Design and Implementation of a Wide-Band Real-Time Mobile Channel Emulator

Juan J. Olmos, *Member, IEEE,* Antoni Gelonch, *Member, IEEE,*
Fernando J. Casadevall, *Member, IEEE,* and Guillem Femenias, *Member, IEEE*

***Abstract*—In this paper, a new wide-band mobile channel emulator for the CODIT project is designed and implemented. The UMTS code-division testbed (CODIT R2020) is a research project within the European RACE-II program set up by the Commission of the European Community. Our goal is to be able to simulate in the laboratory, in real time, the multipath propagation found in the mobile radio channel. As code-division multiple access (CDMA) is the access technique within the CODIT project, it was realized that the channel emulator must have simultaneously good delay resolution between propagation paths and long duration of the impulse response. These considerations led to a very flexible channel emulator specifically designed to host the new wide-band channel models developed within the CODIT project. Our emulator features three independent inputs and two outputs, up to 20 complex propagation paths, 10-MHz radio frequency (RF) bandwidth, a delay resolution of 50 ns, and a maximum duration of the channel impulse response of 80 μs. Starting with an explanation of the global structure of the new channel emulator, we derive the optimum design of the interpolation procedures and present the main implementation issues arising from our initial architecture. Finally, we report the results of the laboratory tests of the first prototype of the channel emulator.**

***Index Terms*—Channel emulator, digital signal processing, fading channel, mobile communications.**

I. Introduction

THIS PAPER describes the design and implementation of the wide-band real-time mobile channel emulator which has been built to be used in the laboratory tests of the CODIT project. The UMTS code-division testbed (CODIT R2020) is a research project within the European RACE-II program set up by the Commission of the European Community. Its objective is to explore the potential of code-division multiple access (CDMA) as the access method for a third-generation mobile communications system having an open and flexible multirate radio interface [1]. The complete system demonstrator (testbed) of the CODIT project comprises test mobile stations, radio base stations, a radio network controller, and a channel emulator. Although this paper deals only with the channel emulator issues, information regarding the CODIT system concept can be found in [2] and [3]. The CODIT project was finished in 1995. It is worth noting that recently the three major standards bodies, ETSI, TIA, and NTT, have selected wide-band CDMA as the access method for third-generation cellular systems and that the features of the channel emulator presented in this paper are well aligned with the air interface of these systems.

The aim of the channel emulator is being able to emulate in the laboratory and in real time the highly dispersive propagation conditions of the mobile radio channel due to multipath caused by reflection and scattering. During the initial phase of the project it was specified that the emulator had to implement the new wide-band channel models to be developed within CODIT for both outdoor and indoor environments and take into account the Doppler effect due to the movement of the vehicle at different speeds. Moreover, as soft handover had to be demonstrated in the testbed, the emulator should also be able to handle more than one mobile channel at the same time.

At the beginning of the CODIT project, in 1992, there had been proposed several channel emulators for mobile radio environments ([4], for example), some of which were developed in relationship with the global system for mobile communication (GSM) systems, for the purpose of equipment and system testing by emulating the channels corresponding to the standard delay profiles accepted by COST207 [5]. However, the channel emulators used for these channel models could not be used in the CODIT project. The reason is that in a CDMA system it is possible to have both long impulse response of the channel and high bit rate in the radio interface since the RAKE receiver [6] takes advantage of the statistical independence of the different rays that characterize the multipath propagation. This means that the channel emulator must had simultaneously good delay resolution between propagation paths and long duration of the impulse response. As these features were not present in the commercially available channel emulators at that time, a new channel emulator to implement the CODIT propagation models was required. This paper presents the design and implementation of such a wide-band channel emulator. The organization of the paper is as follows: in Section II an overview of the CODIT approach for wide-band channel modeling is presented and then the architecture and main specifications for the channel emulator are derived. Section III deals with the implementation aspects, and it is divided into sections, each one giving a functional description of the printed

Manuscript received May 2, 1996; revised July 27, 1998. This work was supported in part by Telefónica I+D, Spain, CICYT (Spanish Education Ministry) under Contract TIC940870C0201 and the European Community in the RACE program (Research and development into Advanced Communications technologies for Europe) under the RACE 2020 CODIT project.

J. J. Olmos, A. Gelonch, and F. J. Casadevall are with the Department of Signal Theory and Communications, Universitat Politècnica de Catalunya (UPC), 08034 Barcelona, Spain.

G. Femenias is with the Departament de Ciències Matemàtiques i Informàtica, Universitat de les Illes Balears (UIB), E07071 Palma de Mallorca, Spain.

Publisher Item Identifier S 0018-9545(99)01046-4.

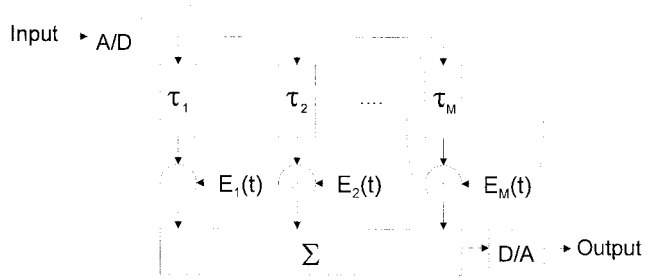

Fig. 1. Complex FIR filter having the desired channel impulse response.

circuit boards present in the emulator, i.e., radio frequency (RF) interface cards, tap circuit cards, digital signal processing (DSP) card, and control card. Section IV presents the results obtained during the laboratory tests of the emulator, and finally, Section V shows the conclusions. For a more clear exposition, two Appendixes have been also included.

