Diophantine Frequency Synthesis A Number Theory Approach to Fine Frequency Synthesis

Paul P. Sotiriadis
Department of Electrical and Computer Engineering
Johns Hopkins University, Baltimore, MD, U.S.A.
Email: pps@jhu.edu

Abstract— **Diophantine Frequency Synthesis[1] is a new approach to fine-step[2], fast-hopping, low-spurs frequency synthesis that is based on mathematical properties of integer numbers and linear Diophantine equations. Diophantine Frequency Synthesis overcomes the constraining relation between frequency step and phase-comparator frequency inherent in conventional phase-locked loops. It leads to fine-step, fast-hopping, modular-structured frequency synthesizers with potentially very low spurs, especially in the vicinity of the carrier.**

I. Introduction

Fine-step frequency synthesis is important in many applications including scientific instrumentation, atomic clocks, timing systems, satellite navigation, certain classes of communication systems and medical systems. Several fine-step frequency synthesis architectures have been proposed and a rich collection of them can be found in [1] and [2].

The *Diophantine Frequency Synthesis*[1,3] (DFS) is a new approach to fine-step[2] frequency synthesis that is based on mathematical properties of integer numbers and linear Diophantine equations. By definition, a Diophantine equation is an algebraic equation whose solutions are required to be integers, [3].

DFS uses two or more basic PLLs. The output frequencies of the PLLs are added or subtracted to give the output frequency of the synthesizer. DFS provides the mathematical algorithm for choosing the sizes of the prescalers and adjusting the sizes of the feedback dividers.

DFS has a significant advantage: it leads to PLL-based architectures for which the output frequency step can be made very small without using large prescalers or small reference frequencies. This allows for simultaneously having very small output frequency step and high phase-comparator frequencies resulting in large loop bandwidths and therefore fast frequency hopping. DFS *distributes* the frequency resolution among the PLLs.

[1]Patent Pending: Johns Hopkins Univ. Applied Physics Laboratory

[2] E.g. 10^{-10} step relative to carrier can be easily achieved.

[3]More details are available in the corresponding journal paper by the same author [4]

II. Preliminaries

DFS uses two or more basic *phase locked loops* (PLLs) like the one in Figure 1[4].

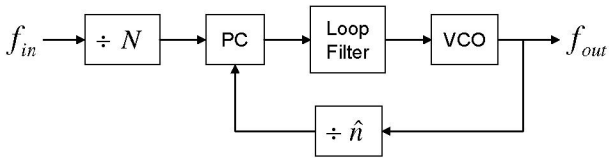

Fig. 1. Basic PLL

Throughout the paper, the *prescaler* (divider N) is assumed to have a *fixed* size, N. The size of the *feedback divider* ($\hat{n}$) is the sum $\hat{n} = \bar{n} + n$, of a fixed value $\bar{n}$, and, a variable n which can take both *negative and positive values* within a predefined range. For all values of n, $\hat{n}$ is positive. The output frequency of the PLL is

$$f_{out} = \frac{\hat{n}}{N} f_{in}.$$

This paper focuses on high-level architectures for frequency synthesis and not in the technical details of the individual PLLs, so Figure 2 is used for convenience instead of Figure 1. It is agreed however that simplification of the fraction $\hat{n}/N$ is *not* allowed i.e. $\hat{n}/N$ and $k\hat{n}/(kN)$ correspond to two *different* PLLs.

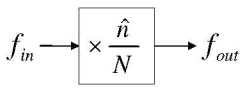

Fig. 2. Simplified schematic of the basic PLL

Note that frequency synthesis using a single PLL, as that in Figure 1, implies frequency steps that are equal to the phase-comparator frequency, i.e. equal to f_{in}/N.

[4]The Phase Comparator (PC) may also be a phase-frequency comparator or phase detector etc.

1-4244-0074-0/06/$20.00 © 2006 IEEE.

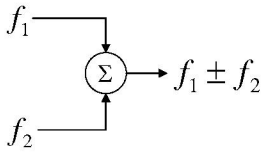

Fig. 3. Mixer

This means that small frequency steps (using large N and/or small f_{in}) requires low phase-comparator frequency f_{in}/N and therefore small loop bandwidth [1], [2]. The last one implies slow frequency hopping and possibly increased spurious signals close to f_{out}. DFS overcomes these problems as it allows for both high phase-comparator frequency and very small frequency step at the same time.

