Implementation of a Real-Time Frequency-Selective RF Channel Simulator Using a Hybrid DSP-FPGA Architecture

Mark A. Wickert, *Member, IEEE*, and Jeff Papenfuss, *Member, IEEE*

***Abstract*—A low cost frequency-selective RF channel simulator architecture is explored in this paper. The technique of quadrature amplitude modulation (QAM) by independent low-pass filtered white Gaussian noise sources forms a rational function approximation (RFA) to the desired Doppler spectrum for flat Rayleigh fading. To simulate frequency-selective fading, this QAM/RFA architecture may be extended by combining delayed outputs from multiple flat fading generators. In this paper, the noise shaping filter considered is in the form of an infinite-impulse-response digital filter followed by an interpolator (upsampler) using linear interpolation. The performance requirements are those in the standard channel simulator section of TIA IS-55-A. The system is implemented almost entirely in the digital domain by use of IF sampling, with the signal processing performed in a high-end floating-point digital signal processor and a field-programmable gate array. The theoretical performance of the simulator is studied with respect to the TIA standard, and limitations of the hardware prototype are identified. A system capable of simulating 12 delay taps, with a processing bandwidth of 5 MHz, can be built at about one-tenth the cost of commercially available channel simulators of comparable performance.**

***Index Terms*—Mobile radio channel, Rayleigh fading channel, simulation techniques.**

I. Introduction

DESIGNING a modern wireless communication system is a formidable task. Today's users demand good voice quality, have a low tolerance for busy signals or dropped connections, and desire error-free high-speed data transmission. For a system to efficiently meet these requirements, it must perform well in many different environments where the radio propagation characteristics vary considerably. Ensuring a product's performance requires not only analysis and simulation, but also prototyping and testing. Unfortunately, field testing a wireless product in all of the possible radio environments is cost prohibitive and time consuming. A much more practical approach is to use a real-time channel simulator that may be configured to emulate the various radio propagation characteristics encountered in the real world.

In the classic book edited by Jakes [1], several techniques for simulating the flat fading Rayleigh statistics of a mobile radio channel are presented. The two most frequently cited techniques for generating Rayleigh flat fading in a channel simulator are a discrete approximation to the desired power spectrum (often referred to in the literature as the Jakes method [1], [2]) and quadrature amplitude modulation (QAM) by independent low-pass filtered white Gaussian noise (WGN) sources [1], [3]–[5]. Since the second method uses realizable low-pass filters to shape the spectrum, the transfer function is of rational form.

In this paper we discuss a digital-signal-processor (DSP)-based channel simulator that uses a rational function approximation (RFA) by employing an infinite impulse response (IIR) digital filter followed by an upsampler and interpolation filter. The computational burden of this particular implementation is low when compared to the Jakes method, yet statistical performance is good. The general performance goals in this paper are the requirements called out in the standard channel simulator section of TIA IS-55-A [6].

The available channel simulators on the market today are complex and costly (from $24 000 to $500 000 [7]), often requiring several circuit cards and multiple processors. Recent increases in the performance of DSPs and field-programmable gate arrays (FPGAs) suggest the possibility of greatly reducing the cost and complexity required in implementing a channel simulator. This paper presents a summary of the background theory, DSP design issues, prototype implementation, and test of a low-cost real-time frequency-selective RF channel simulator.

The following sections of this paper will first overview the basic theory of mobile radio multipath propagation and the RFA simulation technique. Secondly, the real-time DSP architecture will be described, which results in a prototype simulator. In Section III, performance results are given, both in terms of theory and experimental results from the prototype. Lastly, conclusions and suggestions for further investigation and implementation improvements are given.

II. Background

Several methods exist in practice for simulating the RF channel environment. All of the methods are based on observations and measurements of signal propagation in the end-user environment. The most computationally efficient method, and the focus of this paper, uses QAM to apply the RFA fading to an RF carrier signal (Fig. 1) [1], [2].

The RFA filtering of the noise sources approximates the desired Doppler spectrum. The architecture of Fig. 1 produces flat fading. To simulate frequency-selective fading, this architecture

Manuscript received December 15, 2000.

M. A. Wickert is with the Department of Electrical and Computer Engineering, University of Colorado at Colorado Springs, Colorado Springs, CO 80933 USA.

J. Papenfuss is with the Wireless Technology Group, Xircom, Colorado Springs, CO 80907 USA.

Publisher Item Identifier S 0018-9480(01)06151-8.

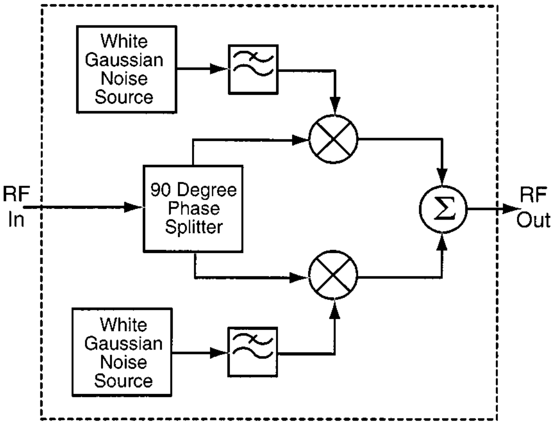

Fig. 1. QAM method for flat channel fading.

