

Research Article

Diophantine Frequency Synthesizer Design for Timekeeping Systems

Paul P. Sotiriadis¹ and Gregory L. Weaver²

¹ *Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD 21218, USA*

² *Space Department, Johns Hopkins University, Applied Physics Laboratory, Laurel, MD 20723, USA*

Correspondence should be addressed to Paul P. Sotiriadis, pps@ieee.org

Received 2 August 2007; Accepted 21 November 2007

Recommended by Demetrios Matsakis

Diophantine Frequency Synthesis (DFS), a number-theoretic approach to the design of very high resolution frequency synthesizers, was introduced in 2006. Further work concerning the impact of controlling mixing products for high-spectral purity was addressed and reported at the 2007 European Frequency and Time Forum. The focus of this paper is on the implementation of nested DFS architectures targeting microphase-type applications for precision timekeeping systems. We have shown that DFS does not impart any extraordinary design constraints on spectral purity in comparison to commonly used high resolution frequency synthesis techniques such as DDS or fractional N . Here we describe a design approach for 10 MHz synthesizers with $1E-13$ fractional resolution in consecutive steps ranging ± 10 Hz. The synthesizers generate their output from a 10 MHz reference standard. Such synthesizers are essential to accomplishing precision frequency correction in timekeeping systems.

Copyright © 2008 P. P. Sotiriadis and G. L. Weaver. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

In timekeeping systems, a local frequency and/or phase must be generated and maintained to a very high degree of accuracy. For example, the Time and Frequency Laboratory of the Johns Hopkins University Applied Physics Laboratory maintains UTC (APL) within ± 10 nanoseconds, based on monthly reports from the Bureau International des Poids et Mesures (BIPM). Modern timekeeping systems use phase-frequency correction (steering) through auxiliary synthesizers to maintain the accuracy of their master clocks to UTC. The frequency step resolution of synthesizers for steering timekeeping systems is typically $1 \mu\text{Hz}$ or better. Designers of these very fine resolution synthesizers must carefully consider signal purity, resolution (accuracy to the global reference), and complexity. Our paper describes the Diophantine Frequency Synthesis (DFS) design approach for very fine frequency resolution synthesizers suitable for the maintenance of autonomous clock holdover and microphase steering in laboratory timekeeping systems.

The novel DFS approach was introduced in 2006 [1, 2]. We have found that DFS alleviates the conventional trades in performance for frequency synthesizer design without sig-

nificantly taxing system complexity or resources. DFS provides high spectral purity, even in synthesizers with much less than 0.1 PPM resolution steps. In general, we make this claim in comparison with other fine resolution frequency synthesizer methods such as Direct Digital Synthesis (DDS) or fractional- N modulators which are known to present a high degree of unwanted spurious signals into the output spectrum through the fundamental process that they impart on the input reference signal [3]. The use of DDS and fractional- N synthesis design techniques has been widely adopted for timekeeping systems as high-frequency resolution (accuracy) and fast acquisition (settling time) can be achieved without the complexity of traditional multiple loop synthesizers. However, DDS and fractional- N synthesizers both cause phase perturbations in their basic operation schemes leading to coherent spurious generation [3]. In the case of DDS, accuracy to a desired frequency necessarily compromises the spectral purity of the output signal by the incidence of truncation spurious attributable to the finite size of sin/cos lookup table and the DAC [4].

DFS uses only exactly periodic signals, without employing dithering, interpolation, pulse removal, or any other approximately-periodic waveform that may corrupt the

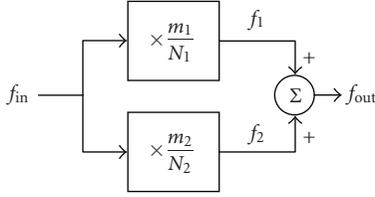


FIGURE 1: A two-PLL DFS scheme.

spectrum close to the carrier. DFS-based synthesizers present no discontinuity of the reference frequency phase, such as DDS or fractional- N , and unlike these methods, DFS does not require any special devices such as high-resolution DACs, accumulators, or sigma-delta modulators to control the spurious level of the output signal.

