# An Interpolated Frequency-Hopping Spread-Spectrum Transceiver

Norman M. Filiol, *Student Member, IEEE*, Calvin Plett, *Member, IEEE*, Thomas A. D. Riley, *Member, IEEE*, and Miles A. Copeland, *Fellow, IEEE*

***Abstract*— A technique of spread-spectrum transmission, interpolated frequency-hopping (IFH), is presented. IFH employs a carrier that moves smoothly and continuously in frequency, helping to alleviate problems, such as spectral splatter and transient mismatch, which are a concern in conventional phase-locked loop (PLL)-based frequency-hopping spread-spectrum systems. In IFH, the pseudorandom hopping code is passed through a digital interpolation filter prior to controlling the synthesizer instantaneous frequency output. While such filtering is commonly used in data pulse-shaping to improve the spectral characteristics of the modulated carrier, such filtering has not been reported for IFH codes, where the frequency deviations are changing and can span several MHz. The implications of matching the transient responses of two PLL-based frequency synthesizers using this method have also not been reported. Initial simulation and laboratory measurements indicate that, for certain cases, IFH shows a 1.9 dB improvement in received IF power, has a much sharper roll-off of inband phase noise when compared to conventional hopping, and provides a phase-coherent IF after despreading. An IFH transceiver system using $\Delta$–$\Sigma$ frequency synthesis and a $\Delta$–$\Sigma$ frequency discriminator is proposed. The system would be suitable for integrated mobile radio applications in slow-fading environments.**

***Index Terms*—Digital filters, frequency hopping, phase-locked loop, spread spectrum, VLSI.**

## I. Introduction

MOBILE radio channels present system designers with many difficult design challenges. In order to combat problems, such as multipath fading and frequency allocation, transmission methods must be robust, efficient, and cost-effective. Spread-spectrum, used by the military for over 30 years, has become of interest for use in commercial mobile radio applications [1]. In spread-spectrum, a signal is spread over a frequency bandwidth that is much wider than the minimum bandwidth required to transmit the signal. This frequency-spreading offers the transmitted signal several advantages, such as resistance to jamming, resistance to interference and multipath fading, low probability of intercept, and the possibility of using code-division multiple access (CDMA) [2]. For commercial applications, resistance to jamming and low probability of intercept are of little concern.

Manuscript received June 21, 1996; revised January 10, 1997. This work was supported in part by the Telecommunications Research Institute of Ontario, Micronet (a Federal Centre of Excellence), and by the Canadian Microelectronics Corporation. This paper was recommended by Associate Editor J. Choma.

The authors are with the Department of Electronics, Carleton University, Ottawa, Ont., K1S 5B6 Canada.

Publisher Item Identifier S 1057-7130(98)00048-2.

The three basic kinds of spread-spectrum signals are frequency-hopping, time-hopping, and direct-sequence [3]. In a frequency-hopping system, the carrier frequency is “hopped” around in a pseudorandom sequence, which is known by both the transmitter and the receiver. In time-hopping, the data is transmitted in bursts that occur at pseudorandom times known by both the transmitter and the receiver. In direct-sequence spread-spectrum, the data is modulated by a high-speed pseudorandom binary bitstream, which spreads it over a frequency bandwidth that is determined by the pseudorandom code bit rate. Again, the code is known by both the transmitter and the receiver. IFH, introduced in this paper, is offered as an alternative form of frequency-hopping. IFH alleviates some of the problems inherent in phase-locked loop (PLL)-based frequency-hopping systems, while offering the same advantages and some additional benefits.

## II. Review of Frequency-Hopping

A general review of a conventional frequency-hopping approach is useful before the introduction of the alternative, which is presented in this paper. Fig. 1 shows the block diagrams of a typical frequency-hopping transmitter and receiver. In the transmitter, a pseudorandom hopping code is used to control the output frequency of a PLL-based synthesizer. The carrier is then modulated with a data signal and transmitted. An example of a hopping signal (frequency versus time) is shown in Fig. 2. In the receiver, an identical copy of the hopping pattern plus an IF-offset is subtracted from the incoming signal in the frequency domain (i.e., by mixing). If the received signal is the desired one, and the transmitter and the receiver are properly synchronized, the result of the subtraction is the FM carrier modulated with the data and centered at the IF. As the output will only contain significant energy at the desired IF if the transmitter and the receiver codes are well-correlated, frequency-hopping presents an opportunity for the use of CDMA. In frequency-hopped CDMA, each user in the system is assigned a specific frequency-hopping code. If the different codes used have low cross correlation, several users can occupy the same frequency bandwidth with only a gradual degradation in the signal-to-noise ratio (SNR) in the receiver of each user [2]. Since the carrier is regularly being hopped in frequency, frequency-hopping provides resistance to stationary interference and multipath fading. The degree of effectiveness of a particular frequency-hopping system for a given transmission channel depends on several factors, such as the hopping rate, the size of the hops, and the number of data

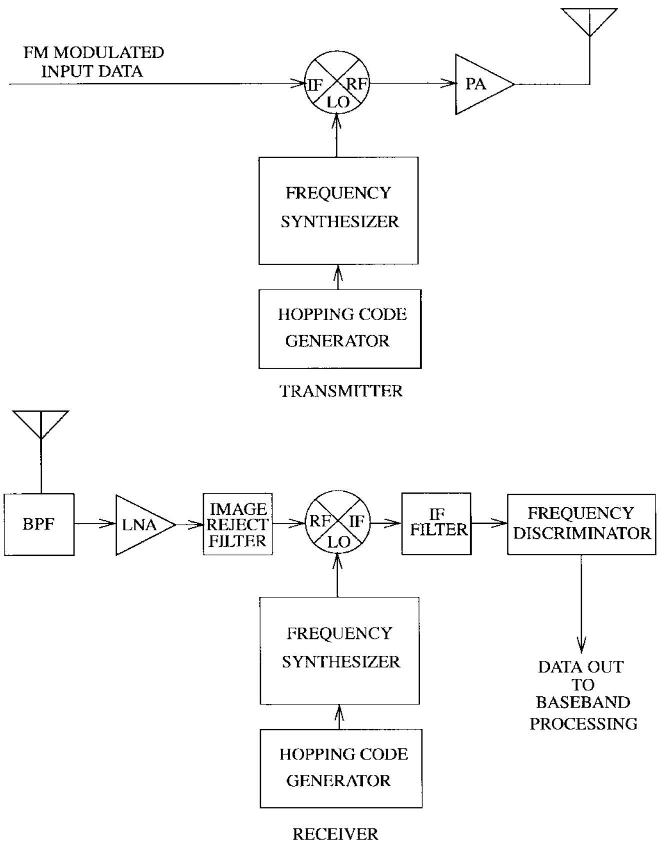



Fig. 1. A frequency-hopping spread-spectrum system.

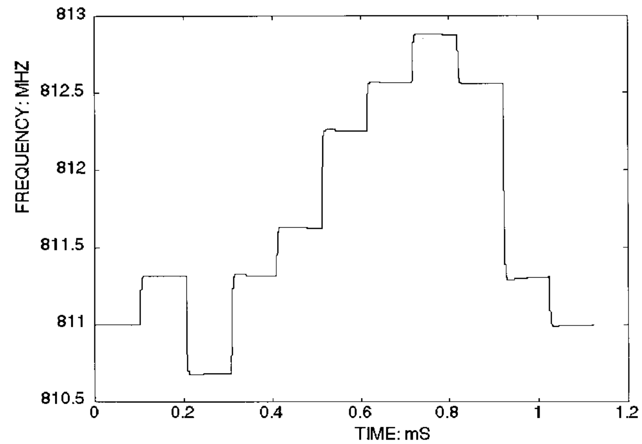



Fig. 2. Frequency-hopping: Frequency versus time.