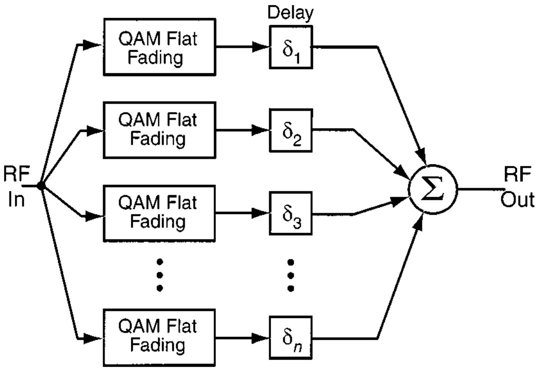

Fig. 2. Frequency-selective QAM method.

may be extended by combining delayed outputs from multiple flat fading generators (Fig. 2).

Examples of the QAM method are abundant in the literature. They range in complexity from a fully analog flat fading simulator to an IF sampled frequency-selective channel simulator described here. All of the frequency-selective implementations require complex hardware consisting of several circuit cards with multiple processors. For example, the NoiseCom MP-2500 Multipath Fading Emulator [8] uses the QAM method with up to 12 delay paths. It consists of 11 circuit boards, not including the RF circuitry, cooling fans, or external computer interface [8].

A. Theory

The theory of mobile radio multipath propagation is developed in [1]. Here, it is assumed that the signal arriving at the mobile receiver results from the linear superposition of plane waves of random phase. Since the receiver is in motion, each plane-wave component has associated with it a Doppler frequency shift. Here, we assume that the transmitted signal is vertically polarized and, in the absence of modulation, has received an electric-field component of the form

$$E_z = E_0 \sum_{n=1}^{N} C_n \cos(\omega_o t + \theta_n) \tag{1}$$

where $E_0 C_n$ is the nth wave amplitude, $\theta_n = 2\pi f_n t + \phi_n$, f_n is the nth wave Doppler frequency, and ϕ_n is a random phase uniform on $[0, 2\pi)$.

The maximum Doppler frequency, denoted f_m, is the vehicle velocity times the carrier frequency f_o divided by the speed of light. The QAM representation of (1) results in baseband signals $I(t)$ and $Q(t)$ modulating the carrier frequency to produce

$$E_z = I(t)\cos(2\pi f_o t) - Q(t)\sin(2\pi f_o t). \tag{2}$$

The in-phase and quadrature signals $I(t)$ and $Q(t)$, respectively, are each the result of low-pass filtering WGN. The filter cutoff frequency is related to the desired f_m being simulated. The envelope $r = \sqrt{I^2 + Q^2}$ has a Raleigh probability distribution.

Assuming the receive antenna is a vertical monopole and the power distribution is uniform over the arrival angle, the power spectrum of I and Q is the well-known bathtub shape [1]

$$S_I(f) = S_Q(f) = \begin{cases} \dfrac{3b}{\pi f_m}\left[1 - \left(\dfrac{f}{f_m}\right)^2\right]^{-1/2}, & |f| < f_m \\ 0, \quad \text{otherwise.} \end{cases} \tag{3}$$

Related to (3) via the inverse Fourier transform is the autocorrelation function of the I and Q components

$$R_I(\tau) = R_Q(\tau) = \frac{3}{2} b J_0(2\pi f_m \tau) \tag{4}$$

where J_0 is the Bessel function of the first-kind of zeroth order. The second-order statistical characterization of the flat fading Rayleigh channel, as embodied in the power spectrum of (3) or autocorrelation function of (4), is the foundation of all baseband channel simulators. When the angular spread is no longer uniform, as is often the case in rural and suburban settings, the autocorrelation function of (4) must be modified. In this paper, however, we use (4) since it is the basis for hardware performance testing in most wireless standards.

Another quantity of interest is the autocorrelation function of the phase $\theta = \tan^{-1}(Q/I)$. The autocorrelation function of the phase can be written as [1]

$$R_\theta(\tau) = \frac{3}{2\pi}\sin^{-1}\left[\rho(\tau)\right] + 6\left\{\frac{1}{2\pi}\sin^{-1}\left[\rho(\tau)\right]\right\}^2 - \frac{3}{4\pi^2}\sum_{n=1}^{\infty}\frac{\rho^{2n}(\tau)}{n^2} \tag{5}$$

where $\rho(\tau) = J_0(2\pi f_m \tau)$ is the unit normalized autocorrelation function of (4).

An envelope characterization of practical interest is the envelope level crossing rate (LCR). The LCR N_R characterizes the average rate at which the envelope r crosses a level R in the positive direction. In [1], it is shown that

$$N_R = \sqrt{\frac{b_2}{\pi b_0}}\, A e^{-A^2} = \sqrt{2\pi}\, f_m A e^{-A^2} \tag{6}$$

where

$$b_n = (2\pi)^n \int_0^\infty S_I(f) f^n \, df \tag{7}$$

is the nth spectral moment of the I or Q Gaussian processes, $S_I(f)$ being the power spectrum of $I(t)$, and $A = R/r_{\text{rms}}$ being

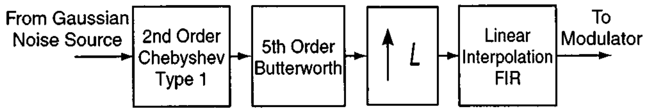

Fig. 3. Shaping-filter realization.