II. Global Structure of the Channel Emulator

Within the propagation studies work package of the CODIT project a new approach for wide-band channel modeling has been considered [7]. The model classifies the encountered situations into several categories each one having different wide-band characteristics, i.e., urban, suburban, rural, suburban hilly, rural hilly, microcell LOS and NLOS, and indoors. Appropriate software simulations tools have been developed for each one of these environments in order to properly characterize the channel by means of its time-variant complex impulse response $h(t,\tau)$, that is, the impulse response at time $t+\tau$ to an impulse applied at time t. Assuming a wide-sense stationary uncorrelated scattering (WSSUS) environment, it can be seen that the scattering function $S(\tau,f)$, defined as the Fourier transform of the temporal autocorrelation of the complex impulse response, completely describes the statistical channel behavior of the channel and is closely related to the physical environment characteristics. This idea has been used to validate the simulation results against field measurements undertaken also by the CODIT project partners. A good agreement has been found between the simulated and measured scattering functions in the different environments. From a practical point of view, the emulated channel impulse response will be

$$h(t,\tau)=\sum_{i=1}^{M} E_i(t)\cdot\delta(\tau-\tau_i) \tag{1}$$

where $E_i(t)$ is the field contribution due to the ith scatterer, τ_i is the associated delay, and M is the number of resolvable paths. More details on the channel model and on how the $E_i(t)$ coefficients are obtained can be found in Appendix A.

According to (1), we see that, assuming an equivalent low-pass model, the channel emulator could be implemented by means of a complex digital FIR filter with slowly time-variant coefficients as shown in Fig. 1. The base-band digital implementation is also convenient if we wish to achieve the maximum degree of flexibility. In order to guarantee the maximum precision in the use of the CODIT channel models, the samples of the $E_i(t)$ processes will be obtained by off-line software simulations and stored in a memory inside the channel emulator. A different simulation will be carried out for each environment (urban, suburban, rural, etc.), and also the delays τ_i and the number of resolvable paths M must be fully programmable in accordance with the environment type. The emulator will then "play back" the stored time-variant impulse response in real time. The part of the block diagram enclosed in a dotted ellipse in Fig. 1 (a tap circuit) has been considered as the building unit of the emulator. This means that the core of the emulator consists of a set of identical tap circuits whose inputs and outputs are properly multiplexed.

In the CODIT project, an asynchronous direct-sequence (DS)-CDMA technique with three different chip rates is investigated, i.e., 1.023, 5.115, and 20.46 Mchip/s, but only the two lower chip rates are implemented in the testbed. In order to assess the impact of the adjacent channel interference, it was considered that the emulator bandwidth should span at least two adjacent CDMA channels, so the final bandwidth specification for the channel emulator was 5 MHz in baseband or 10 MHz in RF.

Taking into account that in the laboratories it is easy to build up and down converters (using passive mixers and synthesized signal generators), it was decided that the emulator would operate at a fixed intermediate frequency of 70 MHz. This assumption not only simplifies the design of the RF interfaces, but also gives more flexibility to the emulator since it would be able to work in different bands with different channel spacing. So, the 70-MHz input signal must be downconverted and the samples of the complex envelope [in-phase and quadrature (I/Q) components] must be obtained. Then, this complex samples are digitally processed to emulate the channel behavior in real time and upconverted again to the intermediate frequency. In order to achieve a large dc and image rejection, the I/Q components are extracted by means of a double Nyquist digital product detector [8]. This means that the digital sampling rate is four times the bandwidth of the baseband components, that is, 20 Msamples/s, and so the time resolution (minimum delay between the FIR taps) is 50 ns.

In accordance with the propagation models for outdoor environments, the maximum impulse response duration has been taken equal to 80 μs. Obviously, if we want to emulate such a long impulse response, the FIR filter taps cannot be equally spaced, since it would require a not feasible number of taps (for example, 1600 taps are required for a duration as large as 80 μs). So, the delay of each particular tap is independently achieved by first-in–first-out (FIFO) memories. In our practical implementation the maximum number of FIR taps, that is, the value of M in (1) is always less or equal than 20. This value has proven to be sufficient for most of the emulated situations, since in fact this is equivalent to consider as relevant only a few number of scatterers placed near the base and mobile station.

In order to demonstrate soft handover in the testbed, it was envisaged a laboratory test configuration as shown in Fig. 2. In Fig. 2, we see that the mobile station (MS) is linked

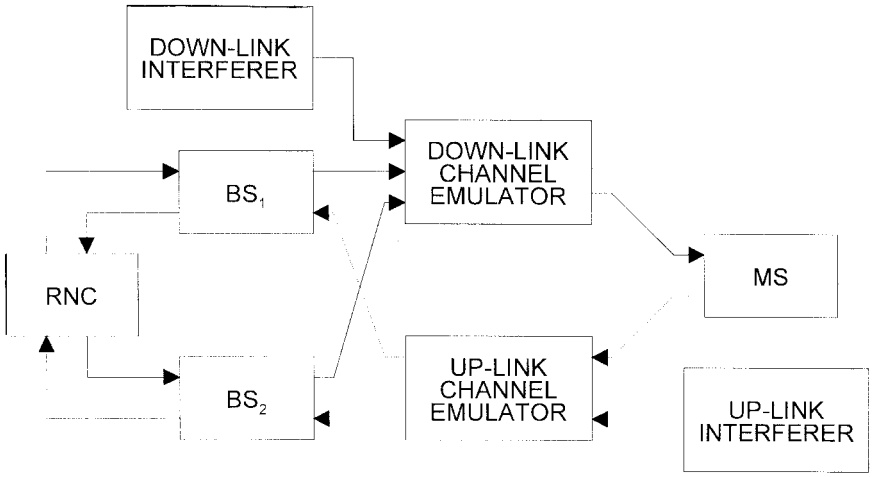

Fig. 2. Laboratory configuration for soft handover testing.