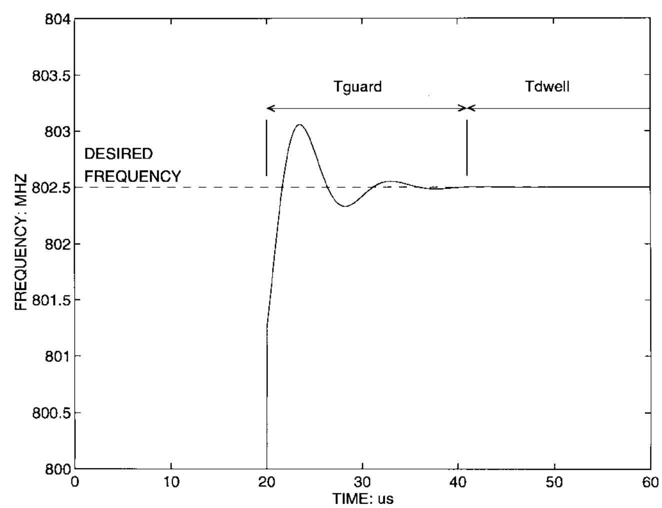



Fig. 3. Hop transient.

symbols transmitted during each hop interval [4]–[6]. If the number of data symbols transmitted during a hop interval is less than or equal to unity, the process is called fast frequency-hopping. In fast frequency-hopping, a high level of redundancy is present because there can be several hops/data symbol. This means that even if some of the hops are not properly received, due to interference or fading, there is still enough information available by using diversity combining techniques to make a correct symbol decision. The number of hops taken/symbol interval $L$ is known as the order of diversity [7]:

$$L = \frac{T_{\text{symbol}}}{T_{\text{hop}}} \tag{1}$$

where $T_{\text{symbol}}$ is the data symbol duration and $T_{\text{hop}}$ is the hop duration. The above equation is given under the assumption that the frequency separation between any two successive hops is large enough that the channel degradations encountered by the two carriers are statistically independent. Two main problems encountered in PLL-based frequency-hopping are spectral splatter, and transient mismatch between the transmit and the receive synthesizers [8]. During a hop transient, shown in Fig. 3, a simple frequency-hopping system has no control over the transmitted frequency spectrum, and many undesired frequency components are present in the output of the transmitter. This phenomenon is known as spectral splatter and results in a loss of useful transmitter energy during each hop, as well as adjacent channel interference. Additionally, if there is a mismatch in the hopping transient of the receive synthesizer from that of the transmit synthesizer, bursts of frequency error occur in the receiver IF at the hopping rate, producing an overall degradation in the receiver SNR. Some common solutions used to reduce spectral splatter and the effects of synthesizer mismatch are voltage-controlled oscillator (VCO) pretuning [8], "ping-ponging" multiple synthesizers [8], and transient hop interval dwell and guard times [9]. These methods are briefly reviewed here.

VCO pretuning makes use of a digital-to-analog converter (DAC) and a summing block placed in the synthesizer PLL between the loop filter and the VCO of both the transmitter and the receiver. When a hop occurs, the VCO control voltage is changed directly, as well as the normal loop frequency control, in a coordinated manner to reduce the effects of transient mismatch. As the DAC and summing blocks are outside of the loop, the output frequency of the synthesizer can be changed very rapidly, and the loop catches up afterwards, removing residual error.

A frequency-hopping system that "ping-pongs" two synthesizers in both the transmitter and the receiver can be designed. When one frequency is being transmitted, the second synthesizer is preset to the next hop frequency. To transmit the next frequency in the hopping sequence, the two synthesizers are exchanged (ping-ponged), eliminating the effects of PLL

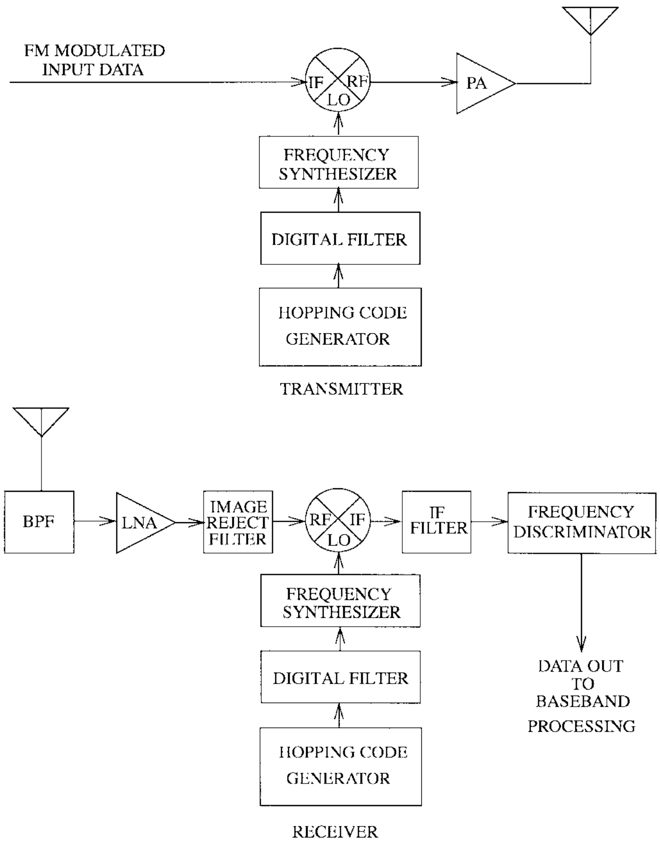



Fig. 4. An interpolated frequency-hopping spread-spectrum system.

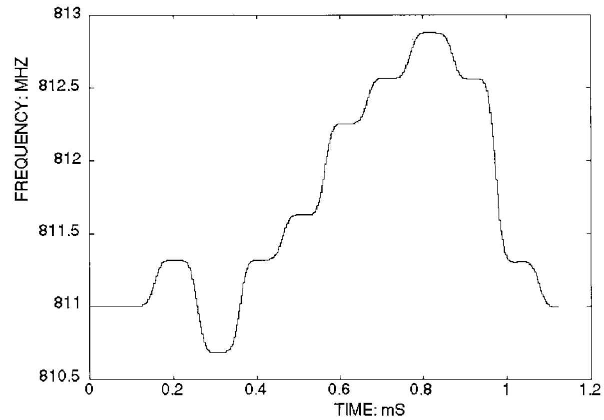



Fig. 5. Interpolated frequency-hopping: Frequency versus time.

transient error and increasing the potential hopping rate, but making phase-coherence very difficult to achieve.

Transient guard and dwell times, as shown in Fig. 3, are often employed in frequency-hopping systems to avoid the effects of IF error. During a frequency hop, the IF signal is not sampled at the receiver until the loop transients have settled out. Since the IF contains no frequency error during the sample (dwell) time, the received data is not corrupted. In some systems, the transmitter output power is turned off during the guard times in order to reduce the spectral splatter produced by the hop transients.