the ratio of the threshold to the rms envelope level. The final result in (6) is obtained by inserting the spectrum of (3) into (7), and evaluating b_2 and b_0. This simplification is significant since it tells us that the LCR is directly related to the maximum Doppler frequency by the ratio of second-to-zeroth spectral moments, i.e.,

$$f_m \equiv \frac{1}{2\pi}\sqrt{\frac{2b_2}{b_0}}. \tag{8}$$

The TIA IS-55A standard channel simulator specification [6] sets tolerances that a hardware simulator must meet or exceed with respect to (3), (5), (6), and (8). The specification is written with hardware simulators in mind, but applies equally well to purely software-based simulators. A key aspect of this specification is that the simulated Doppler frequency must satisfy (8). Additionally, the simulated power spectral density must have at least 6 dB of peaking above $S_I(0)$ and have rolled off at least 30 dB with respect to $S_I(0)$ for $f > 2f_m$. There are many realizable approximations to (3) that meet these conditions. Both of these conditions only partially constrain the I/Q power spectrum. The ratio of spectral-moments condition insures that properly calibrated threshold crossings will occur as well. Additionally, the phase autocorrelation function must match to within a ±0.1 tolerance, the true phase autocorrelation function (5) at $f_m\tau = 0.01$ and 0.15.

B. Implementation Architecture

The QAM method RFA noise shaping filter chosen here is based on [9]. In Fig. 3, we see an IIR filter in cascade with an upsampler and a finite impulse response (FIR) linear interpolation filter. The combination of Chebyshev and Butterworth IIR filters was chosen based on the requirements of the TIA IS-55A specification described earlier. Upsampling and interpolation by L significantly reduces the number of calculations needed in the simulation.

The Chebyshev/Butterworth shaping-filter synthesis is described in detail in [10]. To summarize, the digital IIR filter is derived from an analog prototype using the bilinear transform. For analytical purposes, the effective system function of the noise shaping filter is given by

$$H(z) = H_{\mathrm{IIR}}(z^L)H_{\mathrm{LIN}}(z) \tag{9}$$

where $H_{\mathrm{IIR}}(z)$ is the system function of the Chebyshev/Butterworth filter cascade and $H_{\mathrm{LIN}}(z)$ is the system function of the FIR linear interpolation filter. The impulse response of the linear interpolation filter is the causal triangle sequence

$$h_{\mathrm{LIN}}[n] = \begin{cases} n+1, & 0 \le n \le L-1 \\ 2L-1-n, & L \le 2L-2 \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

The I/Q Doppler power spectrum, in the discrete-time domain, is of the form

$$S_{I,Q}(e^{j2\pi f/f_s}) = \sigma_w^2 = \left|H_{\mathrm{IIR}}\left(e^{j2\pi Lf/f_s}\right)H_{\mathrm{LIN}}\left(e^{j2\pi f/f_s}\right)\right|^2, \quad 0 \le f \le f_s/2 \tag{11}$$

where f_s is the sampling rate in hertz, and σ_w^2 is the variance of the input WGN.

Note that the upsampling operation compresses the frequency response of H_{IIR} by the factor L, thus, the IIR low-pass response is repeated $L/2$ times on the interval [0, 1/2]. The interpolation filter following the upsampler should ideally remove all replications of H_{IIR}, except for the baseband portion, but the linear interpolation filter has only a $|\sin(Lx)/\sin(x)|^2$ magnitude response. Undesired replication terms do pass through H_{LIN}, but as we shall see later in Section III, these terms are typically small.

To calibrate the filter to achieve a desired Doppler frequency f_m, the spectral moments b_o and b_2 are calculated using a discrete-time version of (8)

$$b_n = (2\pi)^n \int_0^{f_s/2} \left|H\left(e^{j2\pi f/f_s}\right)\right|^2 (f/f_s)^n \, df \tag{12}$$

for a particular IIR filter 3-dB cutoff frequency f_c. If f_m from (12) is not within the desired tolerance, then the cutoff frequency of the IIR filter is changed and a second iteration commences. Iteration via a bisection algorithm continues until a filter giving an acceptable f_m results.

The unit normalized autocorrelation function at the noise filter output should ideally be $J_0(2\pi f_m k/f_s)$, where k is the lag value in samples. The noise shaping filter produces the discrete approximation to $\rho(t)$

$$\hat{\rho}[k] = \frac{c[k]}{c[0]} \tag{13}$$

where

$$c[k] = \mathcal{Z}^{-1}\left\{H(z)H(1/z)\right\} \tag{14}$$

and $\mathcal{Z}^{-1}$ is the inverse z-transform operator. The corresponding phase autocorrelation function approximation will be examined in Section III.