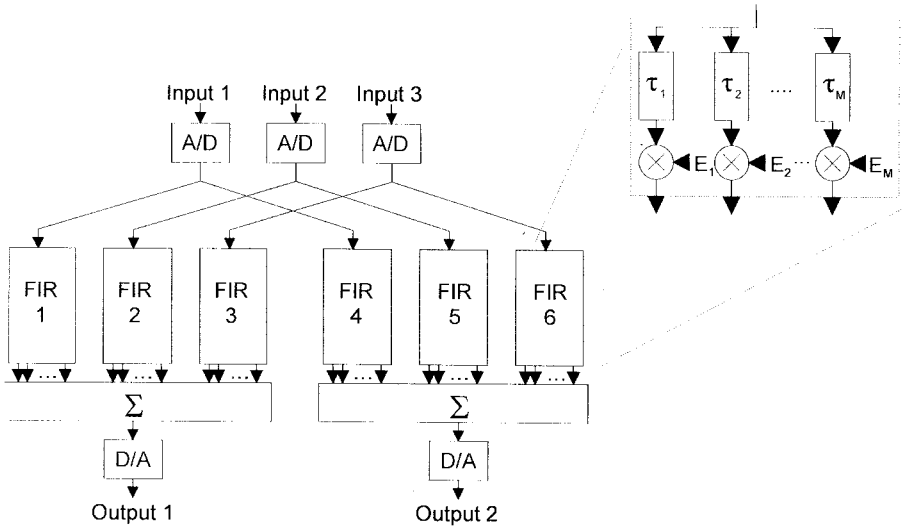

Fig. 3. Conceptual block diagram of the channel emulator.

simultaneously to two base stations (BS_1 and BS_2) which exchange information through the radio network controller (RNC). Uplink and downlink interferers are also included. To build such a configuration, two channel emulators are needed, and each one of them should be able to emulate several independent radio channels and have at least three independent inputs and two independent outputs. To achieve these features the conceptual block diagram of the emulator is as shown in Fig. 3. Notice that up to six different FIR filters, which can be individually programmed, may be simultaneously present during real-time emulation. If one of these filters is not needed, the free taps can be used for another filter or its coefficients set to zero. The number of taps in each FIR (value of M) can be freely selected as long as the sum of the number of taps of all the filters is less or equal than 20. Input and output multiplexers are provided in each tap circuit to set the desired connections.

Given that the signal sampling rate is 20 Msamples/s, the coefficients of the FIR taps, that is, the E_i processes in Figs. 1 and 3, must be updated every 50 ns. As this is a "stored" emulator, a RAM of very large size would be required in order to have a reasonable emulation time interval. To overcome this limitation, extensive interpolation of the recorded samples must be done. Even at high vehicle speed in an outdoor environment, the rate of change of the radio channel is much smaller than the signal bandwidth, so the interpolation ratio of a single interpolation process would be too high to be realistic. Our approach is based on two cascaded interpolation stages where the first one is software based and is performed off line, and the second one is hardware based and performed in real time. The emulator includes a DSP card with a single TMS320C30 DSP processor and 1 Mb of RAM whose purpose is to link (using the RS232 interface) the emulator to an external personal computer (PC) that runs the user interface program. The hard disk of the PC stores the slightly oversampled samples of previously measured channel impulse responses, so the sample sets for a given emulation experiment are not too big and can be readily transferred to the DSP RAM inside the emulator. As will be explained in Section III, each individual tap circuit card has a RAM memory to store

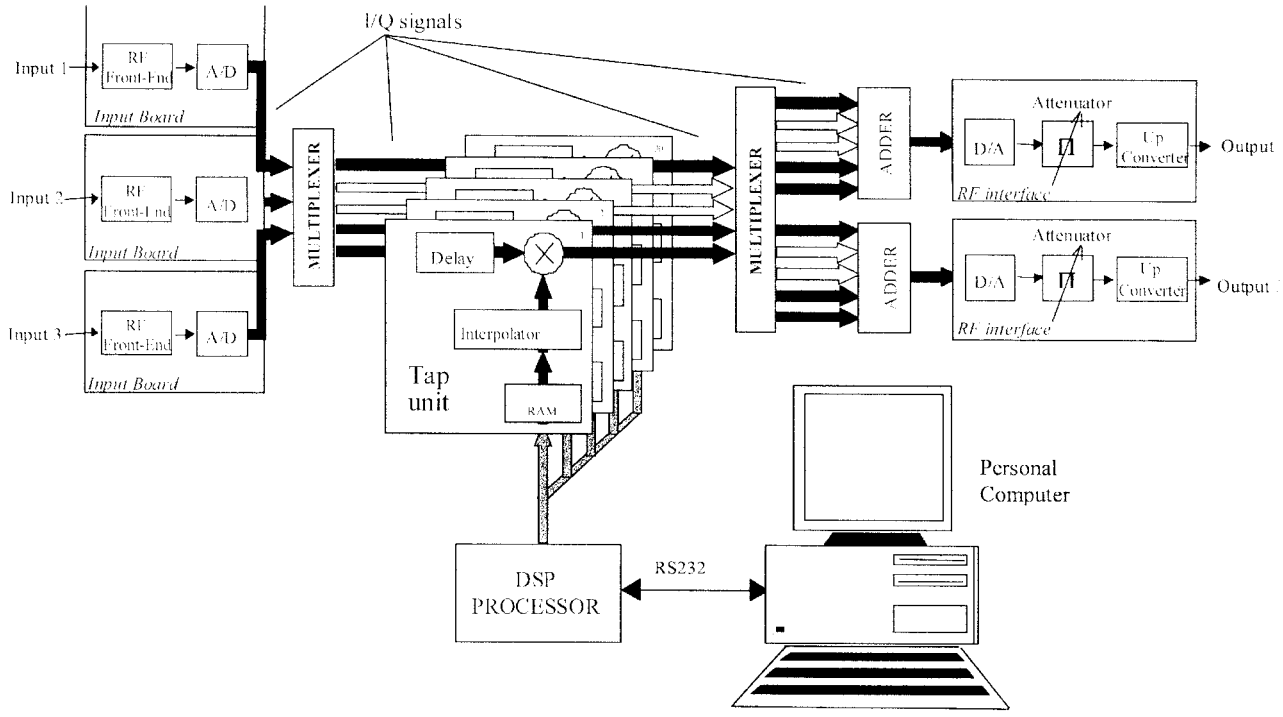

Fig. 4. General structure of the channel emulator.

its impulse response samples and a hardware interpolator. So, once the DSP has read the impulse response samples from the PC and before starting the emulation, it interpolates the original samples and fills the tap circuit RAM's. Then the emulation starts. The RAM banks at each tap are read and interpolated in real time to achieve an updating rate of the channel impulse response of 50 ns. More details on the interpolation design can be found in Appendix B.