The above solutions, though effective, lead to higher system complexity and cost or a reduction in effective data-sampling time that increases the receiver bit error rate (BER). The alternative approach of this paper, by making the hops occur smoothly, allows the sampling to occur throughout the entire hop, and thus results in a lower BER.

## III. Interpolated Frequency-Hopping

IFH, introduced in this section, provides an alternative solution to the problems encountered in simple frequency-hopping, replacing analog complexity with digital complexity, which is a good tradeoff for very large scale integration (VLSI). A block diagram of a possible implementation of an IFH system is shown in Fig. 4 for the transmitter and the receiver. The differences between the system presented there and a standard frequency-hopping system are the inclusion of a digital filter placed between the hopping-code generator and the synthesizer-control word input in both the transmitter and the receiver, as well as differences in the digital-signal-processing (DSP) to be performed on the discriminator output in the receiver.

In the system of Fig. 4, the hopping-code generator generates a series of binary numbers that represent the desired output frequencies, which are interpolated by the digital filter and sent to the synthesizer control input. An example of the output signal (frequency versus time) for an IFH transmitter is shown in Fig. 5. The frequency scale corresponds to that in an experimental validation to follow.

A closer look at the transient error mechanism in the synthesizer will help to define IFH. Fig. 6 shows the block diagram of a fractional-$N$ PLL frequency synthesizer. The output frequency of the loop is controlled by changing $N$, the average division ratio of the divider, which consists of both an integer and a fractional part [10]. Although the divider control input is digital, for the purposes of simplicity and convention, the dynamics of the PLL are modeled as a continuous time system. Assuming that $f_{\text{in}}(s)$ is the normalized instantaneous frequency input to the synthesizer (i.e., The Fourier transform of instantaneous frequency), the instantaneous frequency output, $f_{\text{out}}(s)$, is

$$\begin{aligned} f_{\text{out}}(s) &= f_{\text{in}}(s) H_{\text{cl}}(s) \\ &= f_{\text{ref}} H_{\text{cl}}(s)[n + \delta(s)] \end{aligned} \tag{2}$$

where $f_{\text{ref}}$ is the reference frequency, $H_{\text{cl}}(s)$ is the closed-loop transfer function of the PLL from the digital control input to the VCO output, $n$ is the integer portion of the average divider division ratio, and $\delta(s)$ is the continuous time equivalent of the digital frequency control word applied to the synthesizer input.

For the frequency-hopping transceiver shown in Fig. 1, if $f_{\text{in}}(s)$ is the hopping code and no data is applied to the modulation input of the system, then for the transmitter, the output frequency is given by

$$f_{\text{Tx}}(s) = H_{\text{clTx}}(s) f_{\text{in}}(s) + f_{\text{IF}} \tag{3}$$

and for the receiver the synthesizer output frequency is given by

$$f_{\text{Rx}}(s) = H_{\text{clRx}}(s) f_{\text{in}}(s) \tag{4}$$

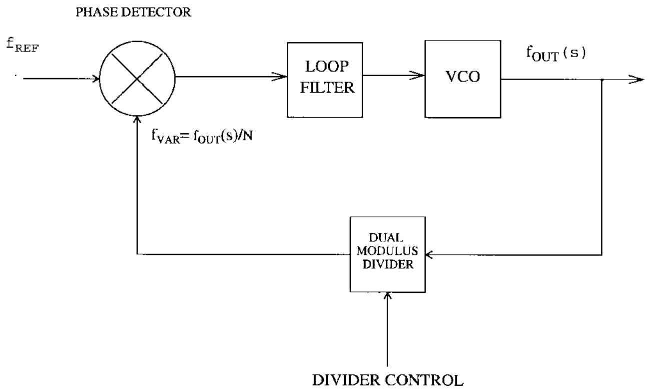



Fig. 6. Fractional-$N$ PLL frequency synthesizer.

where $H_{\mathrm{clTx}}(s)$ and $H_{\mathrm{clRx}}(s)$ are the transmitter and the receiver PLL closed-loop transfer functions, respectively, and $f_{\mathrm{IF}}$ is the IF frequency.

In the receiver, assuming perfect-code phase synchronization, the IF output, in the absence of data modulation, is given by

$$\begin{aligned}\mathrm{IF}(s) &= f_{\mathrm{Tx}}(s) - f_{\mathrm{Rx}}(s)\\ &= f_{\mathrm{in}}(s)[H_{\mathrm{clTx}}(s) - H_{\mathrm{clRx}}(s)] + f_{\mathrm{IF}}\end{aligned} \tag{5}$$

where the closed-loop transfer functions of the synthesizers, $H_{\mathrm{cl}}(s)$, are related to their open-loop transfer functions, $H_{\mathrm{ol}}(s)$, by

$$H_{\mathrm{cl}}(s) = \frac{H_{\mathrm{ol}}(s)}{1 + H_{\mathrm{ol}}(s)}. \tag{6}$$

Then, substituting (6) into (5), the IF output becomes

$$\mathrm{IF}(s) = f_{\mathrm{in}}(s)\left[\frac{H_{\mathrm{olTx}}(s)}{1 + H_{\mathrm{olTx}}(s)} - \frac{H_{\mathrm{olRx}}(s)}{1 + H_{\mathrm{olRx}}(s)}\right] + f_{\mathrm{IF}} \tag{7}$$

where $\mathrm{IF}(s)$ is the IF output signal, and $H_{\mathrm{olTX}}(s)$ and $H_{\mathrm{olRx}}(s)$ are the transmitter and receiver PLL open-loop transfer functions, respectively.

The instantaneous frequency error present in the IF output is the difference between (7) and $f_{\mathrm{IF}}$, given by

$$\mathrm{IF}_{\mathrm{error}}(s) = f_{\mathrm{in}}(s)\left[\frac{H_{\mathrm{olTx}}(s)}{1 + H_{\mathrm{olTx}}(s)} - \frac{H_{\mathrm{olRx}}(s)}{1 + H_{\mathrm{olRx}}(s)}\right]. \tag{8}$$

The above equation shows that if the open-loop transfer functions of the two synthesizers are identical, or if the open-loop gains are infinite, the IF error will be zero. The effects of PLL mismatch are smaller for input frequencies, much lower than the PLL corner frequency, and increase as the input frequency approaches the loop-corner frequency.

We will now assume that the modulated data has been linearly added (in frequency) to the transmitted signal, as shown in Fig. 1, and comes through linearly, in the absence of channel degradations and other sources of noise and distortion, to the input of the frequency discriminator. Assuming the discriminator characteristic is perfectly linear (taking into account discriminator nonlinearities would only confuse the issue), (8) represents a degradation of the received IF introduced by incomplete despreading of the pseudorandom hopping sequence $f_{\mathrm{in}}(s)$, due to synthesizer mismatch. The best-case SNR of the received baseband signal is then given by (9), shown at the bottom of the page, where $f_{\mathrm{data}}(s)$ is the transmitted data, $k_d$ is the discriminator characteristic in V/Hz, and BW is the frequency bandwidth of interest. By filtering the hopping code prior to the synthesizer input, it is possible to reduce the high-frequency components of the transmitted signal, and thus, to minimize the PLL mismatch error.