C. Prototype Hardware Issues

The target bandwidth of the prototype system was 5 MHz. This number was deemed appropriate for encompassing the development needs of evolving third-generation wireless standards. Building an analog system to accomplish the multitap delay line and fading generators proves to be prohibitively expensive at higher bandwidths. In addition, the accuracy and repeatability increases as more of the design is moved into the digital domain. Thus, an IF sampling architecture was chosen for this design. Implementing a digital system with 5 MHz of bandwidth requires a sampling rate of at least 10 MHz according to the Nyquist sampling theorem. This sampling rate is easily achievable with today's high-speed and low-cost A/D converters and D/A converters. The bottleneck for this system arises from the amount of processing required on the digitized

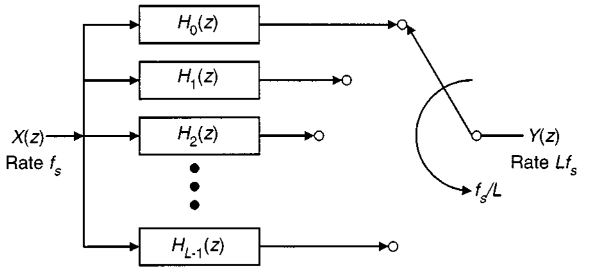

Fig. 4. Polyphase interpolation.

signal. Even with the advances made recently in DSPs, it is still not feasible to implement the entire frequency-selective fading system in a single processor.

A significant portion of the processing power, measured in millions of instructions per second (MIPS), required by the simulator is attributed to the interpolation of the low-bandwidth Doppler spectrum up to the IF of the system. Since the Doppler spectrum must be convolved with the input signal spectrum, or multiplied in the time domain, this requires that the two signals be represented in the digital domain at the same rate. In other words, one sample from the fading generator must be multiplied by a corresponding sample of the input signal. The Doppler spectrum for mobile speeds of interest is typically less than 2-kHz wide [1], while the minimum digital IF for a signal bandwidth of 5 MHz is >2.5 MHz. The resulting interpolation factor is on the order of 1000.

The process of interpolation is described simply as increasing the sample rate of a previously digitized signal by estimating the analog signal's trajectory between the available samples and placing additional samples along this estimated trajectory. This process may be accomplished efficiently and accurately by simply placing zeros between the known samples and filtering the resulting sequence with a properly designed FIR filter, as we see in the last two blocks of Fig. 3. One interesting point to note from this process is that the FIR filter must have a memory, or length, long enough to span the distance between the known samples. In the case of the channel simulator, this would require an FIR filter of minimum length 1000. Using a polyphase filter representation, as described in [11], reduces the complexity of this filter considerably. The polyphase filter model is shown in Fig. 4, where the coefficients of the polyphase filters $h_p[n]$ are related to the coefficients of the original interpolation filter $h[n]$ by

$$h_p[n] = h\big[nL + p\big]. \tag{15}$$

The advantage of the polyphase structure is that it performs the filtering at the lower sampling rate.

Even with the efficiency improvement of the polyphase filter, implementing an interpolation on the order of $L = 1000$ in real time is still impractical. A more efficient approach is to perform the interpolation in multiple steps [12]. This reduces the complexity to a more reasonable level for realization in a DSP chip. Thus, designing the interpolation becomes an exercise in optimizing the number and lengths of the interpolation stages.

Specifically, for the channel simulator, the number of processor cycles required to perform the interpolation is dictated by the inner loop of the polyphase FIR interpolation filter. The execution of this loop in a DSP requires four loads, two additions, two multiplies, and one store; for a total of nine instructions. Assuming that today's very long instruction word (VLIW) DSP architectures can execute some of the instructions in parallel (a complicated issue determined by factors such as pipeline delays and allocation of instructions to separate arithmetic logic units), this operation could be executed in as few as two cycles. The absolute best case is for the fastest available processor, i.e., the Texas Instruments Incorporated TMS320C6701, which has eight parallel execution paths, two of which may perform multiplies [13]. This assumes that the pipeline is kept full 100% of the time. A more realistic, yet still somewhat optimistic, estimate would place the number of cycles at four. Given that each tap delay has both I and Q components, the estimated number of cycles per tap for the inner loop of the FIR filter is eight. Since the system clock is 160 MHz and the minimum sampling rate for the desired bandwidth is 10 MHz, this means that each fading generator must output a value every 16 clock cycles. This analysis constrains the number of taps in the multipath model to two in the very best case if the DSP must do all of the processing. This result is what led to the inclusion of an FPGA in the system to increase the number of available MIPS. The FPGA accelerates the interpolation and multiplication processes by implementing them in parallel.

The final system architecture, as shown in Fig. 5, consists of three hardware components. A standard PC provides the Doppler shaping-filter synthesis, amplitude scaling, and the graphical user interface (GUI). A Texas Instruments Incorporated TMS320C6701 EVM DSP board, which connects to the PC's peripheral component interconnect (PCI) bus is responsible for the random number generator, Doppler spectrum shaping filter, and the first interpolation stage. An AED100 daughterboard from Signalware contains the A/D and D/A components, as well as the FPGA for the tapped delay line, multipliers, and final interpolation stage of each fading generator.

The system employs bandpass sampling of a 70-MHz intermediate frequency signal with up to 5-MHz bandwidth. The bandpass filter is a surface acoustic wave (SAW) device. Bandpass sampling theory states that a signal may be sampled with a rate that is much less than the highest frequency component of the input signal without aliasing, provided that [14]

$$nf_s + B \le 2f_o \le (n+1)f_s - B \tag{16}$$

where n is a positive integer and B is the bandwidth of the signal. This under sampling process effectively translates the input signal to a digital IF, which is an alias of the input signal, as shown in Fig. 6.

