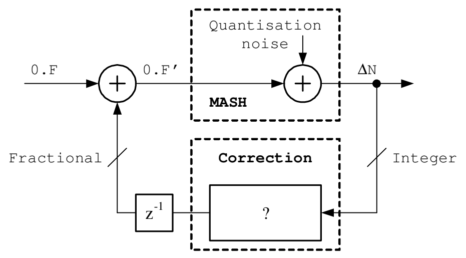


Figure 2

The input command can only change once per reference cycle, whilst the ideal correction transient is much shorter. In fact, the distortion associated with any PFD output pulse cannot be dealt with until one whole cycle later.

The required correction is approximated using a series of discrete impulses, chosen by frequency domain matching [figure 3].

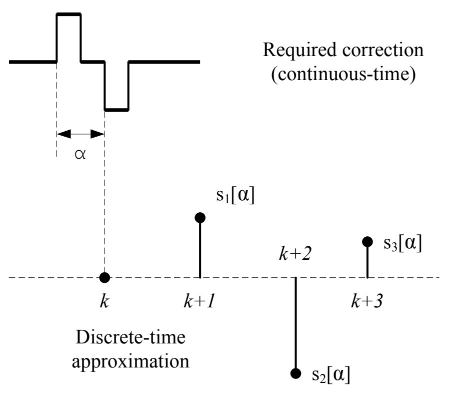


Figure 3

The Fourier transform of the impulse train is:

$$H(\omega) = \sum_{k=1}^{3} s_k e^{i\omega k} \qquad (1)$$

Normalized time and frequency are used:

$$\tau = 1$$

$$0 \le \omega \le \pi$$

The Fourier transform of the required correction, for a pulse of width α, is the difference between the transforms of uniformly and naturally sampled pulses:

$$H_N(\omega) = \frac{1}{i\omega}\left(e^{\frac{1}{2}i\omega\alpha} - e^{-\frac{1}{2}i\omega\alpha}\right) \qquad (2)$$

$$H_U(\omega) = \frac{1}{i\omega}\left(e^{i\omega\alpha} - 1\right) \qquad (3)$$

$$H(\omega) = H_N(\omega) - H_U(\omega) \qquad (4)$$

Each pulse width requires a unique set of impulses. The amplitudes are stored in a read-only-memory (ROM) addressed by a truncated α. Good results are obtainable with quite coarse truncation.

Matching is performed at 50 frequencies distributed (in a Chebyshev manner) over the range 0 to $\omega_{\max}$:

$$\omega_r = \omega_{\max} \cos\left[\left(r - \frac{1}{2}\right)\frac{\pi}{100}\right] \qquad (5)$$

The optimum impulse amplitudes $\{s_k\}_{k=1}^{3}$ are obtained by least squares fitting. The following Scilab script tabulates s_k for $-0.3 \le \alpha \le 0.3$:

```
r = [1:50]';
k = (1:3);

wmax = 2*%pi * 1e3/100e3; // 2*pi*fmax/fref
wr = wmax * cos(%pi/2 * (r-1/2)/length(r));

function e = myfun(sk, correction)
   e = exp(%i*wr*k)*sk - correction;
   e = [real(e); imag(e)];
endfunction

function sk = fit(alpha)
  natural = (exp(%i.*wr*alpha/2) - exp(-%i.*wr*alpha/2)) ./ wr / %i;
  uniform = (exp(%i.*wr*alpha) - 1) ./ wr / %i;
  correction = natural - uniform;
  [fopt, sk, gropt] = leastsq(list(myfun, correction), zeros(k'));
endfunction

for alpha = -0.3:0.01:0.3
  sk = fit(alpha);
  printf("%0.2f,", alpha);
  for i=k, printf("%0.5f,", sk(i)); end
  printf("\n");
end
```