Assuming that the interpolation filter, being digital, is identical in both the transmitter and the receiver, and following the same derivation as before, the IF error for IFH is given by

$$\begin{aligned}\mathrm{IF}_{\mathrm{error}}(s) &= f_{\mathrm{in}}(s)F(s)\\ &\cdot\left[\frac{H_{\mathrm{olTx}}(s)}{1 + H_{\mathrm{olTx}}(s)} - \frac{H_{\mathrm{olRx}}(s)}{1 + H_{\mathrm{olRx}}(s)}\right]\end{aligned} \tag{10}$$

where $F(s)$ is the continuous time equivalent transfer function of the digital interpolation filter. Therefore, using the same assumptions as for frequency-hopping, and including the effects of the interpolation filter transfer function $F(j\omega)$, we can derive (11), shown at the bottom of the page.

The end result of filtering the hopping code prior to the synthesizer control input is the effective low-pass filtering of baseband receiver distortion generated by PLL mismatch, increasing the received SNR. The degenerate case, where $F(j\omega) = 1$, was given by (9). Thus in IFH the carrier signal moves continuously and smoothly in frequency, and there

$$\mathrm{SNR} = \frac{k_d^2 \int_{0^+}^{2\pi \mathrm{BW}} [|f_{\mathrm{data}}(j\omega)|]^2 \, d\omega}{k_d^2 \int_{0^+}^{2\pi \mathrm{BW}} \left[(|f_{\mathrm{in}}(j\omega)|)^2 \left(\left|\frac{H_{\mathrm{olTx}}(j\omega)}{1 + H_{\mathrm{olTx}}(j\omega)} - \frac{H_{\mathrm{olRx}}(j\omega)}{1 + H_{\mathrm{olRx}}(j\omega)}\right|\right)^2\right] d\omega} \tag{9}$$

$$\mathrm{SNR} = \frac{k_d^2 \int_{0^+}^{2\pi \mathrm{BW}} [|f_{\mathrm{data}}(j\omega)|]^2 \, d\omega}{k_d^2 \int_{o^+}^{2\pi \mathrm{BW}} \left[(|f_{\mathrm{in}}(j\omega)|)^2 (|F(j\omega)|)^2 \left(\left|\frac{H_{\mathrm{olTx}}(j\omega)}{1 + H_{\mathrm{olTx}}(j\omega)} - \frac{H_{\mathrm{olRx}}(j\omega)}{1 + H_{\mathrm{olRx}}(j\omega)}\right|\right)^2\right] d\omega} \tag{11}$$

are no bursts of frequency error present in the IF output at the code rate. The PLL synthesizer must continuously track the input frequency, resulting in low levels of frequency error being translated to the IF output. Provided that the frequency error is small enough to maintain an adequate SNR in the receiver, valid data is available at all times, improving performance if the receiver makes use of the whole symbol period in its processing. The smooth carrier movement may also help to reduce the complexity and improve the performance of the hardware required to synchronize the received and the locally-generated codes, since the code phase information is available to the receiver at all times and not just during the hop intervals. A more thorough analysis of the properties of IFH transmission, including the effects of frequency collisions between users in CDMA applications and receiver-code synchronization, would be required for future applications.

A rigorous theoretical analysis of the improvements achieved in the spectral purity of the despread IF and the BER in the receiver, through the use of IFH, would probably be impractical, due to the large number of variables involved and the simplifying assumptions that would have to be made. Unlike standard frequency-hopping, the carrier in IFH is continuously moving and the assumption that carrier movement during frequency changes can be ignored, which is generally used in frequency-hopping analysis, is no longer valid, making the analysis much more difficult. Instead, the tradeoffs involved in the design of an IFH transmitter are identified in the following paragraphs, and a case study, which makes use of computer simulations and experimental results obtained from a hardware implementation of an IFH system, is presented in order to validate the claims made.

As indicated in (11), the frequency error during interpolated hopping can be minimized by lowering the frequency content of the hop code. This can be done by lowering the hopping rate relative to the loop bandwidth, and through digital interpolation. The improvement achieved through filtering the hopping code will diminish as the code rate approaches or exceeds the loop bandwidth. Increasing the loop bandwidth of the synthesizers, relative to the hopping-code rate, and carefully matching the loop filters will help. The size of the frequency steps will also have an effect on the transmission properties. Using large frequency deviations and pulse-shaping for carrier FM data modulation will help to increase the baseband SNR.

The smoothness of the step response of the digital filter is important, as it will partially determine the transient response of the synthesizer to a frequency step. While a narrower filter bandwidth, relative to the PLL corner frequency, provides better transient matching between the transmit and the receive synthesizers, it may also reduce the amount of frequency-spreading achieved, depending on the hopping rate. This presents the designer with a tradeoff between transient matching and frequency diversity. That is, a constant relationship between the hopping rate and the filter bandwidth must be maintained in order to achieve frequency diversity. One method of maintaining the original frequency deviations of the hopping code is to use a filter with an impulse response, which has a zero crossing at the period of the hopping rate, such as a raised-cosine pulse. While using a raised-cosine pulse would guarantee that every frequency in the hopping code would be achieved by the synthesizer, computer simulations performed show that a Gaussian pulse-shape with a BT, (filter bandwidth divided by data rate) ranging from 0.3–0.5, can provide better spectral characteristics, while maintaining most of the desired hop frequencies. Optimal pulse-shapes for use in IFH are the subject of ongoing research, and many possibilities exist.

IFH should have a diversity order that is at least as high as that obtained through the use of a comparable conventional frequency-hopping system. Therefore, (1) can be used to obtain a rough estimate of the diversity order that can be achieved for a given IFH system. It should be noted that for IFH, because the carrier is continuously moving in frequency, it is difficult to define a specific order of diversity as the transmission can no longer be discretely separated into resolvable paths.

## IV. Simulation and Laboratory Results

An initial MATLAB simulation of simple hopping and the equivalent IFH transmission was performed for ten frequency steps in the frequency range of the experiment, assuming a synthesizer loop filter component mismatch of 10% (this is a reasonable assumption if typical components are used) and a noiseless system. Here, "simple" means no guard times, "ping-ponging," etc. The simulations were performed using an array of points, which represented the digitally filtered frequency deviations. This array was then filtered by a mathematical representation of the PLL closed-loop transfer function. The loop filter transfer function used was an integrator with phase-lead correction, and the closed loop PLL bandwidth was 100 kHz. An array was created for both the transmitter and the receiver outputs, and a mismatch was introduced between the two PLL transfer functions. Despreading was achieved by subtracting the filtered transmit and receive arrays. The remainder represents the frequency error present, due to incomplete despreading.

The interpolation filter used for code-smoothing was a Gaussian low-pass filter with a corner frequency of 3.25 kHz ($BT = 0.325$) and double-precision tap coefficients. This filter was chosen for the purpose of demonstrating a proof of concept. While it is not necessarily the optimal filter, computer simulations were performed to ensure that it improved matching between the two synthesizers and maintained most of the original frequency-spreading desired. Also, Gaussian pulses are easy to generate and their properties are well known.

The simulated frequency versus time plot for the transmitted signals are shown in Fig. 7 for the two alternatives. The increased smoothness of the carrier movement for IFH transmission is clearly seen from the plot. Fig. 8 is a plot of the simulated instantaneous IF frequency error, assuming ideal mixing, for the frequency steps shown in Fig. 7. As predicted, the IF frequency error present in simple frequency-hopping transmission takes the form of sharp bursts occurring at the 10 kHz code rate. For IFH, these error bursts are replaced by a low-level, almost constant frequency error, as Fig. 8 demonstrates.