Fig. 4 shows the general structure of the channel emulator. In practice, the multiplexers and the adders are distributed in the 20-tap circuits, but from the point of view of signal flow things go as depicted in Fig. 4, that is, for every tap circuit the input multiplexer selects one of the three inputs and the output multiplexer selects one of the two adders. A control card is included to program the multiplexers and the delays and to decode addresses. All the user programmable features are centralized in a user friendly software interface that runs in the attached PC. Digitally controllable RF attenuators are placed at each output since the DSP runs a program, in real time, that controls them to emulate shadowing fading (if desired) and to produce test attenuation patterns to check the power control loops behavior.

To summarize, the main characteristics of the channel emulator are as follows.

Analog part	
Bandwidth	± 5 Mhz
Input signal level	-20 dBm, ± 5 dB
Input frequency	70 MHz
Spurious signals	below 40 dBc
Number of independent inputs	3
Number of independent outputs	2
Input and output impedances	50 Ω.

Digital part	
Sampling frequency at the A/D converters	40 MHz
Sampling rate (I/Q components)	20 MHz
A/D converter bit resolution	10 bits
D/A converter bit resolution	12 bits
Maximum number of taps	20
Maximum impulse response duration	80 μs
Delay resolution	50 ns
Attenuation/tap amplitude (1-dB step)	40 dB
Doppler frequency (max. 1000 Hz)	variable
Ability to emulate shadowing effects (fading rate $\leq$ 14 Hz)	yes.

III. Implementation Aspects

In this section, the functional description of the different cards that form the emulator is given, as well as the more relevant implementation details of each one. The emulator is implemented on nine printed circuit boards each one with a size of 339 $\times$ 366 mm. Five of these cards are identical, and each one holds four fully programmable tap circuits. The four remaining cards are, respectively, devoted to input RF interfaces, output RF interfaces, control card, and DSP card. The nine printed circuit boards are linked together by a backplane based on a VME bus architecture. The rack is fitted with several power supply units giving all the required voltages, i.e., ± 5 V and ± 12 V. In order to avoid digital noise problems, separate power supplies for the analog and digital parts of the emulator as well as isolated ground planes have been designed. To help integrated circuits (IC's) heat

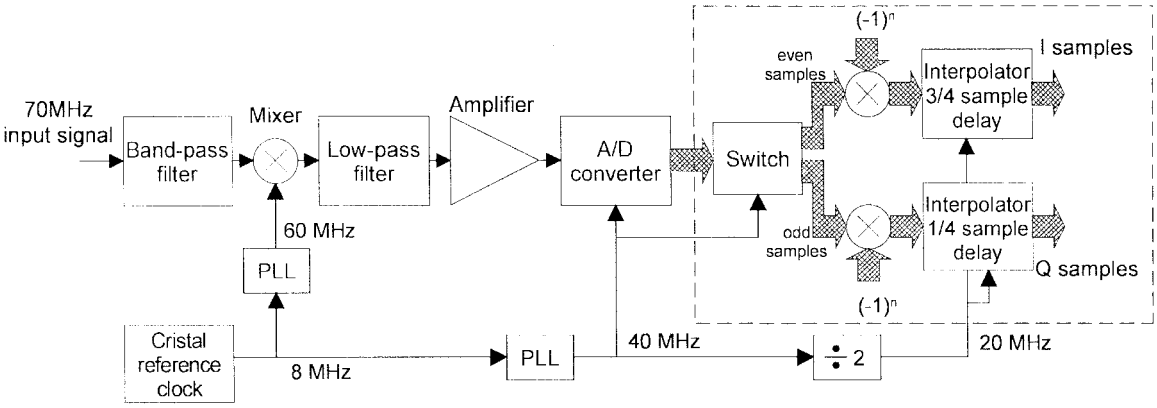

Fig. 5. Block diagram of the input RF interface.

dissipation a set of cooling fans is placed at the bottom of the overall assembly.

A. Input RF Interface

As stated in Section II, three input RF interfaces must be included in the emulator. Fig. 5 shows the block diagram of one input RF interface. All three input interfaces are identical and are placed in the same printed circuit board. First, the input signal at 70 MHz is band-pass filtered (to limit the noise) and downconverted to a band-pass signal centered at 10 MHz. This is done because the adopted I/Q extraction method requires sampling at four times the carrier frequency, so this frequency must be kept as low as possible. Then the 10-MHz signal is amplified to properly drive the analog–digital (A/D) converters and sampled at 40 Msamples/s. The double Nyquist digital product detector (within dotted lines in Fig. 5) finally extracts the I/Q samples of the base-band equivalent input signal.

If we call $r(t)$ to the band-pass signal at the input of the A/D converter, it can be written as

$$r(t) = \Re e(\{r_I(t) + jr_Q(t)\}e^{j2\pi f_0 t}) \tag{2}$$

where $\Re e$ means real part, $r_I(t)$ and $r_Q(t)$ are, respectively, the in-phase and quadrature components of $r(t)$, and f_0 is the carrier frequency. If the sampling rate at the A/D converter is $f_m = T_m^{-1} = 4f_0$, the samples of $r(t)$ can be written as

$$\begin{aligned} r(nT_m) = &r_I(nT_m)\cos\left(\frac{\pi}{2}n\right) \\ &- r_Q(nT_m)\sin\left(\frac{\pi}{2}n\right). \end{aligned} \tag{3}$$

Equation (3) leads directly to the implementation of Fig. 5, where the samples $r_I(nT_m)$ are simply obtained from the even samples of $r(t)$ with alternating sign and the same for $r_Q(nT_m)$ from the odd samples of $r(t)$. The interpolators are needed to align in time the in-phase and quadrature samples. To avoid aliasing, we must take $f_m \geq 2B$, where B is the bandwidth of $r(t)$. More information about this procedure can be obtained from [8].