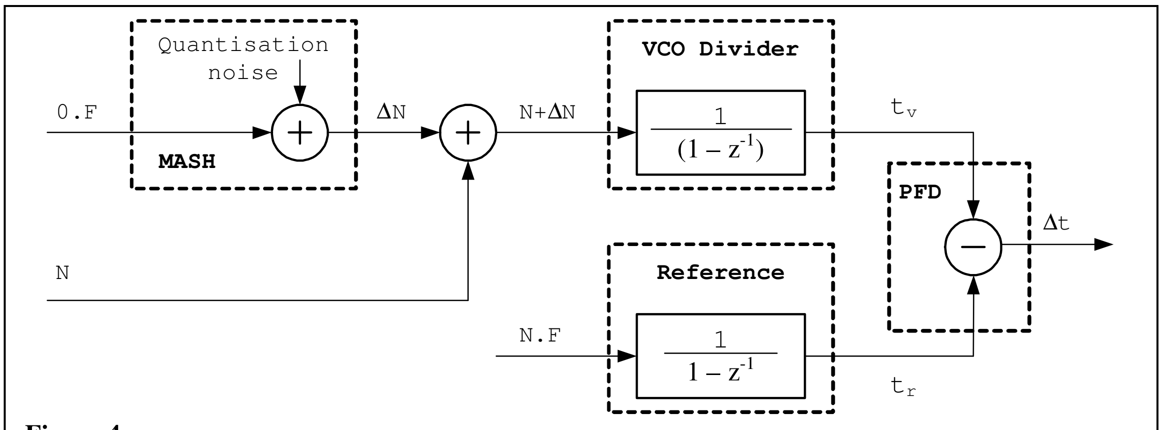


Figure 4

VCO frequency and period of a fractional-N synthesizer are related to reference period by:

$$f_{VCO} = \frac{1}{\tau_{VCO}} = \frac{N + 0.F}{\tau_{REF}} \qquad (6)$$

N+0.F (or N.F for short) is the programmed frequency command; N is the integer; and 0.F the fractional part thereof.

In the analysis which follows, edge times are related to VCO period τ_{VCO}. We assume the PLL is locked, VCO frequency is stable, and the programming command does not change. Index k and discrete time step z^{-1} denote reference cycles i.e. $z = e^{i\omega\tau_{REF}}$

The k^{th} VCO divider edge occurs at time $t_V(k)$, where $t_V(k)$ is an integral number of VCO periods. The k^{th} reference edge occurs at time $t_R(k)$. The VCO divider and the reference are modelled as integrators:

$$\frac{t_V(k)}{\tau_{VCO}} = \frac{N + \Delta N(k)}{1 - z^{-1}} \qquad (7)$$

$$\frac{t_R(k)}{\tau_{VCO}} = \frac{N + 0.F}{1 - z^{-1}} \qquad (8)$$

$$\Delta t(k) = t_V(k) - t_R(k) \qquad (9)$$

$$\frac{\Delta t(k)}{\tau_{VCO}} = \frac{\Delta N(k) - 0.F}{1 - z^{-1}} \qquad (10)$$

Equation 10 is the width of the k^{th} PFD output pulse in VCO periods.

ROM-address α = $\Delta t(k)/\tau_{REF}$

MASH

From [4] the recurrence formula for a 4th order MASH is:

$$\Delta N(k) = 0.F(k) + e_4(k)(1 - z^{-1})^4 \qquad (11)$$

$$\Delta N(k) = 0.F(k) + e_q(k) \qquad (12)$$

- Quantized output ΔN is a signed integer
- *0.F* is the unsigned fractional command
- $0 \le 0.F < 1$
- e_4 is white noise generated in the last accumulator
- e_q is shaped quantization noise rising at 80 dB/decade.

The MASH has an ideal step response: any change in *0.F* is immediately reflected at the quantized MASH output. This is a remarkable characteristic, considering that the ΔN output is only 4-bits wide, and the 0.F fractional input command may be 32-bit.