Mixing of two signals at frequencies f_1 and f_2 is denoted as in Figure 3 where the outcome can be either $f_1 + f_2$ or $f_1 - f_2$. The context in the paper always indicates whether the sum or the difference is considered. However, as shown in the following sections, the choice of the sum or the difference dictates only the central frequency of the output signal and not the resolution or the range of the DFS synthesizer.

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 clean output signal.

Note that minimization of mixing spurs involves the choice of the central frequencies and frequency ranges of the mixed signals, the choice of the sum or difference of their frequencies and of course the type of the mixer. The examples that follow provide some indications regarding these choices, however, the goal of this paper is only to lay out the principles of DFS.

III. DFS: A Motivating Example

DFS uses mathematical properties of integer numbers to achieve very fine frequency step without using large frequency dividers. Specifically, it combines the outputs of two or more PLLs with (small) prescalers, $N_1, N_2, \ldots, N_k$, to achieve output frequency resolution equal to $f_{in}/(N_1 N_2 \cdots N_k)$ and output frequency range $2f_{in}$. Small prescalers imply high phase-comparator frequencies, f_{in}/N_i, and fast frequency hopping.

Consider the simple architecture of Figure 4 consisting of two PLLs whose output frequencies are summed giving

$$f_{out} = \left(\frac{n_1}{3} + \frac{n_2}{5} + \frac{12}{3} + \frac{16}{5} \right) f_{in} \qquad (1)$$

The prescalers are 3 and 5 (fixed). Let the feedback dividers, $12 + n_1$ and $16 + n_2$, be variable and such that

$$-3 \le n_1 \le 3 \quad \text{and} \quad -5 \le n_2 \le 5$$

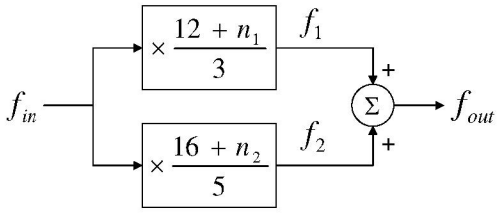

Fig. 4. A simple DFS scheme

i.e. *the range of each feedback divider is twice the size of the corresponding prescaler*[5]. Then, f_1 can take any of the 7 values:

$$f_1 \in \left\{ \frac{9}{3} f_{in}, \frac{10}{3} f_{in}, \frac{11}{3} f_{in}, \ldots, \frac{15}{3} f_{in} \right\} \qquad (2)$$

and f_2 can take any of the 11 values:

$$f_2 \in \left\{ \frac{11}{5} f_{in}, \frac{12}{5} f_{in}, \frac{13}{5} f_{in}, \ldots, \frac{21}{5} f_{in} \right\} \qquad (3)$$

Table I shows a set of output frequencies f_{out} that can be synthesized by appropriately choosing n_1 and n_2 *within their specified ranges*. Specifically, we can synthesize all frequencies of the form

$$f_{out} = \overline{f_{out}} + \frac{a}{15} f_{in}$$

with a taking the values $-15, -14, -13, \ldots, 14, 15$, and the central frequency being $\overline{f_{out}} = \frac{108}{15} f_{in}$. In contrast, the output frequencies of the individual PLLs are

$$f_1 = \frac{12}{3} f_{in} + \frac{n_1}{3} f_{in}$$

and

$$f_2 = \frac{16}{5} f_{in} + \frac{n_2}{5} f_{in}.$$

Table I shows how to find a solution (n_1, n_2) of the Diophantine equation

$$\frac{n_1}{3} + \frac{n_2}{5} = \frac{a}{15}$$

within the set $\{-3, -2, \ldots, 3\} \times \{-5, -4, \ldots, 5\}$, for every value of a in $\{-15, -14, \ldots, 15\}$.