For this design, the sample clock is 11.1 MHz, thereby setting the input frequency range from 66.6 to 71.6 MHz, with a digital IF of 2.5 MHz. An analytic signal is derived by splitting the signal into two paths and Hilbert transforming one path with an asymmetric FIR filter, while introducing the equivalent group delay in the second path via an FIR all-pass filter. The resulting analytic signal enters the tapped delay line where user-selected delays are introduced. The delayed signals are each multiplied

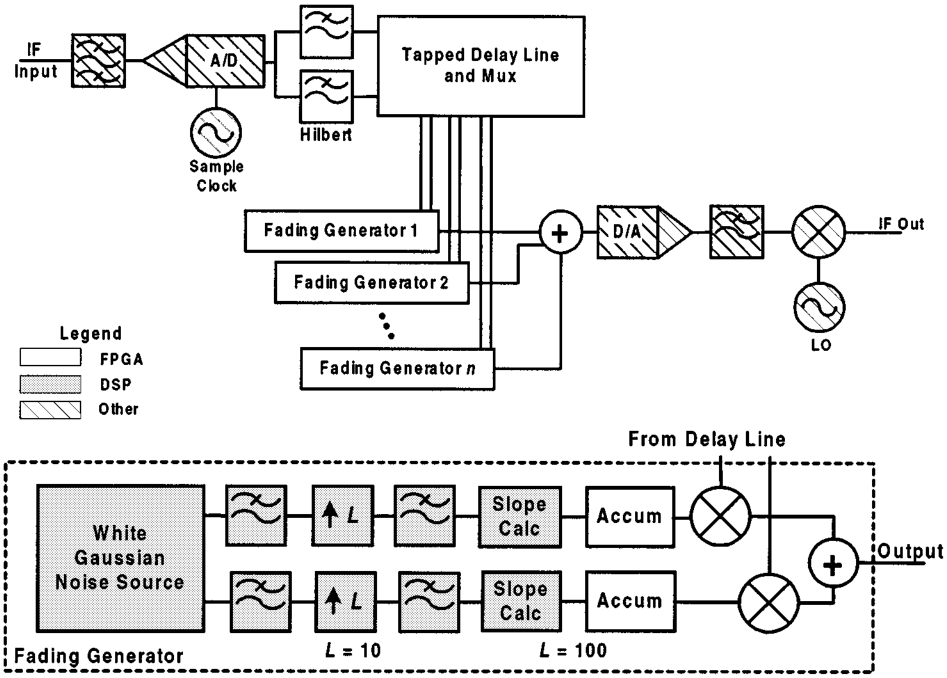

Fig. 5. System architecture.

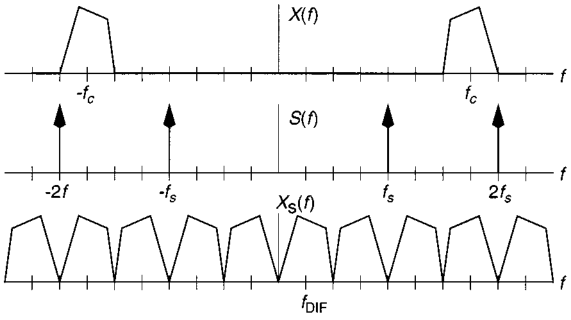

Fig. 6. Using bandpass sampling to digitize and frequency translate a signal about f_c down to $f_{\rm DIF}$.

in the time domain by the independent fading generator signals, as shown in Fig. 5.

The fading generator consists of the random number generator, two interpolation stages, and modulator. The random number generator creates two WGN random number sequences. This method is based on the C language compiler's built-in rand() function. The rand() function generates uniformly distributed random integers, which are converted into a Rayleigh distributed random variable by inverting the Rayleigh cumulative distribution function to obtain

$$R = \sqrt{-2\sigma^2 \log_e(1 - \texttt{rand}()/\texttt{RMAX})} \qquad (17)$$

where σ^2 sets the variance of the Rayleigh variates and RMAX is the maximum random number generated by the rand() function. The resulting uncorrelated outputs are generated using

$$X = R\cos\theta \quad \text{and} \quad Y = R\sin\theta \qquad (18)$$

where θ, a uniform on the $[0, 2\pi]$ random variate, is also generated using the rand() function. To speed up the implementation, a table of R values is stored in the EVM memory along with cosine and sine tables.

Each output from the random number generator has its spectrum shaped by the IIR filter. The mobile speed and carrier frequency parameters are input by the user through a host PC GUI application. The filter coefficients for the Doppler shaping IIR filter is calculated on the host and downloaded over the PCI bus to the DSP card. The first interpolation stage is accomplished in the DSP by using the FIR polyphase representation of a linear interpolation filter. The second interpolation stage is also a linear interpolator. Due to the high sampling rate, this stage is implemented partially in the FPGA. The impulse response for the second interpolation stage is of the same form as that in the first stage, but is realized differently. The DSP calculates a slope value between the previous and current samples output from the first interpolation stage and writes this slope value to the FPGA. The FPGA increments an accumulator by the slope value on each sample clock cycle to obtain the output signal.

Once the interpolation of the Doppler spectrum is complete, the resulting I and Q components are multiplied by the corresponding components of the sampled and delayed input signal. The I and Q product's sum to complete one tap of the multipath simulator. Each additional tap repeats this process with the outputs being summed together immediately before the D/A converter. An analog third-order Bessel low-pass filter is used for baseband reconstruction.