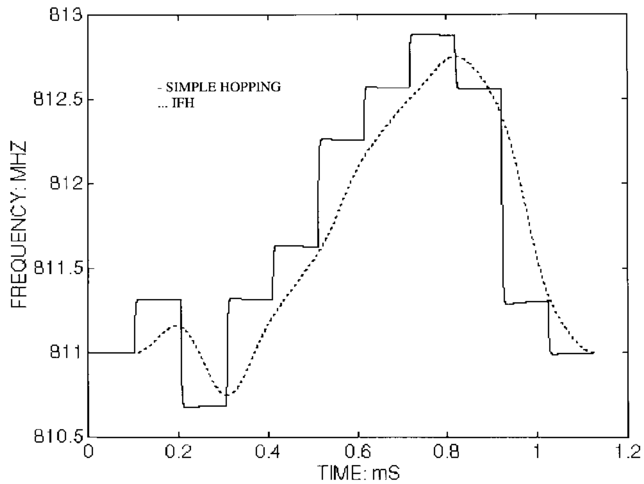



Fig. 7. Simulated frequency versus time display for both hopping methods.

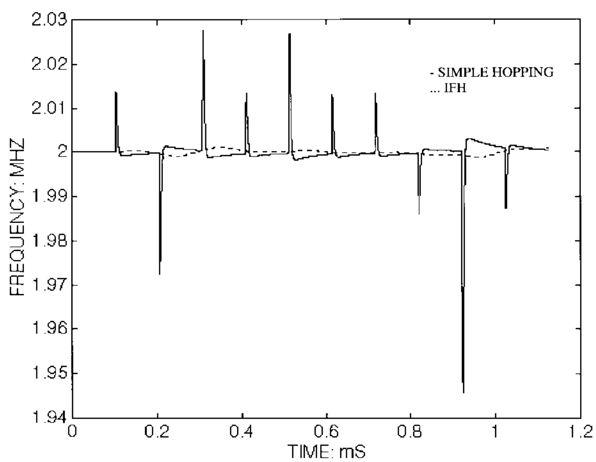



Fig. 8. Simulated IF error for both hopping methods.

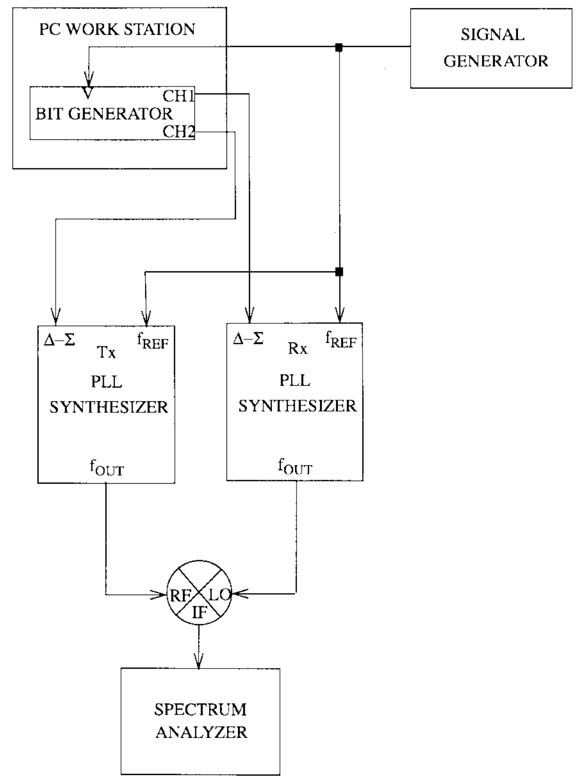



Fig. 9. Laboratory test setup.

Laboratory measurements of the relative IF quality obtained through simple and IFH transmission were obtained using the setup shown in Fig. 9. The fractional-$N$ synthesizers (Tx and Rx) were controlled by single-bit bitstreams, which were noise-shaped using $\Delta$–$\Sigma$ techniques [10], [11]. Note that $f_{\rm in}(s)$, as defined in (7), now contains $\Delta$–$\Sigma$ quantization noise as well as the hopping sequence. Hopping-code generation, code interpolation, and $\Delta$–$\Sigma$ quantization were performed in nonreal time using software running on a PC workstation, and fed to the system in real time using the bit generator shown.

The fractional-$N$ synthesizers were implemented at the printed circuit board level and incorporated with a modulus extension circuit, allowing division ratios between 80 and 85 to be used, for a reference frequency of 10 MHz. The center frequencies of the transmitter and the receiver synthesizers were chosen to be 811 and 809 MHz, respectively, with a spreading bandwidth of 5 MHz. This allowed an IF center frequency of 2 MHz to be chosen in order to take advantage of the narrow resolution bandwidth of the 40 MHz bandwidth spectrum analyzer (HP3585A) used, although the implemented transceiver would have a much higher IF frequency to resolve problems, such as image rejection.

An array of 1024 frequency points uniformly distributed between $\pm 625$ kHz was created and superimposed on a triangular waveform with a repetition time of 102 mS. Individual hop sizes varied from 9.765 kHz to 1.250 MHz and the code rate was 10 kHz. Although the triangular waveform used is a nonstandard frequency-hopping code, it provided a uniform frequency distribution, and substantial simplification to the interpolation filter. In order to allow the measured and simulated results to be compared, the interpolation filter used for code-smoothing was again a Gaussian low-pass filter with a corner frequency of 3.25 kHz.

Two stored sequences of $2^{20}$-bits obtained from the simulation of the $\Delta$–$\Sigma$ modulator were used in real time to control the multimodulus dividers of the transmit and the receive synthesizer boards. A plot of the measured frequency versus time characteristic for IFH transmission for the triangular-coding waveform was obtained using a time-interval analyzer and is shown in Fig. 10. The time-interval analyzer does not have the measurement resolution required to view individual frequency hops, and because of this, the plot for IFH transmission appears to be identical to the one obtained for simple frequency-hopping, which therefore is not shown here.

A plot of the measured IF spectrum for transmission using a stationary carrier is shown in Fig. 11, and is included here for reference purposes to give an indication of the spectral purity of the synthesizers. The absolute power level of the IF center frequency is 0.1 dBm. A plot of the measured IF spectrum, including both simple frequency-hopping and IFH, is shown in Fig. 12.

For simple frequency-hopping it can be seen that the IF in Fig. 12 contains a high-level of phase noise and a series of

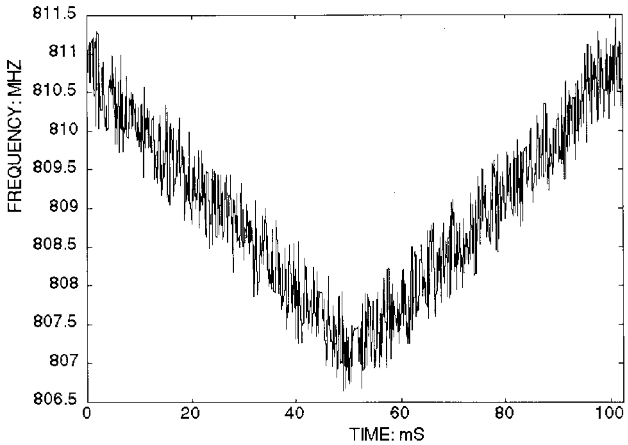



Fig. 10. Measured frequency sequence for interpolated frequency-hopping.