The relation between n_1, n_2 and a is non-trivial and in same cases it is not unique. However, it can be shown, [4] that if we have the solution for $a = 1$, we can very easily generate solutions for every other value of a; therefore, very few numbers have to be stored. A detailed description of how to solve linear Diophantine equations efficiently is available in [4].

Now let's consider the simple architecture in Figure 4 but with feedback dividers 6 and 15 instead of 3 and 5 respectively. Now, n_1 ranges from -6 to 6 and n_2 ranges from -15 to 15.

[5]This specifies the required pull-ability range of the VCOs.

$$f_{out} = \left(\frac{n_1}{3} + \frac{n_2}{5} + \frac{12}{3} + \frac{16}{5}\right) f_{in}$$

n_1	n_2	$\frac{f_{out}}{f_{in}}$	=	$\left(\frac{n_1}{3} + \frac{n_2}{5}\right)$	+	$\left(\frac{12}{3} + \frac{16}{5}\right)$
-3	0	93/15	=	-15/15	+	108/15
-1	-3	94/15	=	-14/15	+	108/15
-2	-1	95/15	=	-13/15	+	108/15
-3	1	96/15	=	-12/15	+	108/15
-1	-2	97/15	=	-11/15	+	108/15
-2	0	98/15	=	-10/15	+	108/15
-3	2	99/15	=	-9/15	+	108/15
-1	-1	100/15	=	-8/15	+	108/15
-2	1	101/15	=	-7/15	+	108/15
-3	3	102/15	=	-6/15	+	108/15
-1	0	103/15	=	-5/15	+	108/15
-2	2	104/15	=	-4/15	+	108/15
-3	4	105/15	=	-3/15	+	108/15
-1	1	106/15	=	-2/15	+	108/15
-2	3	107/15	=	-1/15	+	108/15
0	0	108/15	=	0/15	+	108/15
-1	2	109/15	=	1/15	+	108/15
-2	4	110/15	=	2/15	+	108/15
0	1	111/15	=	3/15	+	108/15
-1	3	112/15	=	4/15	+	108/15
1	0	113/15	=	5/15	+	108/15
0	2	114/15	=	6/15	+	108/15
-1	4	115/15	=	7/15	+	108/15
1	1	116/15	=	8/15	+	108/15
0	3	117/15	=	9/15	+	108/15
2	0	118/15	=	10/15	+	108/15
1	2	119/15	=	11/15	+	108/15
0	4	120/15	=	12/15	+	108/15
2	1	121/15	=	13/15	+	108/15
1	3	122/15	=	14/15	+	108/15
3	0	123/15	=	15/15	+	108/15

TABLE I

FREQUENCIES SYNTHESIZED BY THE DFS ARCHITECTURE OF FIGURE 4

In the line of the previous example one might expect that f_{out} would range from $\overline{f_{out}} - f_{in}$ to $\overline{f_{out}} + f_{in}$ and the frequency step would be $f_{in}/(6 \cdot 15) = f_{in}/90$. By calculating the output frequencies corresponding to all allowed pairs (n_1, n_2) we see that the expected output range is indeed achievable, however, the frequency step (resolution) is $f_{in}/30$ and therefore three times larger that the "expected" $f_{in}/(6 \cdot 15)$. Note that[6] $\mathrm{lcm}(6, 15) = 30$.

Question: *What are the special qualities of the pair of numbers "3" and "5" leading to properties 1,2 and 3?*

Answer: *They are pairwise relatively prime integers, i.e.*[7] $\gcd(3, 5) = 1$. This is not true for the pair $(6, 15)$.

The formal answer to the above question is given in Section IV. Note also that for both presecalers pairs (N_1, N_2), the frequency step is $f_{in}/\mathrm{lcm}(N_1, N_2)$.

The choice of the constants $\bar{n}_1$ and $\bar{n}_2$ ($\bar{n}_1 = 12$ and $\bar{n}_2 = 16$ in Figure 4) as well as the choice of $+$ or $-$ in the mixer define the central frequency $\overline{f_{out}}$ and the frequency ratio f_1/f_2 in the mixer. Therefore, these choices can be used to minimize the spurious signals generated by the mixer. Moreover, $\bar{n}_1$ and $\bar{n}_2$ along with N_1 and N_2 specify the required tuning ranges (pullability) of the VCOs.