To keep input–output phase coherence, it is important that the up-conversion process in the output RF interfaces is done with exactly the same phase used in the down conversion in the input RF interfaces. For this reason, a single 8-MHz crystal clock is used to provide reference signals to the phase-locked loop (PLL) circuits. Two independent PLL circuits in the input RF card produce the 60-MHz tone for the mixers and the 40-MHz clock for the A/D converters. With this design, only the low-frequency reference signal is distributed through the backplane. This helps to prevent high-frequency signals to cause undesired interference with other modules. The 20-MHz clock derived from the frequency divider of Fig. 5 is also used for all the base-band digital signal processing within the emulator.

With regard to the receiver front-end, the band-pass filter is LC with eight sections having a Chebyshev response with low band-pass ripple and high out-of-band rejection. The mixer is a double-balanced diode mixer with high degree of linearity. The low-pass filter has a 3-dB bandwidth of 24.5 MHz and an attenuation of 65 dB at a frequency of 70 MHz.

The selected A/D converter is an ECL monolithic A/D with 10-b resolution and parallel architecture. The ECL technology has been used to cope with the high sampling rate needed in the emulator.

Finally, the interpolators are all-pass filters producing delays that are a fraction of the sampling period. They are implemented by programmable FIR filters PDSP16256/A from Plessey Semiconductors. This devices can store two different sets of coefficients and be configured to work with one set of coefficients for the even samples and the other set for the odd samples. In this way the process of multiplying by $(-1)^n$ in each branch is automatically achieved by reversing the sign of the odd coefficients of the original filter and using this set for the even samples and a set with reversed sign coefficients for the odd samples. For example, if the original all-pass filter coefficients are $\{C_0, C_1, C_2, C_3 \cdots\}$, then for the even samples we should use the set $\{C_0, -C_1, C_2, -C_3 \cdots\}$ and for the odd samples we should use the set $\{-C_0, C_1, -C_2, C_3 \cdots\}$. At 20 Msamples/s, the number of coefficients in each set is 16. The precision is of 16 bits for the input data, 12 bits for the coefficients, and 32 bits for the output data.

The separation of the even and odd 40-MHz samples at the output of the A/D is achieved by inserting two cascaded flip-flops, introducing a total delay of one sampling period (25 ns),

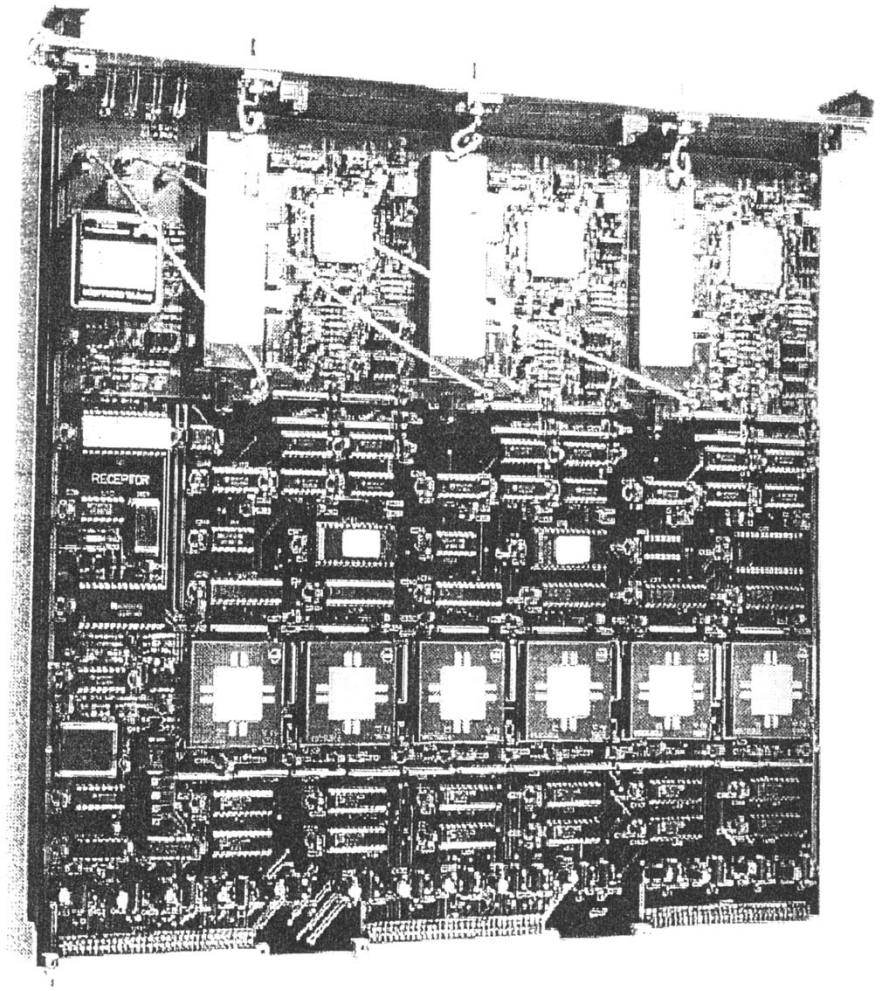

Fig. 6. Photograph of the input RF interface card.