However, like traditional multiple loop PLL synthesizer architectures, DFS does require mixing (or multiplication) to achieve the output signal. This means that DFS synthesizers can suffer from unwanted spurious if attention to the circuit design is not adequately respected. In our 2007 EFTF paper, we described an approach for the design of VHF synthesizers with high-spectral purity of >100 dB spurious free dynamic range and showed that DFS presents no unique design-related constraints [5]. Rather, DFS design flexibility provides an advantage to achieving this level of performance in fine resolution frequency synthesis.

2. DFS—ELEMENTS OF THE THEORY

DFS is a number-theoretic approach to frequency synthesis. It is based on mathematical properties of integer numbers and linear Diophantine equations [2] (by definition, a Diophantine equation is an algebraic equation whose solutions are required to be integers [6]).

DFS results in high-level architectures using two or more Integer- N PLLs. It *distributes* the desirable output-frequency resolution among these constituent PLLs in such a manner that the resultant output fractional-frequency resolution is equal to the *product* of the constituent PLLs' fractional-frequency resolutions. Consequently, this property of DFS allows for the output frequency resolution to be made (arbitrarily) fine, that is, to have a very small frequency step, without using large prescalers or low phase-comparator frequencies in the PLL.

2.1. The abstract DFS concept

DFS considers a PLL as a multiplier of an input frequency f_{in} by a rational number m_i/N_i , as shown in Figure 1.

In Figure 1, two PLLs (i.e., two multipliers by m_1/N_1 and m_2/N_2) are driven by the same reference frequency f_{in} . Their output signals are mixed (and the mixer's output is filtered - not shown) to produce the synthesizer's output signal of frequency f_{out} which typically is $f_{out} = f_1 + f_2$, as it is here, or $f_{out} = f_1 - f_2$. Further discussion on mixing follows in Section 3.

As it is always the case with integer- N PLLs, the frequency resolution (step) of the *individual* PLLs in Figure 1

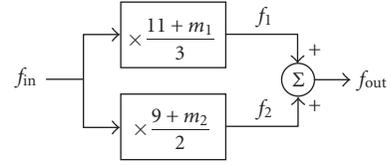


FIGURE 2: A simple two-PLL DFS scheme.

equals PLL's phase-comparator frequency, that is, f_{in}/N_1 and f_{in}/N_2 , respectively. Therefore, to get smaller frequency steps (higher resolution) from a single PLL, a larger prescaler N_i and/or lower reference frequency f_{in} are required. This, necessarily results in a lower phase-comparator frequency f_{in}/N_i implying slowed frequency lock acquisition (agility) and potentially increased spurious signal levels closer to the carrier signal of f_{out} [3].

DFS overcomes these problems as it allows simultaneously for both high phase-comparator frequencies at the constituent PLLs and arbitrarily small frequency step at the output of the synthesizer. In the case of DFS scheme in Figure 1, the frequency step is

$$\delta f_{out} = \frac{f_{in}}{N_1 N_2} \quad (1)$$

which can be orders of magnitude smaller than f_{in}/N_1 and f_{in}/N_2 . This property of DFS is generalized in the case of k PLLs.

Throughout this paper, the prescalers (N_i 's) of the PLLs are considered fixed in size. Moreover, it is assumed that by design, the greatest common divisor of every pair of prescalers, (N_i, N_j), is one, that is, the prescalers are pairwise prime integers; this is a requirement of the DFS methodology [2].

Finally, it is convenient to replace the value of every feedback divider m_i by the sum $\bar{m}_i + m_i$ (e.g., as in Figure 2), where \bar{m}_i is a fixed positive integer and the variable part, m_i , is restricted to take integer values within the range $-N_i$ to N_i . So the range of values of the feedback divider is $\bar{m}_i - N_i, \dots, \bar{m}_i + N_i$.

2.2. Basic numerical example of a two-PLL DFS scheme

Consider the architecture of Figure 2 consisting of two PLLs driven by the same reference frequency f_{in} , whose output frequencies are summed resulting in

$$f_{out} = \left(\frac{11 + m_1}{3} + \frac{9 + m_2}{2} \right) f_{in}. \quad (2)$$

Following DFS methodology [2], the prescalers, $N_1 = 3$ and $N_2 = 2$, are fixed and relatively prime by design (small integers were selected here for illustration purposes).