[6]lcm = least common multiple

[7]gcd = greatest common divider

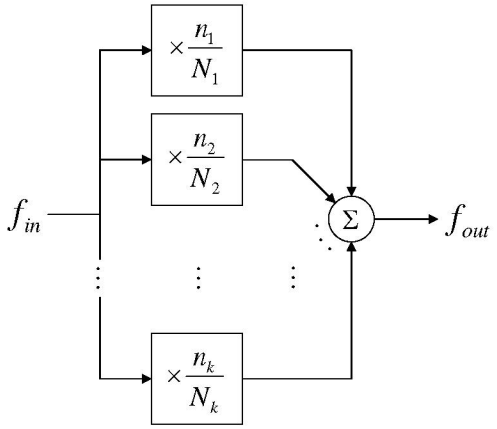

Fig. 5. k-PLLs DFS scheme. Parameters $\bar{n}_1$ to $\bar{n}_k$ are omitted.

IV. THE MATHEMATICAL FRAMEWORK OF DIOPHANTINE FREQUENCY SYNTHESIS

DFS achieves very small frequency steps while using small prescalers, $N_1, N_2, \ldots, N_k$, at the same time. The example in the previous section is a special case of the general DFS theory for k-PLL architectures, like the one in Figure 5.

The mathematical principle of DFS is given by Theorem 4.1 below. First we concentrate on the variable part of the output frequency expression for k PLLs

$$\frac{x_1}{N_1} + \frac{x_2}{N_2} + \ldots + \frac{x_k}{N_k} = \frac{a}{N_1 N_2 \cdots N_k}. \tag{4}$$

Theorem 4.1: If $N_1, N_2, \ldots, N_k$ are pairwise relatively prime positive integers, then for every integer a such that $-N_1 N_2 \cdots N_k \leq a \leq N_1 N_2 \cdots N_k$ the Diophantine equation (4) has a solution $(x_1, x_2, \ldots, x_k)$ where $-N_i \leq x_i \leq N_i$ for all $i = 1, 2, \ldots, k$.

The proof with all the details can be found in [4].

Note that Theorem 4.1 guarantees only the existence of a solution within the specified bounds, $|x_i| \leq N_i$ for all i's, but not the uniqueness of it. In Table I for example we see that the equation $n_1/3 + n_2/5 = -1/15$ has (at least) two solutions: $-2/3 + 3/5 = 1/3 - 2/5 = -1/15$.

Also, Theorem 4.1 guarantees the existence of a solution within the specified bounds when $|a| \leq N_1 N_2 \ldots N_k$, but it does *not* say that a solution within those bounds cannot be found for values of a that are absolutely larger than $N_1 N_2 \ldots N_k$.

Interpretation of Theorem 4.1: Rephrasing Theorem 4.1 we can say that given a set $N_1, N_2, \ldots, N_k$ of pairwise relatively prime positive integers, **all** rational numbers from -1 to 1 with uniform step (resolution) $1/(N_1 N_2 \cdots N_k)$ are generated by the sum $x_1/N_1 + x_2/N_2 + \ldots + x_k/N_k$ when the numerators $x_1, x_2, \ldots, x_k$ vary within the intervals $-N_i \leq x_i \leq N_i$, $i = 1, 2, \ldots, k$.

V. Fixed Frequency DFS An Example

There are applications requiring the generation of a periodic signal of a specific but fixed frequency f_{out} using a reference signal at a given frequency f_{in}. This is typical in atomic clocks and related time reference systems. Lets consider the following example:

The input frequency is $f_{in} = 10MHz$ and the desired output frequency is $f_{out} = 9.285,739,4MHz$ which must be synthesized with accuracy of $0.1Hz$.

To achieve $0.1Hz$ resolution with only one PLL, a prescaler equal to or greater than $f_{in}/0.1Hz = 10^8$ is required. This is definitely impractical for most realistic situations. Although other techniques can be used to achieve this resolution [1]-[2], the Diophantine approach is straight forward. Two scenarios are presented using combinations of two and three basic PLLs respectively.