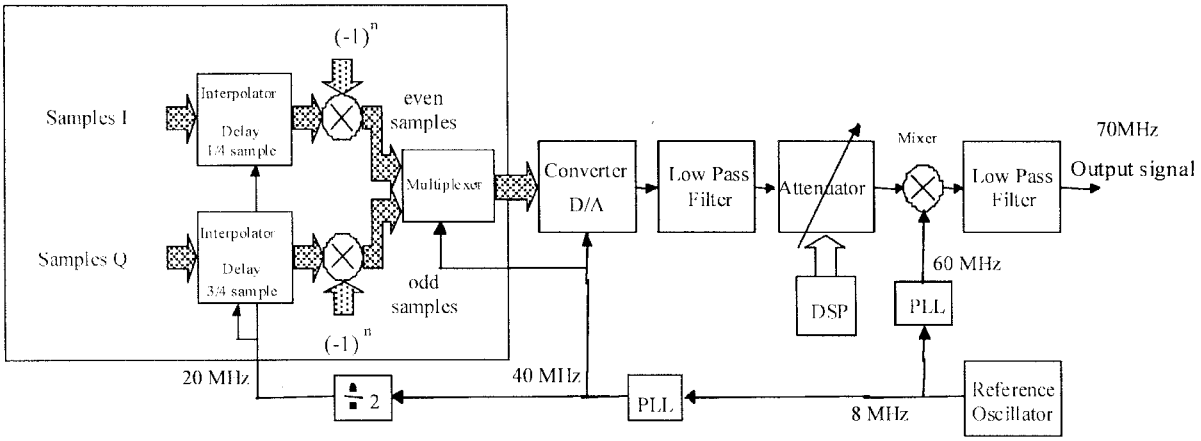

Fig. 7. Block diagram of the output RF interface.

between the A/D and the in-phase interpolator. The quadrature interpolator reads directly from the A/D output. As the FIR filters read their inputs at half rate, that is, at 20 MHz, only the even (odd) samples will be read by the in-phase (quadrature) interpolators.

In Fig. 6, a photograph of the input RF interface card is presented.

B. Output RF Interface

Two independent output RF interfaces are needed in the emulator. Both are placed in the same printed circuit board. The block diagram of a single output RF interface is shown in Fig. 7. It must be noticed that the reference clock at 8 MHz in Fig. 7 is physically the same type that appears in Fig. 5, since its signal is passed through the backplane. The PLL's, in

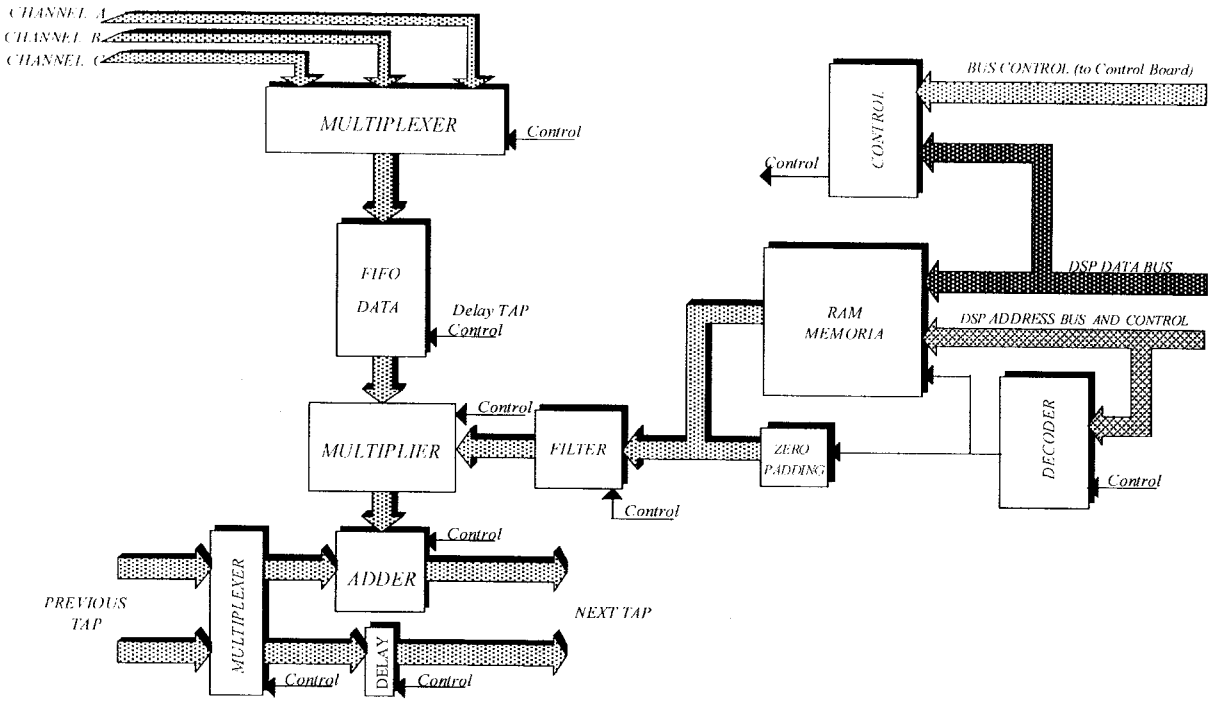

Fig. 8. Block diagram of a single tap circuit.

turn, have been replicated in order to have the high-frequency signals confined within the individual printed circuit boards.

From a conceptual point of view, the design of the output RF interface is a reversed version of the input RF interface. From Fig. 7, it can be seen that the 20-MHz I/Q samples, which have already gone through all the necessary digital processing, are passed through interpolators to delay the quadrature signal with respect the in-phase signal by half a sampling period (25 ns). Then the samples are multiplied by $(-1)^n$ in order to have them centered at 10 MHz. Since there is a delay of 25 ns between the I/Q samples, now they can be multiplexed on a single channel with double sampling rate. Then this 40-MHz samples are D/A converted to an analog band-pass signal centered at 10 MHz and finally upconverted to the 70-MHz IF carrier. Before the mixing process takes place, two cascaded DSP-controlled attenuators adjust the signal level in real time.

With respect to the hardware implementation, most of the elements are the same type that were used in the input RF interface. The D/A converter is also ECL and has 12 bits of resolution.

C. Tap Circuit Cards

The tap circuit cards are the core of the channel emulator. Each printed circuit board holds four complex tap circuits, and so we need five of these cards to implement twenty complex tap circuits. The block diagram of one complex tap circuit is shown in Fig. 8. In that figure, we can distinguish the signal path and the coefficient updating circuits. The signal path is formed by the input multiplexer, the FIFO memory, the complex multiplier, the adder, and the output multiplexer. The input multiplexer selects which of the three independent inputs of the emulator feeds this tap circuit. The FIFO memory introduces the desired delay of the signal path [τ_i in expression (1)] and jointly with the complex multiplier and the adder forms one branch of the FIR filter that models the mobile channel. Depending on the selected position of the output multiplexer, the output of the previous tap circuit is either added to the current tap output or delayed a time interval equivalent to the addition operation. The multiplexers and the FIFO are programmed by the control board following the DSP commands received through the user's interface. All the IC's placed in the signal path work at the full sampling rate of 20 Msamples/s.