The feedback dividers are $11 + m_1$ and $9 + m_2$ with $-3 \leq m_1 \leq 3$ and $-2 \leq m_2 \leq 2$. So, *the range of each PLL feedback divider is twice the size of the corresponding prescaler*. These imply that frequency f_1 can take any of seven values

$f_1 \in \{8/3, 9/3, \dots, 14/3\}$ and frequency f_2 can take any of five values $f_2 \in \{7/2, 8/2, \dots, 11/2\}$.

Table 1 shows (some of) the output frequencies f_{out} that can be generated by using the DFS algorithm in [2] to program the values of m_1 and m_2 within their preassumed ranges $-3 \leq m_1 \leq 3$ and $-2 \leq m_2 \leq 2$, respectively. Every one of the thirteen triplets (m_1, m_2, a) in Table 1 satisfies the linear Diophantine equation

$$\frac{m_1}{3} + \frac{m_2}{2} = \frac{a}{6}. \quad (3)$$

This way we can synthesize all frequencies of the form

$$f_{\text{out}} = \bar{f}_{\text{out}} + \frac{a}{6} f_{\text{in}} \quad (4)$$

with the variable \bar{a} taking the values $-6, -5, \dots, 6$ and the central frequency \bar{f}_{out} being

$$\bar{f}_{\text{out}} = \left(\frac{11}{3} + \frac{9}{2} \right) f_{\text{in}} = \frac{49}{6} f_{\text{in}}. \quad (5)$$

Note: the phase comparator frequencies of the individual PLLs are $f_{\text{in}}/3$ and $f_{\text{in}}/2$ while the synthesizer's frequency resolution (step size) is $f_{\text{in}}/6$.

In general, a two-PLL DFS synthesizer results in output frequency

$$f_{\text{out}} = \left(\frac{\bar{m}_1}{N_1} + \frac{\bar{m}_2}{N_2} \right) f_{\text{in}} + \frac{a}{N_1 N_2} f_{\text{in}}, \quad (6)$$

where the variable a can take any of the consecutive values from $-N_1 N_2$ to $N_1 N_2$. This leads, by inspection of (6), to the fundamental property of DFS that the *frequency step can be made much smaller than the phase-comparator frequencies the constituent PLLs, that is,*

$$\frac{f_{\text{in}}}{N_1 N_2} \ll \frac{f_{\text{in}}}{N_1}, \frac{f_{\text{in}}}{N_2}. \quad (7)$$

Expression (6) itself results from our ability to find a convenient solution of the linear Diophantine Equation

$$\frac{m_1}{N_1} + \frac{m_2}{N_2} = \frac{a}{N_1 N_2}. \quad (8)$$

Note that the relationship between m_1 , m_2 , and a , governed by (8), is *nontrivial* and in some cases is not unique, in the sense that there may be more than one pair of integers (m_1, m_2) that solve (8) for a particular value of integer a .

Furthermore, it has been proven that if we have a solution (m_1, m_2) of (8) for $a = 1$, then we can easily generate solutions for every other value of a ; therefore in a hardware implementation, very few numbers have to be stored. A detailed description of how to solve linear Diophantine Equations efficiently is also available in [2].

2.3. DFS synthesizers with k PLLs

The general abstract high-level architecture of k -PLL DFS synthesizers is shown in Figure 3.

TABLE 1: Frequencies of the DFS example in Figure 2.