A. Two PLLs DFS scheme

Let the prescalers of the two PLLs be the pair of relatively prime integers N_1 and N_2. Moreover lets assume for simplicity that $N_1 \approxeq N_2$. From Section IV we know that the output frequency resolution of the synthesizer is $f_{in}/(N_1 N_2)$. Since an accuracy of $0.1Hz$, or better, is required while having input frequency of $10MHz$, it must be that $N_1 N_2 \geq 10^8$. This implies that $N_1, N_2 \approxeq sqrt(10^8) = 10,000$, assuming that we want to keep the prescalers as small as possible.

We can pick for example the pair of relatively prime integers $N_1 = 10,000$ and $N_2 = 10,003$ with $N_1 N_2 = 100,030,000$. Then the phase-comparator frequencies of the PLLs are about $1kHz$. A DFS scheme based on these prescalers is shown in Figure 6.

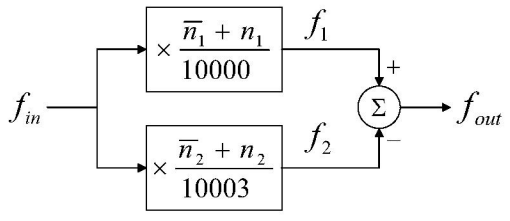

Fig. 6. Two PLLs DFS scheme

For now we ignore $\bar{n}_1$ and $\bar{n}_2$ ($\bar{n}_1 = \bar{n}_2 = 0$) and focus our attention on tuning f_{out} using n_1 and n_2.

From section IV we know that by choosing appropriate values for n_1 and n_2 (and ignoring $\bar{n}_1$ and $\bar{n}_2$), f_{out} can take any value

$$f_{out} = \frac{n}{100,030,000} f_{in} \tag{5}$$

where n ranges from $-100,030,000$ to $100,030,000$.

Choosing $n = 92,885,251$ results in output frequency $f_{out} = 9.285,739,378\ldots MHz$. This is the best possible approximation to the desired frequency $9.285,739,4\,MHz$ by (5). The frequency error is within the acceptable limits.

Now we derive n_1 and n_2 corresponding to the particular value of n. To do so we must solve the Diophantine equation

$$\frac{n_1}{10000} + \frac{-n_2}{10003} = \frac{n}{100,030,000}. \tag{6}$$

Note that the minus sign (due to frequency mixing in the scheme of Figure 6) does not cause any complication since the ranges of n_1 and n_2 are symmetric with respect to zero. We simple solve (6) for n_1 and $(-n_2)$.

To proceed we use the "gcd" function of MATLAB[8]. It gives "$gcd(10003, 10000) = [1, -3333, 3334]$" and so

$$\frac{-3333}{10000} + \frac{3334}{10003} = \frac{1}{100,030,000}.$$

Following the algorithm in [4] we set

$$\begin{aligned} y_1 &= (-3333 \cdot n) \mod 10000 \\ &= 8417 \\ y_2 &= (3334 \cdot n) \mod 10003 \\ &= 869 \end{aligned}$$

Since

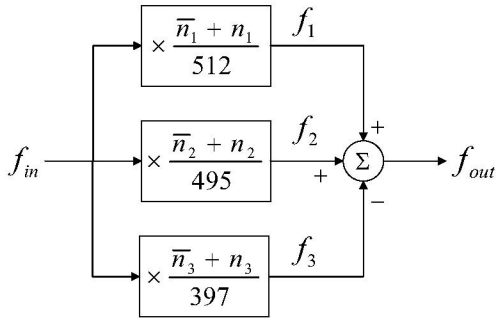

Fig. 7. Three PLLs DFS scheme

$f_2 \cong 49.131,260,6\,MHz$ and therefore a ratio f_2/f_1 close to 0.85 resulting in relatively low mixing spurs [1].

Three PLLs DFS scheme

Use of three PLLs allows for more flexibility. Lets assume again that $N_1 \cong N_2 \cong N_3$. If $N_i \cong \sqrt[3]{10^8} \cong 464$ the we get $N_1N_2N_3 \cong 10^8$. An appropriate triplet of pairwise relatively prime integers is $N_1 = 512$, $N_2 = 495$ and $N_3 = 397$ giving $N_1N_2N_3 = 100,615,680$.