With respect to the complex coefficient updating, a detailed exposition of the interpolation design can be found in Appendix B. The adopted strategy is to interpolate in real time the previously stored samples of the channel impulse response. For this reason a RAM memory and a hardware interpolator are placed in the block diagram. As is explained in Appendix B, two cascaded hardware interpolation stages are needed in the tap circuit. The first one is done by zero padding and low-pass filtering, and the second one is a zero-order hold. The first interpolation ratio is constant and equal to 32. In order to simulate different Doppler frequencies, the second interpolation ratio N is a variable power of two which ranges between 8–8192. This means that the samples at the output of the filter are kept constant during N signal sampling periods of 50 ns. This allows driving the hardware filter with a clock frequency (f_2) ranging between 2.5 MHz and 2.44 KHz. Before the simulation starts, the RAM memory in Fig. 8 is filled by the DSP with the information received from the attached PC. Then, in real time, the memory is read with a rate $f_1 = f_2/32$ and its interpolated samples feed the complex multiplier. In practice two independent RAM banks as well as two independent filters, for the real and the imaginary parts of

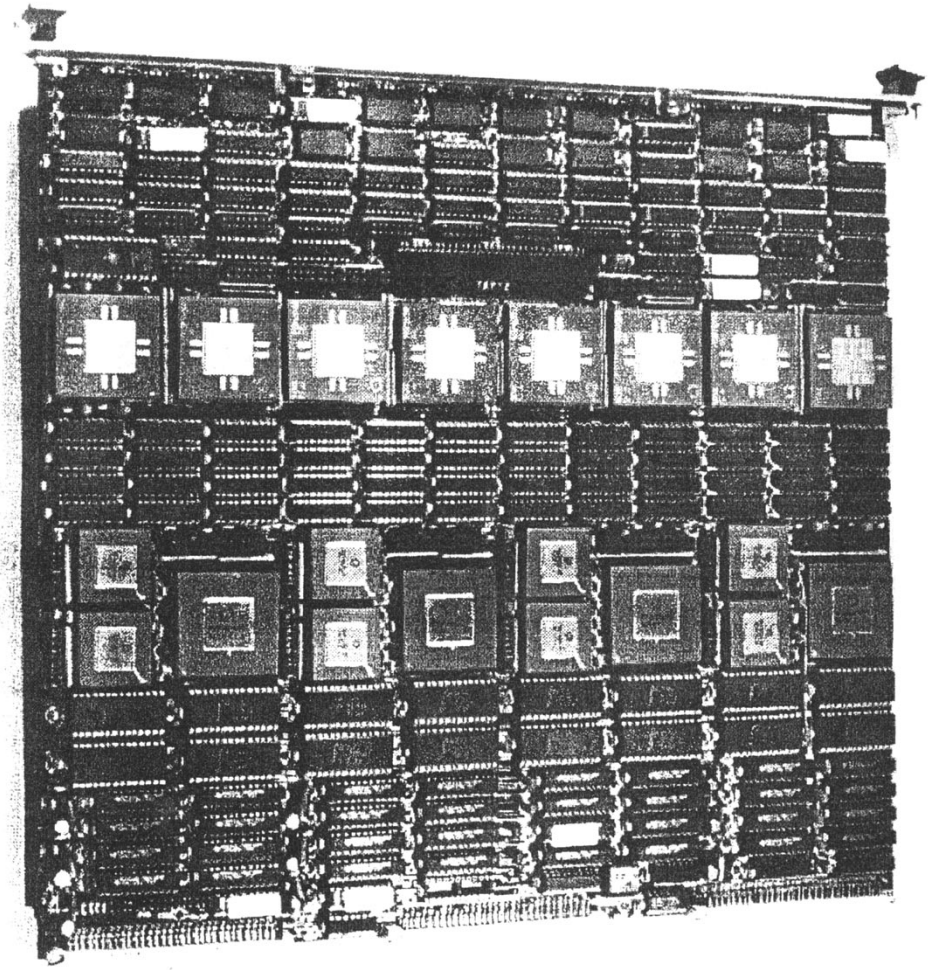

Fig. 9. Photograph of a tap circuit card.

the samples, are needed. The filters are PDSP16256 integrated circuits. The variable frequency clock that drives the filter is generated in the control card. The control card also starts and stops the real-time emulation following the DSP instructions and sets the emulator operation modes.

Fig. 9 is a photograph of one of the five tap circuit cards.

D. DSP Card

The DSP card has several purposes. First, it must hold the communications with the PC that runs the man machine interface. Second, before the emulation can start, it must interpolate the received samples of the impulse response and fill the memory banks of the tap circuits. Then it must program the FIFO memories to get the desired delays and the input/output multiplexers of each tap circuit to set the desired configuration. Finally, in real time it controls the attenuators at the two outputs to generate shadowing fading and to test the power control loops of the radio access system.

The hardware is based on a TMS320C30 DSP from Texas Instruments that runs at 33.3 MHz. It has a floating point ALU and primary and expansion buses with 32 bits for data and, respectively, 24 and 13 bits for addresses. The peripherals include ROM memory to store the programs, 1 M of 32-b RAM, and a DUART to handle serial communication with the PC. The core of the DSP software is a communications program that follows an scheme based on an ARQ protocol with four different block lengths. Each block of bits ends with a 32-b CRC to check for errors and ask for retransmission of the block (if necessary). The maximum data rate is 57 600 bps. The DSP software has been written in assembler code while the PC interface is a C program where all the user selectable items are based on friendly menus.