$f_{\text{out}} = \left(\frac{11}{3} + \frac{9}{2} \right) f_{\text{in}} + \left(\frac{m_1}{3} + \frac{m_2}{2} \right) f_{\text{in}}$					
m_1	m_2	a	$f_{\text{out}}/f_{\text{in}} =$	$\left(\frac{11}{3} + \frac{9}{2} \right) +$	$\left(\frac{m_1}{3} + \frac{m_2}{2} \right)$
-3	0	-6	43/6	=	49/6 + -6/6
-1	-1	-5	44/6	=	49/6 + -5/6
-2	0	-4	45/6	=	49/6 + -4/6
0	-1	-3	46/6	=	49/6 + -3/6
-1	0	-2	47/6	=	49/6 + -2/6
1	-1	-1	48/6	=	49/6 + -1/6
0	0	0	49/6	=	49/6 + 0
-1	1	1	50/6	=	49/6 + 1/6
1	0	2	51/6	=	49/6 + 2/6
0	1	3	52/6	=	49/6 + 3/6
2	0	4	53/6	=	49/6 + 4/6
1	1	5	54/6	=	49/6 + 5/6
3	0	6	55/6	=	49/6 + 6/6

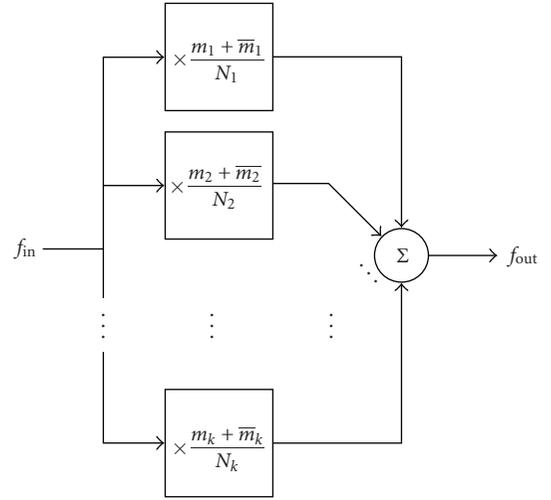


FIGURE 3: Abstract high-level k -PLL DFS scheme.

It has been proven, in [2], that when the integer variables m_1, m_2, \dots, m_k are allowed to take any values in the intervals $-N_1 \leq m_1 \leq N_1, -N_2 \leq m_2 \leq N_2, \dots, -N_k \leq m_k \leq N_k$, respectively, then the following set of frequencies can be synthesized:

$$f_{\text{out}} = \bar{f}_{\text{out}} + \frac{a}{N_1 N_2 \cdots N_k} f_{\text{in}}, \quad (9)$$

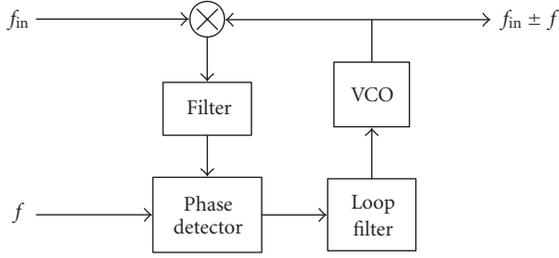


FIGURE 4: Frequency-offset loop.

where a can take any of the values

$$a = -N_1 N_2 \cdots N_k, \dots, N_1 N_2 \cdots N_k \quad (10)$$

and the central frequency \bar{f}_{out} is

$$\bar{f}_{\text{out}} = \left(\frac{\bar{m}_1}{N_1} + \frac{\bar{m}_2}{N_2} + \cdots + \frac{\bar{m}_k}{N_k} \right) f_{\text{in}}. \quad (11)$$

Therefore, the frequency resolution (step) achieved by k -PLL DFS architectures is

$$\delta f_{\text{out}} = \frac{f_{\text{in}}}{N_1 N_2 \cdots N_k}. \quad (12)$$

The central frequency \bar{f}_{out} can be adjusted with resolution δf_{out} as well. The mathematical details, theorems, and their proofs of the general k -PLL DFS architectures can be found in [2].

3. FREQUENCY-OFFSET DFS ARCHITECTURES FOR VERY HIGH FRACTIONAL-FREQUENCY RESOLUTION

Synthesizers with very high fractional-frequency resolution like microphase steppers, advanced signal generators, certain instrumentation equipment, atomic-clock synthesizers, and so forth, often have performance specifications that challenge existing technology solutions especially under cost, power, size, and complexity constraints. DFS offers a new alternative to DDS and fractional- N PLLs in the design of such systems.