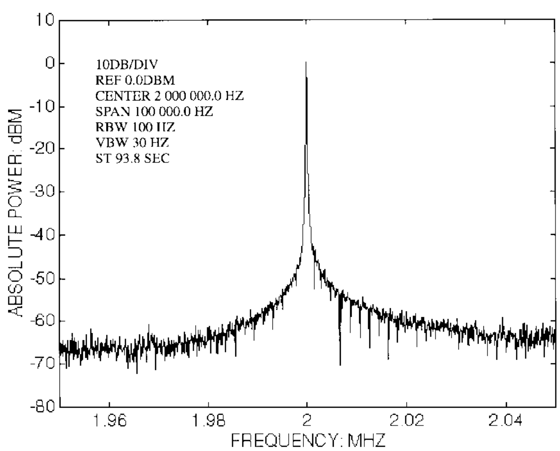



Fig. 11. Measured IF spectrum for stationary carrier transmission.

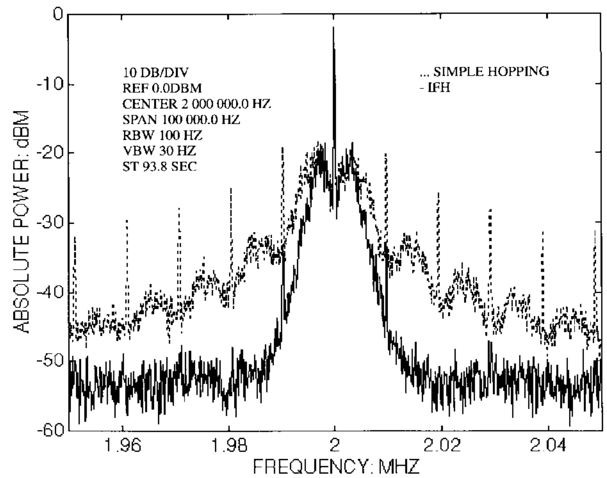



Fig. 12. Measured IF spectrum for frequency-hopping and interpolated frequency-hopping.

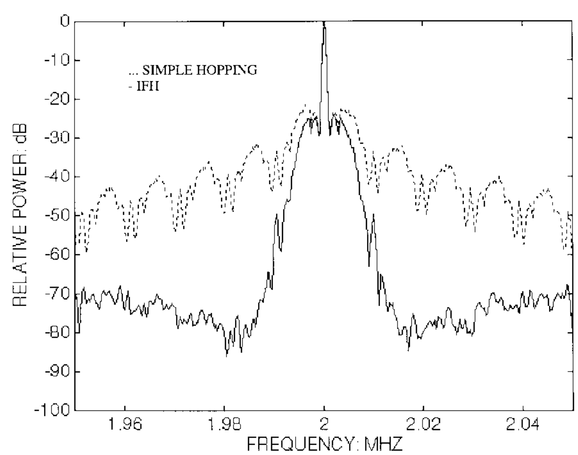



Fig. 13. Simulated IF spectrum for both hopping methods (10% PLL component mismatch).

nulls and spikes occurring regularly at 10 kHz spacings from the center frequency. If the transmitter and receiver synthesizers were identical, this distortion would not be present in the IF. However, in a real system, identical transient response between two synthesizers is impossible to achieve. Because of this, an attenuated version of the hopping code is present in the receiver IF. The hopping code is slightly unbalanced (i.e., contains an unequal number of ones and zeros), resulting in the spikes present in the IF spectrum. If the code length was infinite, the hopping would be nondeterministic in nature, and the IF spectrum would have a continuous $\sin x/x$ form. However, the code used in the laboratory has a finite length and repeats every 0.1 s. The simple hopping IF spectrum is actually made up of discrete frequency components present at 10 Hz spacings, the repetition rate of the hopping code. These components are not visible in Fig. 12 because they are narrower than the resolution bandwidth of the spectrum analyzer. The absolute power measured for simple frequency-hopping at the IF center frequency is $-3.6$ dBm. This shows that about 3.7 dB of transmitter output power is lost due to spectral splatter, when compared with the stationary carrier case shown in Fig. 11.

The plot of the IF frequency spectrum measured for IFH in Fig. 12 shows a sharp reduction in phase noise when compared to simple hopping, and the nulls and spikes are almost completely eliminated. The absolute power measured at the IF center frequency is $-1.7$ dBm, showing a 1.9 dB reduction in spectral splatter over simple frequency-hopping. Fig. 12 can be compared to Figs. 13 and 14, which show the simulated IF frequency spectrums for both simple frequency-hopping and IFH, obtained in MATLAB, for the same code sequence used in the laboratory measurements. Fig. 13 shows the simulated IF spectrum for both simple hopping and IFH when $\Delta$–$\Sigma$ quantization noise and a 10% component mismatch between the loop filters is considered. The results obtained in the laboratory are almost identical to the simulated results except for an increase in the noise floor and other small discrepancies. Because it is almost impossible to fully characterize the mismatch between the two PLL boards used in the laboratory, the simulation results are not completely identical to those obtained in the laboratory.

Fig. 14 shows the simulated IF spectrum for both simple hopping and IFH when the natural frequency of the loop filter in one of the synthesizers is reduced by a factor of four, compared to the other. It can be seen that the quality of the IF spectrum is degraded when compared to the one shown in Fig. 13, particularly for close-in phase noise (i.e., within 50

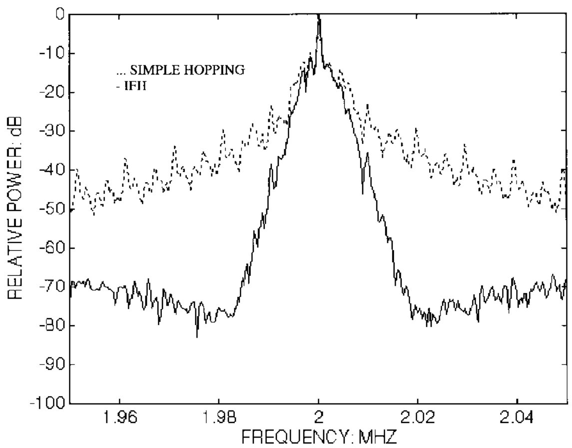



Fig. 14. Simulated IF spectrum (loop natural frequency mismatch of $4\times$).

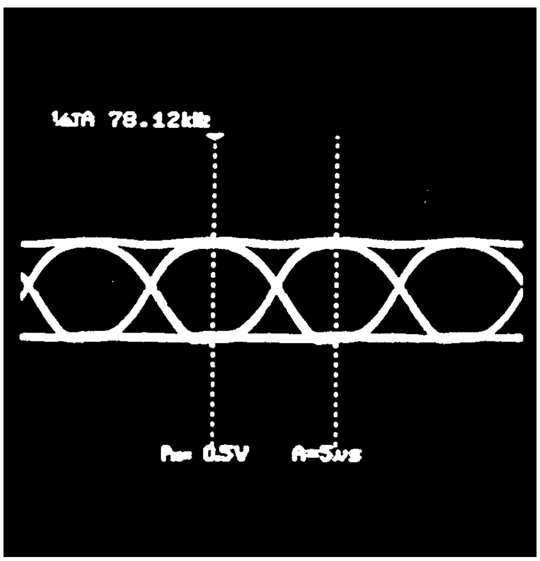


Fig. 15. Demodulated FSK eye diagram.

kHz of the IF). This would seem to indicate that except for normal component variations, the two synthesizers used in the laboratory are fairly well-matched. In order for IFH to work well, the synthesizers used should be carefully matched.