It is worth noting that the flexibility of the architecture allows reconfiguring the system for use with Doppler spectra other than (3) simply by reprogramming the filter coefficients. Furthermore, the system hardware is not limited to using the QAM method. It is feasible that the popular Jakes method could be used as well.

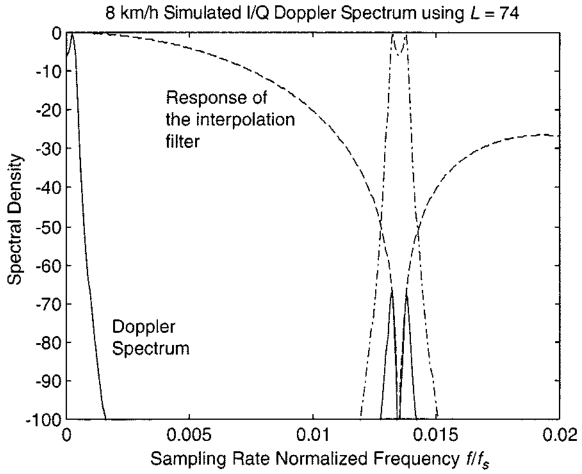

Fig. 7. Simulated Doppler spectrum (magnitude response squared of noise filter) in decibels for $v = 8$ km/h. The unfiltered upsampler response and interpolation filter response are also shown.

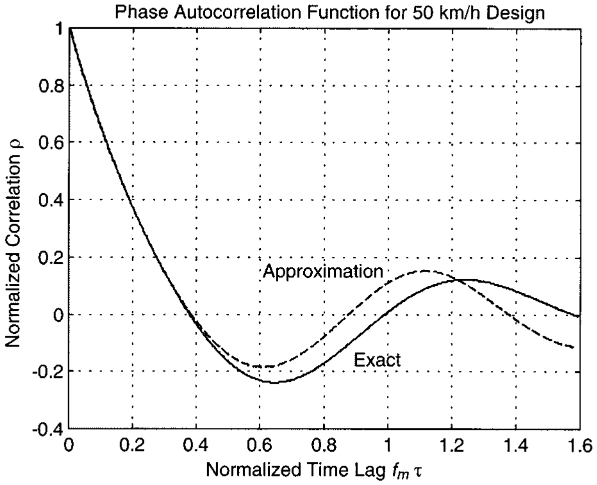

Fig. 8. Normalized phase autocorrelation function showing both ideal theory and the noise filter approximation with the 50-km/h design.

III. System Performance

First, we consider the theoretical performance of the simulator for conditions called out in [6]. Next, actual performance results for the hardware prototype are presented.

A. Theoretical Results

Consider a carrier frequency fixed at 881 MHz and a sampling frequency of 19.2 kHz (or in a data-link simulation 19.2 kb/s at one sample per bit), which is appropriate for a pure baseband channel simulation. The minimum acceptable filter cutoff frequency is constrained to be no smaller than 0.025, thus forcing the upsampling factor L to adjust accordingly. To obtain a simulated Doppler frequency of 6.51 Hz, which corresponds to a velocity of 8 km/h at 881 MHz, requires that $L = 74$. The effective Doppler spectrum that the 8-km/h filter noise filter design produces is shown in Fig. 7. The raw upsampled IIR response is shown, as well as the frequency response of the $L = 74$ linear interpolation filter.

In Fig. 7, we see that the first replication of the IIR response is passed by the linear interpolation filter, but is greater than 60 dB below the dc spectral level. We also see that the desired IS-55A peaking of 6 dB is present and the 30-dB rolloff condition is also satisfied. Rejection of spectral images as a result of the interpolation filter holds as L increases to very large values, as required in the hardware prototype.

In Fig. 8, the normalized phase autocorrelation function is shown for a vehicle velocity of 50 k/h. At $f_m\tau$ values of 0.05 and 0.15, we see that the expected performance is easily within the IS-55A tolerance of ±0.1.

B. Experimental Results

Recall that the hardware prototype system has a bandwidth of approximately 5 MHz with a delay tap spacing of less than 100 nanoseconds. The fading statistics of each tap compare favorably with test specifications in the TIA IS-55A performance standard for mobile stations [6]. The measured LCR is compared to the theoretical in Fig. 9. The dynamic range of the prototype is limited by the 8-b A/D and D/A converters. More dynamic range is desirable because the multipath environment typically causes fades greater than 40 dB. This could easily be implemented by integrating 10- or 12-b A/D and D/A converters. The prototype is limited to three delay taps due to the amount of logic in the FPGA. This prototype uses a Xilinx XC4044XL part with a maximum of 80 000 logic gates. This component is at 97% capacity with three taps and a maximum delay spread of about 20 μs. Given this limitation on FPGA resources, the delay spacing is not dynamically reconfigurable in the prototype. A new FPGA file must be loaded into the FPGA for each desired delay configuration. These limitations could be easily overcome by upgrading the FPGA on the daughter board. The state-of-the-art Xilinx Virtex FPGA has a capacity greater than 4 million gates. With this factor of 50 increase in capacity, it is safe to assume that 12 configurable tap delays could be implemented in the FPGA. The DSP has enough available bandwidth to easily accommodate 12 taps.