For this kind of applications, most appropriate DFS architecture has been proven to be the one based on frequency offsetting [3]. Since frequency offsetting requires mixing, a few comments are in order without any intention to cover the topic of mixing.

3.1. Frequency mixing

Mixing of two periodic signals at frequencies f_1 and f_2 is denoted by \otimes , see Figure 4, and the outcome is typically chosen to be either $f_1 + f_2$ or $f_1 - f_2$.

Mixing of three or more signals has a similar interpretation, note however that the order of performing the mixing of the signals may be important for getting a spectrally pure output signal. In general, minimization of mixing spurs involves the choice of the central frequencies of f_1 and f_2 , their frequency ranges, the choice of the sum or difference,

the harmonic contents of the mixed signals, and of course the type of the mixers.

The key to low-output spurs in DFS synthesizers is the mixing method since the mixers are the dominant spurs generating circuit elements.

3.2. Frequency offsetting

The synthesizer architecture in Figure 4 is convenient for deriving the sum or difference between a large f_{in} and a small offset frequency f .

When $f/f_{\text{in}} \ll 1$, the mixing of f_{in} with $f_{\text{in}} \pm f$ can be performed without difficulty and the mixing spurs can be minimal, for example, [5]. Therefore frequency offsetting is an effective approach to achieving the frequency summations and/or subtractions needed to realize DFS with central output frequency close to f_{in} .

The following subsections illustrate this approach for the case of two- and three-PLL DFS schemes. In principle, the structure of Figure 4 can be cascaded k times to create k -PLL DFS architectures.

3.3. Two-PLL frequency-offset DFS architecture

Figure 5 shows how two DFS-determined PLLs can be cascaded using an offset synthesizer structure to form a DFS architecture, where the variable f_{out} can be adjusted in very small-frequency steps from the reference f_{in} .

Based on the DFS theory [2], the two PLL output frequencies f_1, f_2 (we can consider divider R as part of the PLLs) are determined by the common dividers Q, R , the two relatively prime integers N_1, N_2 , and the feedback dividers $pN_1 + m_1$ and $pN_2 + m_2$ which are partitioned into the fixed, pN_1, pN_2 , and the variable, m_1, m_2 , parts. The values of m_1, m_2 program the value of parameter a in expression (6). The fixed integers pN_1 and pN_2 partially define the central frequencies \bar{f}_1, \bar{f}_2 of the PLLs. In this application, we also like to have $\bar{f}_1 = \bar{f}_2$ which implies that $f_{\text{out}} = f_{\text{in}}$ when $m_1 = m_2 = 0$.

Variables m_1 and m_2 are allowed to take any value within their ranges $-N_1, \dots, N_1$ and $-N_2, \dots, N_2$, respectively. This results in output-frequency resolution equal to $\delta f_{\text{out}} = f_{\text{in}}/(QRN_1N_2)$ and output-frequency range (equal to or greater than) $\Delta f_{\text{out}} = \pm f_{\text{in}}/(QR)$. (Note the product of N_1 and N_2 in the denominator, in contrast to Q that is accounted for on its first power.)

The factor Q in the denominators determines the pullability ranges of the VCOs in the PLLs. Specifically, given the ranges of m_1 and m_2 , the VCO's fractional pullability is $PL_{\text{PLL},1,2} = \pm (100/Q) \%$. The role of p in the numerators is to adjust for the central output frequencies of the PLLs by counterbalancing Q . The phase-comparator frequencies of the PLLs are $f_{\text{PC,PLL},i} = f_{\text{in}}/(QN_i), i = 1, 2$.

Finally, R is a large divider necessary to generate the relatively small frequencies f_1, f_2 from the output frequencies of the PLLs. Divider R also contributes to the output resolution of the synthesizer and the spectral purity of signals entering the frequency-offset blocks.