A DFS scheme using these numbers is shown in Figure 7. Lets ignore the constants $\bar{n}_1$, $\bar{n}_2$ and $\bar{n}_3$ for the moment (consider $\bar{n}_1 = \bar{n}_2 = \bar{n}_3 = 0$ for now). Then, by adjusting n_1, n_2 and n_3 the output frequency can take any of the values

$$f_{out} = \frac{n}{100,615,680}\, f_{in} \tag{7}$$

where n ranges from $-100,615,680$ to $100,615,680$. The best approximation of the desired frequency, $9.285,739,4\,MHz$, with $f_{in} = 10MHz$, is achieved using $n = 93,429,098$. Now we have to solve (8) for n_1, n_2 and n_3. The minus sign in n_3 is due to frequency mixing in the scheme of Figure 7.

$$\frac{n_1}{512} + \frac{n_2}{495} + \frac{-n_3}{397} = \frac{93,429,098}{100,615,680} \tag{8}$$

Using the algorithm in [4] we conclude that $n_1 = -114$, $n_2 = 217$ and $n_3 = -283$.

Putting $\bar{n}_1$, $\bar{n}_2$ and $\bar{n}_3$ back into the equation we have

$$f_{out} = \Big(\frac{\bar{n}_1}{N_1} + \frac{\bar{n}_2}{N_2} - \frac{\bar{n}_3}{N_3}\Big) f_{in} + \frac{93,429,098}{100,615,680}\, f_{in}$$

As in the previous example, we can choose the values of $\bar{n}_1$, $\bar{n}_2$ and $\bar{n}_3$ to minimize mixing spurs or noise, or bring f_1, f_2 and f_3 within the operating range of existing PLLs, or optimize some other criterion. However, we would like to do so without changing f_{out} since it already has the desirable value. Therefore

$$\frac{\bar{n}_1}{N_1} + \frac{\bar{n}_2}{N_2} - \frac{\bar{n}_3}{N_3} = 0 \tag{9}$$

Since N_1, N_2 and N_3 are pairwise relatively prime, equation (9) implies that[9] $\bar{n}_1 = c_1N_1$, $\bar{n}_2 = c_2N_2$ and $\bar{n}_3 = c_3N_3$ with $c_1 + c_2 - c_3 = 0$.

An eligible choice, for example is $c_1 = 3$, $c_2 = 1$ and $c_3 = 4$. This gives $f_1 = 27.773,437,50$, $f_2 = 14.383,838,38$ and $f_3 = 47.128,463,48$.

VI. Variable Frequency DFS An Example

Suppose we want to design a DFS synthesizer that can generate frequencies from $2MHz$ to $4MHz$ with resolution of about $1Hz$. From the theory in Section IV we know that the general architecture of Figure 5, with input frequency f_{in}, can generate all frequencies from $\overline{f_{out}} - f_{in}$ to $\overline{f_{out}} + f_{in}$ with resolution $f_{in}/(N_1N_2\cdots N_k)$. Since the frequency range is $2f_{in}$ we can choose

$$f_{in} = 1MHz. \tag{10}$$

Then, the resolution requirement is satisfied if

$$N_1N_2\cdots N_k \geq 10^6. \tag{11}$$

Suppose we add the requirement that the phase-comparator frequencies in all PLLs are about $10kHz$. This means that

$$\frac{f_{in}}{N_i} \cong 10kHz\,, \quad i = 1,2,\ldots,k. \tag{12}$$

Relations (10) and (12) imply that $N_1 \cong N_2 \cong N_3 \cong 100$. Therefore, the minimum number of PLLs, k, that satisfies (11) is $k = 3$. Three convenient pairwise relatively prime numbers are $N_1 = 100$, $N_2 = 101$ and $N_3 = 103$.

The next step is to decide what the central frequencies of the three PLLs should be and how they will be mixed i.e. added or subtracted. Since the purpose of this paper is solely to present the mathematical principles of DFS, many technical issues[10] involved in these decisions are not discussed here.