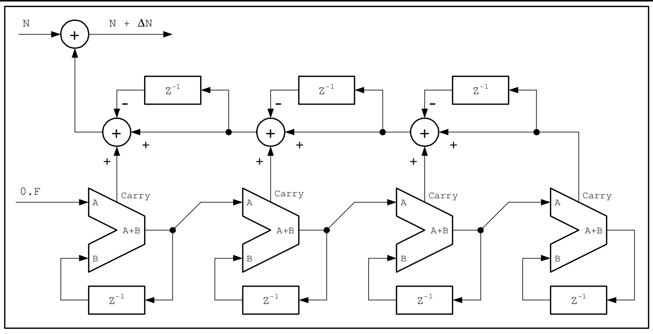


Figure 5 – 4th order MASH structure

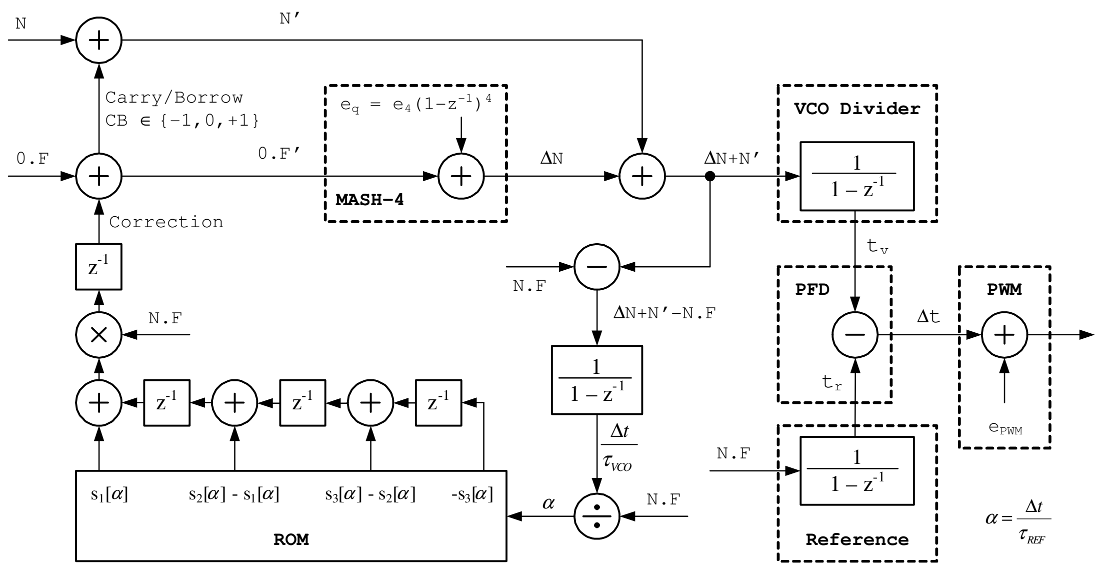


Figure 6 – Practical system

Implementation

Using the method of equation 10, a prediction of the PFD output pulse width is computed by integrating ΔN+N'-N.F; and this will only be accurate if the integrator is reset to zero when the MASH accumulators are also zero.

Corrections are summed with the fractional frequency command *0.F*, and a modified command designated *0.F'* is applied to the MASH input. Since $0 \le 0.F < 1$, and *0.F* may be close to either extreme, the correction can cause underflow or overflow. To deal with this, a carry/borrow signal is added to or subtracted from the integer command *N*.

Although *0.F'* propagates unchanged through the MASH, the VCO divider interposes an integration. To achieve corrections of *s1, s2, s3* at the PWM output, the sequence *s1, s2-s1, s3-s2, -s3* must be generated. The differentials are pre-calculated and stored in the ROM.

ROM address α and ROM outputs $\{s_k\}$ are normalized, whilst N, F and ΔN represent VCO cycles. Scaling by N.F is required between domains: division at the address input, and multiplication of the data output.

Division is expensive; however, the quotient need not be evaluated to the full precision of the dividend and divisor. Good results are still obtainable if ROM address α is truncated to as few as 6-bits. Similarly, the $\{s_k\}$ data need not be as wide as the MASH fractional input.

Simulation

The method was evaluated using a fast and accurate behavioral PLL simulator described on the author's website [5].

PLL parameters for the simulation were:

Reference frequency	f_{REF}	100 KHz
VCO frequency	f_{VCO}	2.05 MHz
Frequency command	N.F	20.5
VCO gain	k_{VCO}	2 MHz/Volt
PFD gain	k_{PD}	$5/2\pi$
Loop filter	Gain	$k = 10$
	Zero	$\omega_Z = 2\pi * 100$ Hz
	Poles	$\omega_{P0} = 0$ $\omega_{P1} = 2\pi * 1$ KHz $\omega_{P2} = 2\pi * 10$ KHz $\omega_{P3} = 2\pi * 10$ KHz

The loop filter transfer function was:

$$F(s) = \frac{k\left(1+\frac{s}{\omega_Z}\right)}{s\left(1+\frac{s}{\omega_{P1}}\right)\left(1+\frac{s}{\omega_{P2}}\right)\left(1+\frac{s}{\omega_{P3}}\right)}$$