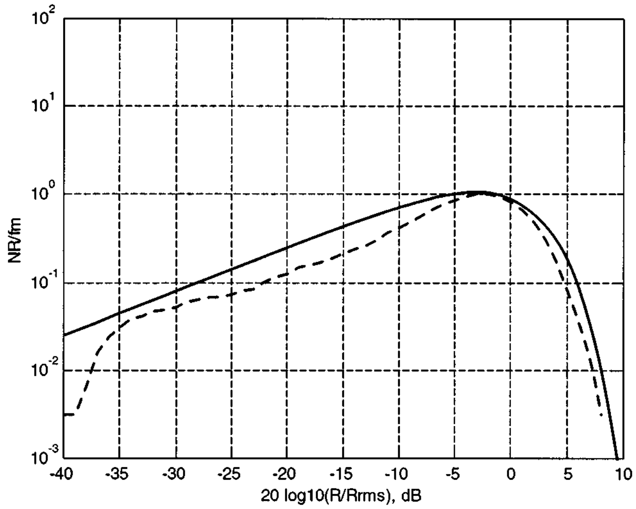

Fig. 9. Measured (dashed) versus theoretical (solid) LCRs.

Fig. 10 shows the measured Doppler spectrum for a simulated signal with mobile velocity of 100 k/h and carrier frequency of 850 MHz. The 3-dB bandwidth is approximately 150 Hz and

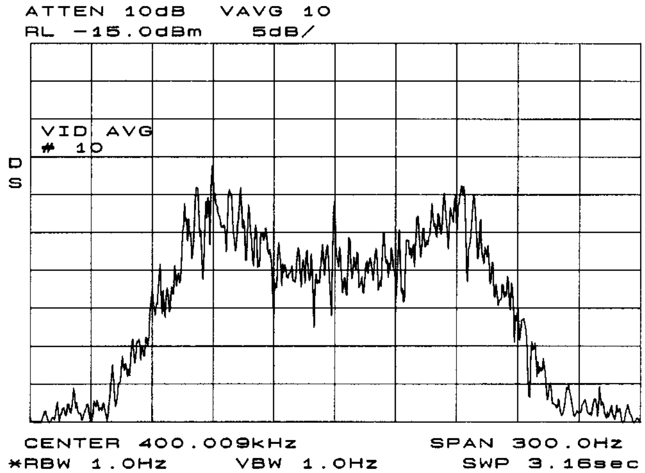

Fig. 10. Measured Doppler spectrum.

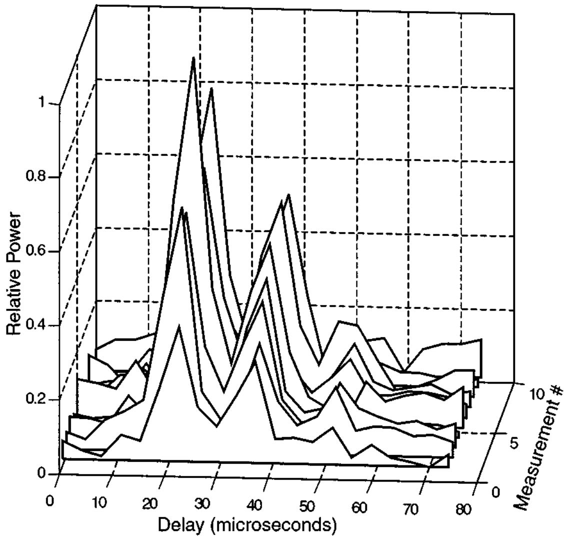

Fig. 11. Delay spread characterization.

the peak is approximately 60 Hz from the carrier f_o, which is roughly 75% of f_m. These values are consistent with the desired results.

The characterization of the delay properties of the channel simulator is based on the sliding correlator measurement system [2]. Since a spread-spectrum sliding correlator measurement system with the desired frequency range was not available for use in this testing, a system using a global system for mobile communications (GSM) transmitter and receiver was implemented. The transmitter outputs GSM access bursts during one time slot of each frame. Each access burst contains a training sequence with good correlation properties, which is used for timing advance measurements in the GSM reverse link [15]. A hardwired signal from the transmitter to the receiver provides timing synchronization of the bursts. The receiver samples the GSM signal at the baud rate and correlates the received samples with the known training sequence. As in the spread-spectrum sliding correlator, the resulting correlation shows the delay spread profile of the channel. Fig. 11 shows a waterfall plot of ten consecutive bursts, with the channel simulator configured for three taps with power levels of 0, -6, and -12 dB. The second tap is configured for an excess delay approximately 14.5 μs. The third tap is configured for an excess delay of approximately 29 μs.

IV. Conclusion

In this paper, we have presented a real-time DSP approach for a frequency-selective mobile radio channel simulator. An RFA to the true Doppler spectrum is implemented using QAM with IF sampling. In particular, the noise shaping filter is implemented as a cascade of an IIR filter, a rate L upsampler, and simple linear interpolation filter. A cascade of polyphase FIR filters efficiently implements the linear interpolation filter for large L. The performance of this approach was found to be favorable with the IS-55A specifications, and although not as theoretically exact as the commonly used Jakes method, it is computationally very efficient.