The system has been initially tested with FSK data modulation at a symbol rate of 78 kb/s, frequency deviations of $\pm500$ kHz, and Gaussian pulse-shaping. This was done with the same laboratory setup as in Fig. 9, with pseudorandom data generated in software. The received eye diagram is shown in Fig. 15, and indicates that transmission of data using IFH is viable. Experimental work has also been presented, using various hopping rates and frequency deviations, which shows that receiver BER's can be lowered through the use of IFH compared to standard frequency-hopping [12].

## V. Proposed System

A proposed architecture for an RF transceiver that employs IFH transmission is shown in Fig. 16. The block diagram is similar to a typical FM system with some additional components included for code generation and filtering, as well as for phase-synchronization of the received and the locally-generated codes. The transmitted carrier signal is frequency-shift keying (FSK) modulated. The data is filtered prior to its addition to the IFH carrier in order to improve its spectral characteristics. Although baseband processing is not covered in this paper, it is well-known that the use of bandwidth efficient data modulation improves the performance of a frequency-hopping system [13].

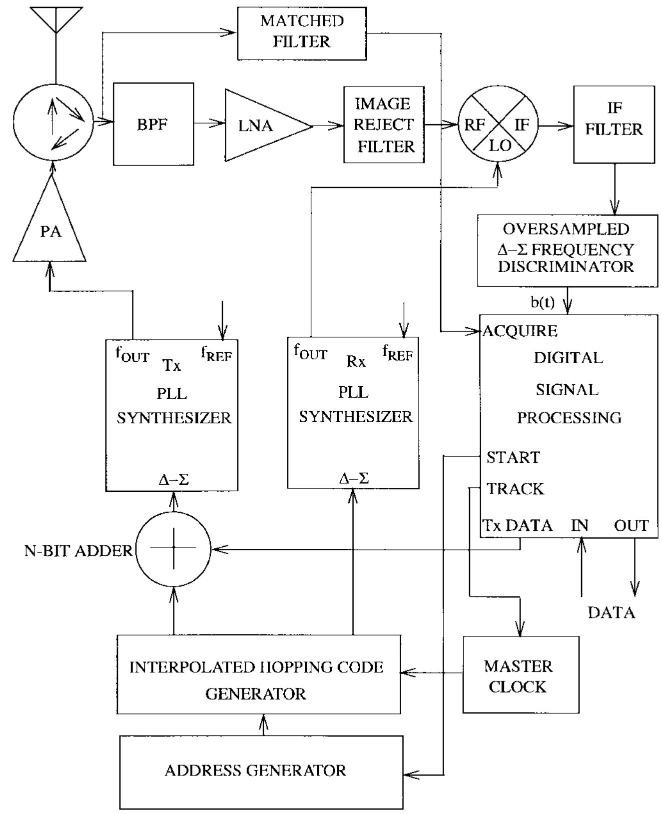



Fig. 16. An interpolated frequency-hopping spread-spectrum transceiver system.

In the receiver side of Fig. 16, the received signal is down-converted to an IF by mixing it with a copy of the transmitted code produced by the Rx synthesizer. In order to obtain the full benefit of IFH, it is necessary to process the IF signal over the whole symbol period, implying a high-sampling rate or continuous-time integration. A frequency discriminator has recently been demonstrated [11], [14], [15], which can accept an FM signal at a relatively high IF and produce a 1-bit frequency-to-digital conversion at high-oversampling rates through $\Delta$–$\Sigma$ techniques [16]. The discriminator is well-suited for data signals that contain large frequency deviations, such as might be used in IFH, and is indicated as a block in Fig. 16. A fully integrated $\Delta$–$\Sigma$ frequency discriminator has been tested [15] and PCB implementations also exist [11]. The output 1-bit digital bitstream, $b(t)$, has an average density of ones which is proportional to the frequency deviation of the input signal. This represents the baseband data and channel phase noise as well as an error signal directly related to the phase difference between the received and local codes. The DSP block of the transceiver is used to extract the error signal and to decimate the output of the discriminator. The error signal is then used as feedback to control the phase of the PN-code generator clock for fine-time code tracking. One possible method for acquisition of the IFH signal is to place a matched filter in parallel with the input signal path, as shown in Fig. 16. This filter produces a correlation peak when a short sequence of hops, part of the larger sequence transmitted, is detected. In order to use an analog matched filter for acquisition, the carrier would probably first have to be down converted to an IF frequency, since high-frequency matched filters are difficult to design. The resulting signal is used by the DSP block to set the local-code generator to the starting ROM address. This method of coarse synchronization provides the system with the fast acquisition times that are required in mobile radio applications. Since in the transmitter, the data clock is derived from the clock that drives the code-interpolation filter, once code phase-synchronization is achieved in the receiver, data-timing is also recovered.

The $\Delta$–$\Sigma$ frequency synthesizer used in both the transmitter and the receiver side of Fig. 16 is a fractional-$N$ PLL synthesizer similar in construction to the one shown in Fig. 6, with the addition of a digital $\Delta$–$\Sigma$ modulator. The input to the modulator is a $K$-bit wide digital control word, which represents the desired instantaneous frequency output of the synthesizer. The modulator converts this into a binary bitstream, whose average density of ones represents the required division ratio. High-frequency resolution and low-phase noise levels are obtained through oversampling, the noise-shaping properties of the $\Delta$–$\Sigma$ modulator [16], and additional filtering by the PLL.

The receive synthesizer in Fig. 16 would be modulated with the interpolated hopping code, at the local oscillator (LO) frequency. The transmit synthesizer in Fig. 16 would be modulated with the IFH code as well as the transmitted data, at the RF. It would be possible, in an alternate realization of the transceiver, to share a single synthesizer for both the Tx and the Rx function, modulated only with the interpolated hopping code. A mixer would then be required in the Tx path. An experimental demonstration of the transmitter path has recently been presented [17], for GMSK signals, which gives an efficient embodiment of the digital interpolation filter assumed for the hopping-code generator.

## VI. Conclusions

In order to meet the increasing demands on system designers imposed by mobile radio environments, use of spread-spectrum transmission methods for commercial applications is being considered. One of the common forms of spread-spectrum, PLL-based frequency-hopping, performs well in such environments but suffers from problems, such as spectral splatter and IF error produced by PLL synthesizer mismatch, which reduce system efficiency and performance. These problems are usually controlled using VCO pretuning, multiple synthesizers, or transient guard and dwell times. In this paper, IFH has been presented as another solution to the problems of reducing spectral splatter and IF error, as well as providing the receiver with a continuous stream of data that can be sampled over the entire symbol interval by oversampling techniques, such as the $\Delta$–$\Sigma$ frequency discriminator discussed, giving an opportunity for improvements in the received IF power.