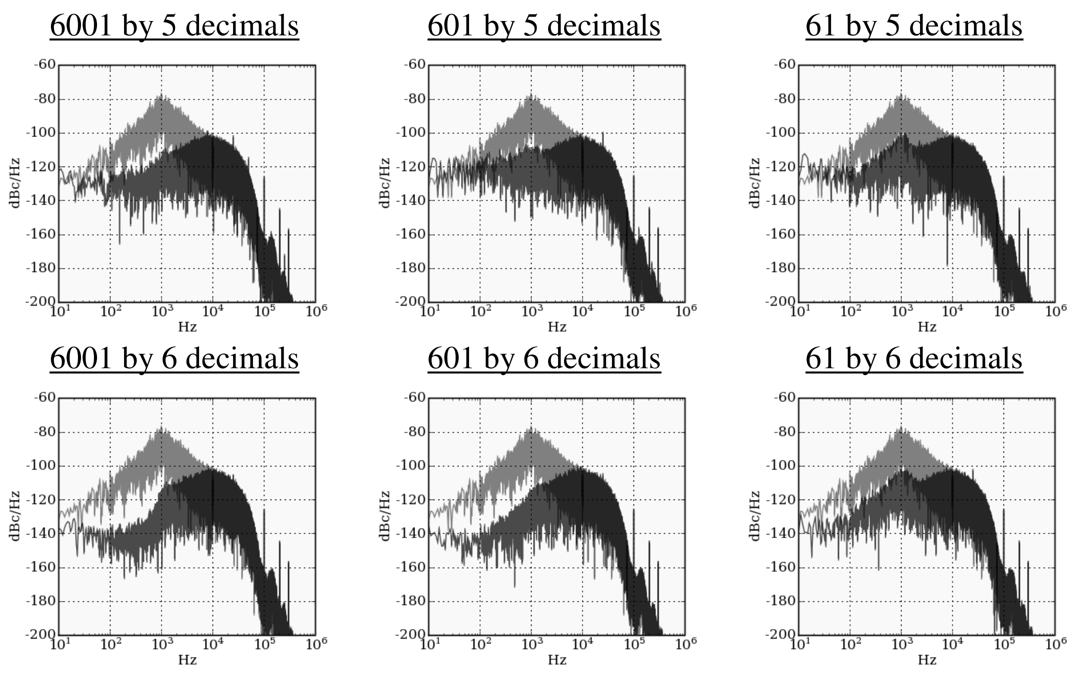


Figure 7 – Simulation results with ROM address and data truncated to different widths

All loop components, including the VCO, were simulated as perfect and noiseless. The only noise sources were: PWM distortion; MASH quantization; and the finite numerical precision of the simulator.

The combination of a 4th order MASH and a low N.F command of 20.5 were selected to maximize PWM modulation index and thereby maximize distortion. Going below N.F=20, the calculated $\{s_k\}$ exceed α and the correction loop is then potentially unstable.

Two $\{s_k\}$ truncations: to 5 and 6 decimal places; and three α truncations: to 61, 601 and 6001 levels were investigated. Figure 7 shows the resultant VCO phase noise plots.

Reference spurs and harmonics are evident at 100, 200 and 300 KHz. Fractional-spurs are also evident in some plots.

In each case, the light-gray trace is the level of uncorrected phase noise due to PWM distortion, which peaks at -80 dBc/Hz at an offset of 1 KHz from the carrier. The dark traces shows phase noise after linearization.

A phase noise reduction of 20 dB, flattening the distortion peak to the same level as the quantization noise peak at 10 KHz, is achieved by even the coarsest truncations, equivalent to 6-bits in α, and roughly 16-bits in $\{s_k\}$. ROM size is $2^6 * 16 * 4 = 4096$ bits.

Conclusion

PWM distortion is only significant when 4th or higher order noise shaping is used; and even then, the PWM-related phase noise will often be swamped by other noise sources, such as the VCO. Nevertheless, in synthesizers which contain a microprocessor, even a modest noise reduction may justify the additional overhead of only one division and one multiplication per reference cycle.

References

1. Anders Eklof , “Fractional N synthesizer with reduced fractionalization spurs,” US Patent 6914935.
2. Peter Craven, “Towards the 24-bit DAC. Novel Noise-Shaping Topologies Incorporating Correction for the Nonlinearity in a PWM Output Stage”, J. AES Vol 41, No 5 1993 May.
3. Peter Craven, “Analogue and digital convertors using pulse edge modulators with non-linearity error correction,” US Patent 5548286.
4. Andrew Holme, “15-25 MHz Fractional-N Frequency Synthesizer: Multi-stage noise-shaping (MASH)” http://www.aholme.co.uk/Frac2/Mash.htm
5. Andrew Holme, “15-25 MHz Fractional-N Frequency Synthesizer: PLL Simulation” http://www.aholme.co.uk/Frac2/Simulate.htm