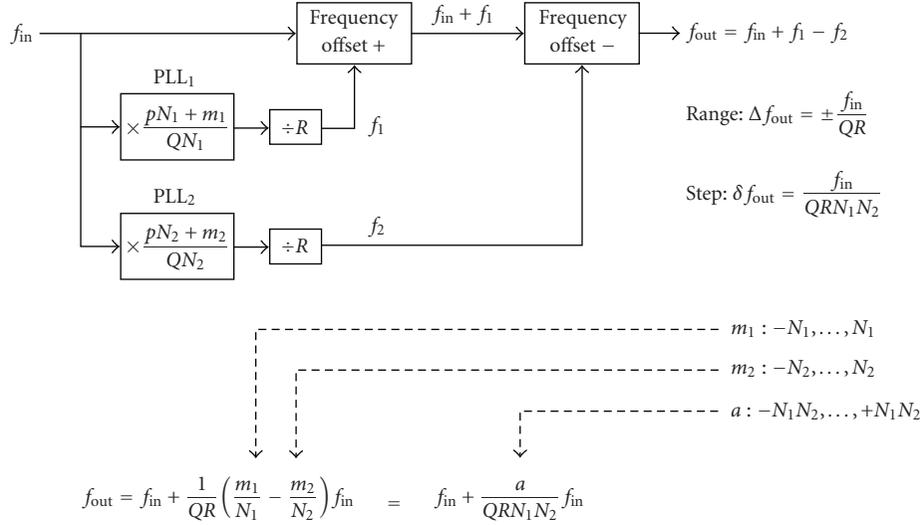


FIGURE 5: Two-PLL frequency-offset DFS scheme.

$\left. \begin{array}{l} \delta f_{\text{out}} \cong 10 \mu\text{Hz} \\ \Delta f_{\text{out}} = \pm 10 \text{Hz} \\ f_{\text{in}} = 10 \text{MHz} \\ \bar{f}_{\text{out}} = 10 \text{MHz} \end{array} \right\} \rightarrow \left\{ \begin{array}{l} Q = p = 10 \\ R = 100000 \\ N_1 = 1000 \\ N_2 = 1017 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \bar{f}_{\text{PLL1}} = \bar{f}_{\text{PLL2}} = 10 \text{MHz} \\ f_{\text{PC.PLL1}} = f_{\text{PC.PLL2}} \cong 1 \text{kHz} \\ \text{PL}_{\text{PLL1}} = \text{PL}_{\text{PLL2}} = \pm 10\% \\ \bar{f}_1 = \bar{f}_2 = 100 \text{Hz} \end{array} \right.$
--

FIGURE 6: Numerical example of the two-PLL frequency-offset DFS scheme in Figure 5.

With a microphase stepper application in mind, Figure 6 shows a choice of values for N_1 , N_2 , R , Q , and p , and the corresponding characteristics and performance of the synthesizer they result in. A resolution of $10 \mu\text{Hz}$ is probably the best that the two-PLL scheme with 10MHz input frequency could give. Note that although the 100Hz frequency offset is not uncommon in these types of systems, a higher frequency would be helpful. The pullability range of $\pm 10\%$ is achievable by tunable LC oscillators, and because of the large divider R , the phase noise of the oscillators is not a critical issue.

As described in Figure 6, the magnitude of R was made large compared to Q , N_1 , and N_2 to achieve the desired frequency-step resolution of $\pm 10 \mu\text{Hz}$ while keeping the PLL phase-comparator frequencies relatively large and easy to filter. This choice was not directed through any fundamental constraint in the DFS method, but was made from our design emphasis on high-spectral purity over acquisition speed in a simple, practical circuit implementation. In the following subsection, we see how adding one more PLL allows for more choices of the parameters and, in principle, better overall performance.