The equivalent cost of the hardware prototype is much less than current commercial simulators, in spite of the fact that this implementation has some inefficiency due to the number of unused parts included in both the DSP board and the daughterboard. Integrating the system onto one custom printed circuit board and eliminating unused parts could reduce the cost considerably. It is believed that this system could be built in small quantities for as little as $2000 per unit. This does not include the cost of the PC or the signal generator used for the local oscillator, which are common items in most engineering laboratories. When compared to the cost of the commercially available channel simulators, this is a reduction of at least one order of magnitude.

Improved performance from the prototype will require increased precision A/D and D/A converters. It is proposed that a more optimal IIR filter be considered. One approach is to consider the filter design as a constrained optimization problem. Subject to performance specifications/constraints, the error between the true autocorrelation function and the filter generated value can then be minimized. Future testing requirements may also dictate that angular spreads more representative of rural and suburban environments be available in channel simulators. Work on developing appropriate RFA models for these cases is also of interest. In any case, it will be desirable for the overall IIR filter order to be kept low so as to insure low computation counts, or equivalently provide many simulated RF paths per DSP processor.

References

[1] W. C. Jakes, Ed., *Microwave Mobile Communications*. New York: IEEE Press, 1974.

[2] T. S. Rappaport, *Wireless Communications Principles and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[3] G. Arrendondo, W. Chriss, and E. Walker, "A multipath fading simulator for mobile radio," *IEEE Trans. Veh. Technol.*, vol. VT-22, pp. 241–244, May 1973.

[4] E. Caples, K. Massad, and T. Minor, "A UHF channel simulator for digital mobile radio," *IEEE Trans. Veh. Technol.*, vol. VT-29, pp. 281–289, May 1980.

[5] E. Casas and C. Leung, "A simple digital fading simulator for mobile radio," *IEEE Trans. Veh. Technol.*, vol. 39, pp. 205–212, Aug. 1990.

[6] *Recommended Minimum Performance Standard of 800 MHz Dual-Mode Mobile Stations*, TIA IS-55A, 1993.

[7] B. Schweber, "RF-channel simulators: Bring reality's challenges to your prototype," *Electron. Design News*, pp. 52–63, Sept. 1998.

[8] *Model MP-2500 Multipath Fading Emulator Technical Manual*, NoiseCom, Paramus, NJ, Jan. 1996.

[9] M. A. Wickert and J. M. Jaycobsmeyer, "Efficient Rayleigh mobile channel simulation Using IIR digital filters with upsampling," in *Proc. ICSPAT*, Oct. 1995, pp. 391–396.

[10] C. M. Keen, "C language program simulator for a Rayleigh fading channel," Master's thesis, Dept. Elect. Comput. Eng., Univ. Colorado at Colorado Springs, Colorado Springs, CO, 1994, submitted for publication.

[11] F. Harris, "Multirate digital signal processing in digital receivers," presented at the 2nd Int. DSP Commun. Syst. Symp., Adelaide, Australia, Apr. 1994.

[12] R. E. Crochiere and L. R. Rabiner, "Interpolation and Decimation of digital signals—A tutorial review," *Proc. IEEE*, vol. 69, pp. 300–331, Mar. 1981.

[13] "TMS320C62x/C67x CPU and Instruction Set Reference Guide," Texas Instruments Incorporated, Dallas, TX, Texas Instruments SPRU189C, Mar. 1998.

[14] B. L. Fox, "Analysis and dynamic range enhancement of the analog-to-digital interface in multimode radio receivers," Master's Thesis, Dept. Elect. Comput. Eng., Virginia Polytech. Inst. and State Univ., Blacksburg, VA, 1997, submitted for publication.

[15] S. M. Redl, M. K. Weber, and M. W. Oliphant, *An Introduction to GSM*. Norwood, MA: Artech House, 1995.

Mark A. Wickert (S'76–M'78) received the B.S. and M.S. degrees in electrical engineering from the Michigan Technological University, Houghton, in 1977 and 1978, respectively, and the Ph.D. degree in electrical engineering from the University of Missouri at Rolla, in 1983.

From 1978 to 1981, he was a Design Engineer with the Motorola Government Electronics Group, Scottsdale, AZ, where he was involved with very high-speed digital communication systems. In June 1984, he joined the faculty of the University of Colorado, Colorado Springs, where he is currently a Professor of electrical engineering. For the 1998–1999 academic year, he was on sabbatical leave at Pericle Communications Inc., Colorado Springs, CO. His current research interests include wireless communication systems, both time-division multiple access (TDMA) and code-division multiple access (CDMA), wireless local area networks, statistical signal processing, and real-time digital signal processing.

Dr. Wickert is a member of Tau Beta Pi, Eta Kappa Nu, Phi Kappa Phi, and Sigma Xi.

Jeff Papenfuss (M'00) received the B.S. degree in electrical engineering from the University of New Mexico, Albuquerque, in 1992, and the M.S. degree in electrical engineering from the University of Colorado, Colorado Springs, in 2000.

Since 1996, he has been involved with wireless data infrastructure and terminals as a Test Engineer, DSP Engineer, and Software Engineer. He has been with Omnipoint Technologies and Voicestream Wireless, and is currently with Xircom (an Intel Company), Colorado Springs, CO. His current interests are in software and hardware architectures for wireless terminals.