A simple choice is[11] $\overline{f_1} = 55MHz$, $\overline{f_2} = 40MHz$, and $\overline{f_3} = 18MHz$ and the output frequency is chosen to be $f_{out} = -(f_1 - f_2) + f_3$ resulting to $\overline{f_{out}} = 3MHz$. Since $\overline{f_i} = (\bar{n}_i/N_i)f_{in}$, $i = 1,2,3$, we have $\bar{n}_1 = 5500$, $\bar{n}_2 = 4040$ and $\bar{n}_3 = 1854$. The corresponding DFS architecture is shown in Figure 8.

The frequency ranges of the PLLs and of the output signal, along with their resolutions, are shown in Table II. The output frequency can take all values

[9]The proof is available in Ref. [4].

[10]like the pullability of the PLLs, the spurs generated by the mixing, the possible filtering of the PLLs' signals before mixing, the minimization of the output phase noise etc.

[11]No effort has been made to optimize this choice. Using more elaborate mixing schemes one can possibly reduce $\overline{f_1}$, $\overline{f_2}$ and $\overline{f_3}$ while maintaining a clean output spectrum.

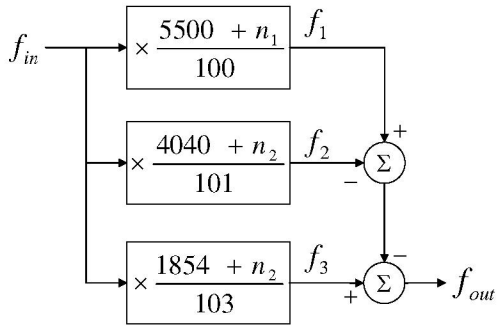

Fig. 8. 3-PLLs Variable Frequency DFS scheme

All frequencies are in MHz

	Min	Central	Max	Frequency step (resolution)
f_{in}	-	1	-	-
f_1	54	55	56	1 / 100
f_2	39	40	41	1 / 101
f_3	17	18	19	1 / 103
f_{out}	2	3	4	1 / 1,040,300

TABLE II

FREQUENCY RANGES AND FREQUENCY STEPS (RESOLUTIONS) OF THE SIGNALS IN THE DFS SCHEME OF FIGURE 8.

$$f_{out} = \left(3 + \frac{n}{1,040,300}\right) MHz,$$

where n ranges from $-1,040,300$ to $1,040,300$.

Given the desirable value of n, parameters n_1, n_2 and n_3 can be derived using the theory and algorithms in [4].

VII. CONCLUSIONS

The Diophantine Frequency Synthesis (DFS) approach for fine frequency synthesis has been presented.

It is based on number theory and Diophantine equations, uses two or more basic PLLs and allows for independent choices for the output frequency step (resolution) and the phase-comparator frequencies of the PLLs.

The Diophantine Frequency Synthesis (DFS) leads to very small frequency steps, fast frequency hopping architectures with potentially very low spurs, especially in the vicinity of the carrier.

ACKNOWLEDGMENT

The author is grateful to Dr. Tom Krimigis, Dr. Marion L. Edwards, Gregory Weaver, Sheng Cheng, Wesley Millard and Christopher Haskins from the Johns Hopkins University Applied Physics Laboratory for their support, encouragement and many technical discussions on this work.

Further credit and appreciation is extended to Michael M. Driscoll from the Northrop Grumman Corporation for his encouragement and feedback on this work.

Finally, the author would like to thank his students Dimitri Loizos and Dan Zeilberger for their help with the manuscript.

The project was supported in part by the Defense Technical Information Center 06MISP7 - Johns Hopkins University Applied Physics Laboratory, Laurel, Maryland.

REFERENCES

[1] Vadim Manassewitsch, "*Frequency Synthesizers*" , third edition, John Willey & Sons, 1987.

[2] William F. Egan, "*Frequency Synthesis by Phase Lock*", second edition, John Willey & Sons, 1999.

[3] Daniel E. Flath, "*Introduction to Number Theory*", John Willey & Sons, 1989.

[4] Paul P. Sotiriadis, "Diophantine Frequency Synthesis, *IEEE Transactions on UFFC*, To appear Oct. 2006.