3.4. Three-PLL frequency-offset DFS architecture

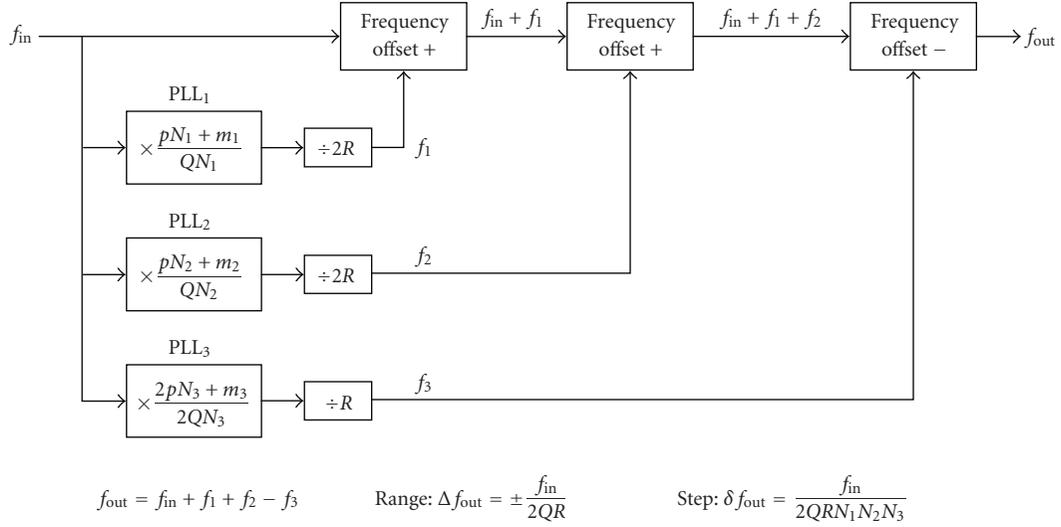
A three-PLL frequency-offset DFS architecture is shown in Figure 7. Its principles of operation are very similar to those of the two-PLL one in Figure 5. The major difference is that

because of the odd number of PLLs, centering f_{out} with respect to the input reference f_{in} requires further design consideration.

Specifically, it is desirable that $m_1 = m_2 = m_3 = 0$ implies $f_{\text{out}} = f_{\text{in}}$. To achieve this, we add f_1 and f_2 to f_{in} and subtract f_3 . Moreover, we introduce factors of 2 in PLL 3 and in the R -dividers of PLLs 1 and 2. These result in $2\bar{f}_1 = 2\bar{f}_2 = \bar{f}_3 = p f_{\text{in}} / (QR)$ and equal ranges of f_1 , f_2 , and f_3 . However, the pullability range of PLL 3 is the half of that of PLLs 1 and 2. The expression for f_{out} is shown in Figure 7.

Integers N_1, N_2 , and N_3 are chosen to be pairwise prime and the variables m_1 , m_2 , and m_3 take values within their ranges $-N_1, \dots, N_1$, $-N_2, \dots, N_2$, and $-N_3, \dots, N_3$, respectively. The resulting output frequency resolution is $\delta f_{\text{out}} = f_{\text{in}} / (2QRN_1N_2N_3)$ and the output frequency range is (equal to or greater than) $\Delta f_{\text{out}} = \pm f_{\text{in}} / (2QR)$.

Again, with a microphase stepper application in mind, Figure 8 shows a choice of numerical values for N_1, N_2, N_3, R, Q , and p , as well as the corresponding characteristics of the resulting synthesizer. Output frequency resolution of $1 \mu\text{Hz}$ and output range of about $\pm 16 \text{Hz}$ are achieved. The frequency offset has been raised to 500Hz for PLLs 1, 2 and to 1000Hz for PLL 3, and the pullability ranges have dropped to about $\pm 3\%$ and $\pm 1.6\%$, respectively. Therefore, as expected, the three-PLL case provides much more flexibility in the design and much better characteristics.



$$\begin{aligned}
 & \text{---} m_1 : -N_1, \dots, N_1 \\
 & \text{---} m_2 : -N_2, \dots, N_2 \\
 & \text{---} m_3 : -N_3, \dots, N_3 \\
 & \text{---} a : -N_1N_2N_3, \dots, +N_1N_2N_3
 \end{aligned}$$

$$f_{out} = f_{in} + \frac{1}{2QR} \left(\frac{m_1}{N_1} + \frac{m_2}{N_2} - \frac{m_3}{N_3} \right) f_{in} = f_{in} + \frac{a}{2QRN_1N_2N_3} f_{in}$$

FIGURE 7: Three-PLL frequency-offset DFS scheme.