Interpolation of the hopping code is used to reduce the high-frequency content of the hopping steps, thereby reducing the PLL phase error. The result of this filtering is a carrier that moves smoothly and continuously in frequency. Preliminary laboratory tests show that IFH reduces spectral splatter by almost 2 dB, and greatly reduces phase noise in the receiver IF, when compared to simple frequency-hopping.

The possibility may exist to use higher code rates for a given synthesizer with IFH than could be used with simple frequency-hopping. This is because IFH produces constant phase error, which increases with the code rate. As the code rate increases, the SNR of the receiver will decrease proportionally, and eventually, reception will become impossible. In simple frequency-hopping, as code rates begin to approach the loop settling time, dwell times become unusable and reception also becomes impossible. This transition should be more gradual with IFH than hopping. There may exist a point where reception with IFH is still possible when reception with conventional frequency-hopping is not. Further research is needed to determine if this is the case.

By shaping the frequency-hopping through the use of a digital interpolation filter, we have increased the digital complexity of the system, but have reduced the analog problems. As VLSI technologies continue to progress, this kind of tradeoff becomes increasingly favorable to the approach presented here, particularly for applications requiring low-power consumption and small size.

## References


[1] R. A. Scholtz, "The origins of spread-spectrum communications," *IEEE Trans. Commun.*, vol. COM-30, pp. 822–854, May 1982.

[2] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of spread-spectrum communications—A tutorial," *IEEE Trans. Commun.*, vol. COM-30, pp. 855–884, May 1982.

[3] M. P. Ristenbatt and J. L. Daws, Jr., "Performance criteria for spread-spectrum communications," *IEEE Trans. Commun.*, vol. COM-25, pp. 756–762, Aug. 1977.

[4] D. J. Torrieri, *Principles of Secure Communication Systems*. Dedham, MA: Artech House, 1985.

[5] D. N. Romalo and Q. Wang, "Design and implementation of a frequency-hopping modem," in *Canadian Conf. Electrical Computer Engineering*, Ottawa, Ont., Canada, Sept. 1990.

[6] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1983.

[7] J. S. Bird and E. B. Felstead, "Antijam performance of fast frequency-hopped M-ary NCFSK—An overview," *IEEE J. Select. Areas Commun.*, vol. SAC-4, pp. 216–232, Mar. 1986.

[8] J. R. Alexovich and R. M. Gagliardi, "Effect of PLL synthesizer in FSK frequency-hopped communications," *IEEE Trans. Commun.*, vol. 37, pp. 268–275, Mar. 1989.

[9] L. E. Miller, J. S. Lee, and D. J. Torrieri, "Analysis of transients affecting the selection of hopping rate for FH/CPFSK systems," in *IEEE MILCOM '91*, pp. 0590–0600.

[10] T. A. Riley, M. A. Copeland, and T. A. Kwasniewski, "Delta–sigma modulation in fractional-$N$ frequency synthesis," *IEEE J. Solid-State Circuits*, vol. 28, pp. 553–559, May 1993.

[11] T. P. Kenny, "Components and architectures for an integrated wideband FM transceiver," M.Eng. thesis, Carleton Univ., Ottawa, Ont., Canada, 1994.

[12] N. M. Filiol, C. Plett, T. A. D. Riley, and M. A. Copeland, "Bit-error rate measurements for a high frequency interpolated frequency-hopping spread-spectrum system," in *IEEE Int. Symp. Circuits Syst.*, Seattle, WA, Apr. 1995, vol. 1, pp. 97–100.

[13] Y. M. Lam and P. H. Wittke, "Frequency-hopped spread-spectrum transmission with band-efficient modulations and simplified noncoherent

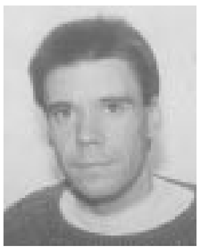
sequence estimation," *IEEE Trans. Commun.*, vol. 38, pp. 2184–2195, Dec. 1990.

[14] R. D. Beards and M. A. Copeland, "An oversampling delta–sigma frequency discriminator," *IEEE Trans. Circuits Syst.*, vol. 41, no. 1, pp. 26–32, Jan. 1994.

[15] D. Beards and M. A. Copeland, "An oversampling delta–sigma frequency discriminator for digital radio," in *1993 CCVLSI*, Banff, Alta., Canada, Nov. 1993.

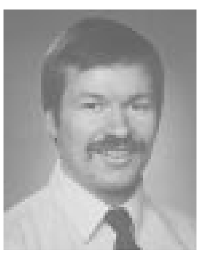
[16] J. C. Candy and G. C. Temes, "Oversampling methods for A/D and D/A conversion," in *Oversampling Delta–Sigma Data Converters*. New York: IEEE Press, 1991.

[17] T. A. D. Riley and M. A. Copeland, "A simplified continuous phase modulator technique," *IEEE Trans. Circuits Syst.*, vol. 41, May 1994.



**Norman M. Filiol** (S'94) received the B.Eng. degree in electrical engineering in 1992 and the M.Eng. degree in electrical engineering in 1994, both from Carleton University, Ottawa, Ont., Canada. He is currently a Ph.D. candidate at Carleton University, in a cooperative research agreement with Nortel Semiconductor.

From 1989 to 1992, he spent three consecutive summer terms at Transport Canada working on navigational aids. His research interests are in the area of mixed signal VLSI design for mobile radio. Ongoing research is making use of CMOS, bipolar, and BiCMOS technologies to develop low-power, highly integrated, building blocks for frequency synthesis and frequency discrimination. He is partially funded by Nortel Semiconductor and by Communications and Information Technology Ontario (CITO), a provincially funded university-based research organization.



**Calvin Plett** (S'82–M'82) received the B.A.Sc. degree in electrical engineering from the University of Waterloo, Waterloo, Ont., Canada, in 1982, and the M.Eng. and Ph.D. degrees from Carleton University, Ottawa, Ont., Canada, in 1986 and 1991, respectively.

From 1982 to 1984, he worked at Bell-Northern Research. In 1989, he joined the Department of Electronics, Carleton University, where he is now an Assistant Professor. Since 1995, he has done consulting work for Nortel Semiconductor in the area of RF integrated circuit design. His research interests are in the area of analog integrated circuit design including filters, radio-frequency front-end components, and communications applications.

**Thomas A. D. Riley** (S'81–M'82) received the B.E.Sc. degree in electrical engineering from the University of Western Ontario, London, Ont., Canada, in 1982.

From 1982 until 1986, he worked in the power products division of Phillips Cables Ltd., Brockville, Ont., Canada. His M.Eng. research was performed at Caleton University and Mitel Semiconductor on modeling Sigma–Delta modulator architectures for primary rate ISDN. In 1988, he joined the VLSI in Communications Group at Carleton University as a Research Engineer. He is currently pursuing research in IF A/D converters at the University of Oulu, Finland.

**Miles A. Copeland** (M'65–SM'85–F'89) received the B.Sc. degree in electrical engineering from the University of Manitoba, Winnipeg, Man., Canada, in 1957, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, Toronto, Ont., Canada, in 1962 and 1965, respectively.

He has been a Professor in the Department of Electronics, Carleton University, for 31 years, and a Consultant with Nortel Technology Ltd. He is now officially retired, but maintaining an office and the honorary title of Distinguished Research Professor. He is active in research and graduate student supervision, and is presently a Co-Investigator at Carleton University in the VLSI for Communications Group. As well, some of his research is supported through cooperative graduate student thesis projects with industry.