$ \left. \begin{aligned} \delta f_{out} &\cong 1 \mu\text{Hz} \\ \Delta f_{out} &= \pm 16 \text{ Hz} \\ f_{in} &= 10 \text{ MHz} \\ \bar{f}_{out} &= 10 \text{ MHz} \end{aligned} \right\} \rightarrow \left\{ \begin{aligned} Q &= p = 31 \\ R &= 10000 \\ N_1 &= 319 \\ N_2 &= 317 \\ N_3 &= 158 \end{aligned} \right\} \Rightarrow \left\{ \begin{aligned} \bar{f}_{PLL_1} &= \bar{f}_{PLL_2} = \bar{f}_{PLL_3} = 10 \text{ MHz} \\ f_{PC.PLL_1} &= f_{PC.PLL_2} = f_{PC.PLL_3} \cong 1 \text{ kHz} \\ PL_{PLL_1} &= PL_{PLL_2} = \pm 3\%, PL_{PLL_3} = \pm 1.6\% \\ \bar{f}_1 &= \bar{f}_2 = 500 \text{ Hz}, \bar{f}_3 = 1 \text{ kHz} \end{aligned} \right. $
--

FIGURE 8: Numerical example of the three-PLL frequency-offset DFS scheme in Figure 7.

4. SUMMARY

In summary, the general structure of DFS architectures provides the following desirable properties: the ability to achieve a predetermined center frequency

$$\bar{f}_{out} = \left(\frac{\bar{m}_1}{N_1} + \frac{\bar{m}_2}{N_2} + \dots + \frac{\bar{m}_k}{N_k} \right) f_{in} \quad (13)$$

with frequency range

$$\bar{f}_{out} - f_{in} \quad \text{to} \quad \bar{f}_{out} + f_{in} \quad (14)$$

and frequency step (resolution) of

$$\delta f_{out} = \frac{f_{in}}{N_1N_2 \dots N_k}, \quad (15)$$

while the phase-comparator frequencies of the constituent PLLs are

$$\frac{f_{in}}{N_1}, \frac{f_{in}}{N_2}, \dots, \frac{f_{in}}{N_k}. \quad (16)$$

The application of DFS permits high flexibility on the relationship of the fixed-frequency reference to output frequency (9) with wide-frequency range (14). Based on (15), we have shown the design of very fine frequency resolution using two- and three-PLL nested DFS frequency-offset loops.

In the case of the three-loop system described in Figure 7, a fractional frequency synthesizer capable of 1E-13 has been numerically demonstrated. The method of cascading nested frequency-offset DFS architectures to higher orders would ultimately result in frequency steering resolution approaching 1E-15, consistent with the needs of most precision time-keeping laboratories contributing to UTC.

ACKNOWLEDGMENTS

The authors would like to express their appreciation to Dr. Demetrios Matsakis, Mr. Warren Walls, and Mr. Andradige Silva for their help on the Diophantine project.

REFERENCES

- [1] P. P. Sotiriadis, "Diophantine frequency synthesis a number theory approach to fine frequency synthesis," in *Proceedings of IEEE International Frequency Control Symposium and Exposition*, pp. 48–53, Miami, Fla, USA, June 2006.
- [2] P. P. Sotiriadis, "Diophantine frequency synthesis," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 53, no. 11, pp. 1988–1998, 2006.
- [3] W. F. Egan, *Frequency Synthesis by Phase Lock*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1999.
- [4] S. Cheng, J. R. Jensen, R. E. Wallis, and G. L. Weaver, "Further enhancements to the analysis of spectral purity in the application of practical direct digital synthesis," in *Proceedings of IEEE International Frequency Control Symposium and Exposition*, pp. 462–470, Montreal, Canada, August 2004.
- [5] P. P. Sotiriadis and G. L. Weaver, "A diophantine frequency synthesizer for the examination of high spectral purity," in *Proceedings of IEEE International Frequency Control Symposium Jointly with the 21st European Frequency and Time Forum*, pp. 1092–1098, Geneva, Switzerland, May-June 2007.
- [6] D. E. Flath, *Introduction to Number Theory*, John Wiley & Sons, New York, NY, USA, 1989.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