Fig. 10 shows a functional block diagram of the DSP card. Fig. 11 is a photograph of the whole emulator assembly.

E. Operation Modes

The channel emulator is able to emulate the following kinds of channels.

1) Dispersive Static Channel: In this operation mode the relative delay between echoes as well as the magnitude and phase of every path are chosen by the operator. During real-time emulation all these parameters are kept constant.

2) Fast Fading Channels: In this case, two possible operation modes can be selected.

Flat Rayleigh Channels: The Doppler frequency and the mean value of the received signal power are selected by the operator.

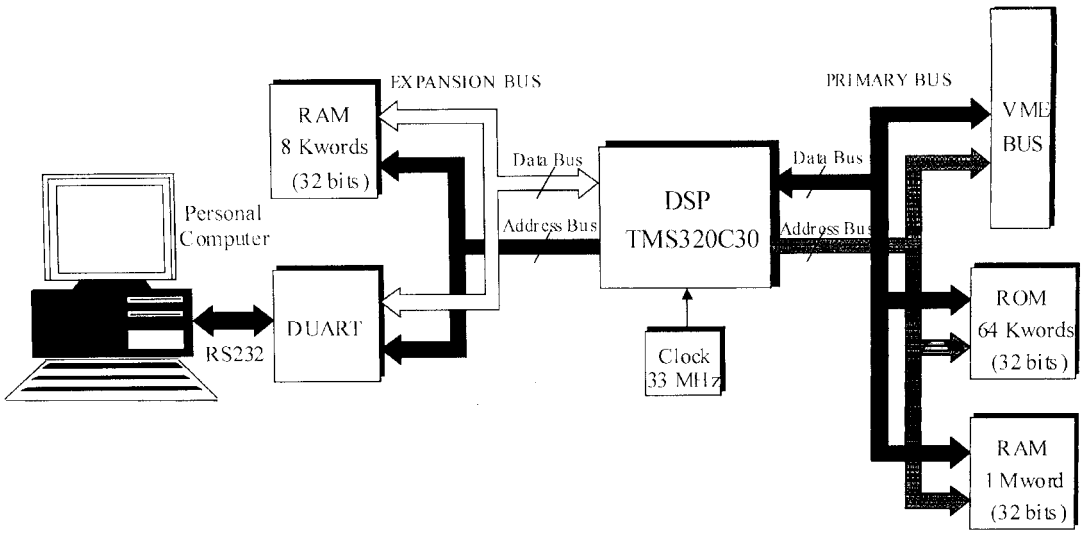

Fig. 10. Block diagram of the DSP card.

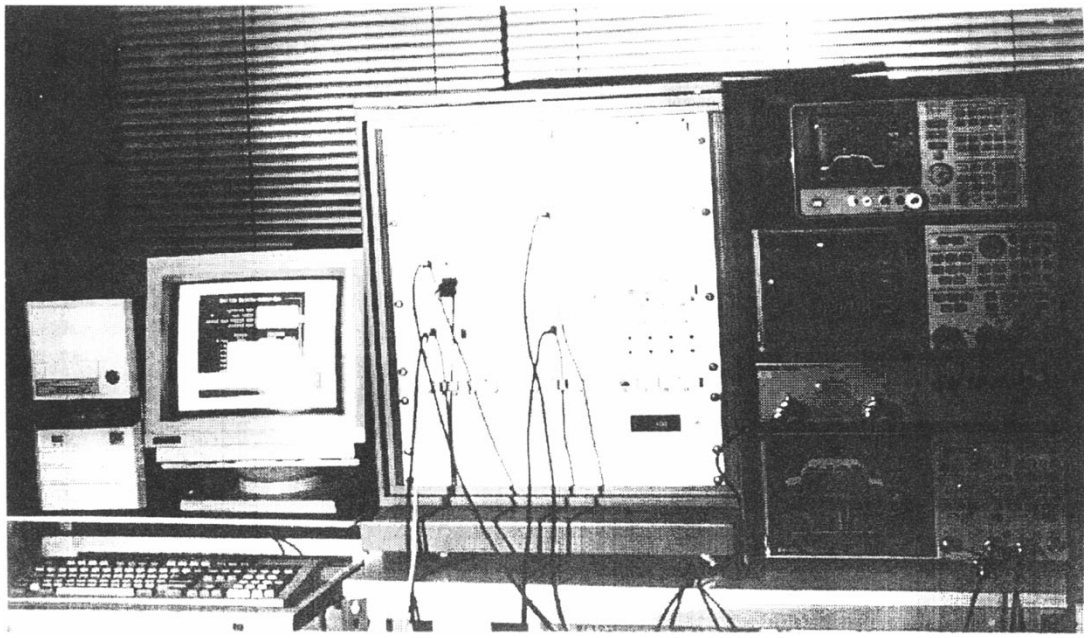

Fig. 11. Photograph of the channel emulator.

Faded Time Dispersive Channels: In this operation mode, the time evolution of the channel impulse, previously stored in the PC, is transferred to the emulator. Every set of samples of the evolution of the channel impulse response represents a different environment. The operator can select the Doppler frequency as well as the received mean signal power.

These operation modes can be freely combined, that is, some of the six independent channels that the emulator handles can be in static mode and some others in the fast fading modes.

In addition to these operation modes, the emulator is able to superpose a shadowing fading over the impulse response generated in both static or fast fading channel situations. The operator can adjust the fading rate from 0.08 to 14 Hz and the standard deviation from 0.1 to 10 dB.

The emulator can also introduce time-variant attenuation in one or both outputs in an independent way. The attenuation patterns are either step or ramp functions which are useful to test the power control systems. The user can adjust the slope of the ramp and the starting and ending instants. The attenuation range is from 0 to 43 dB.

From an operational point of view, we can distinguish two different ways of real-time scanning the RAM memories that store the channel impulse response.

Single Scan: In this case, the emulator reads once its memory contents and then it stops. This is equivalent to have a mobile station that runs a definite path within a given environment and then stops.

Continuous Mode: In this mode, the emulator continuously scans, up and down, its memory contents until it receives a

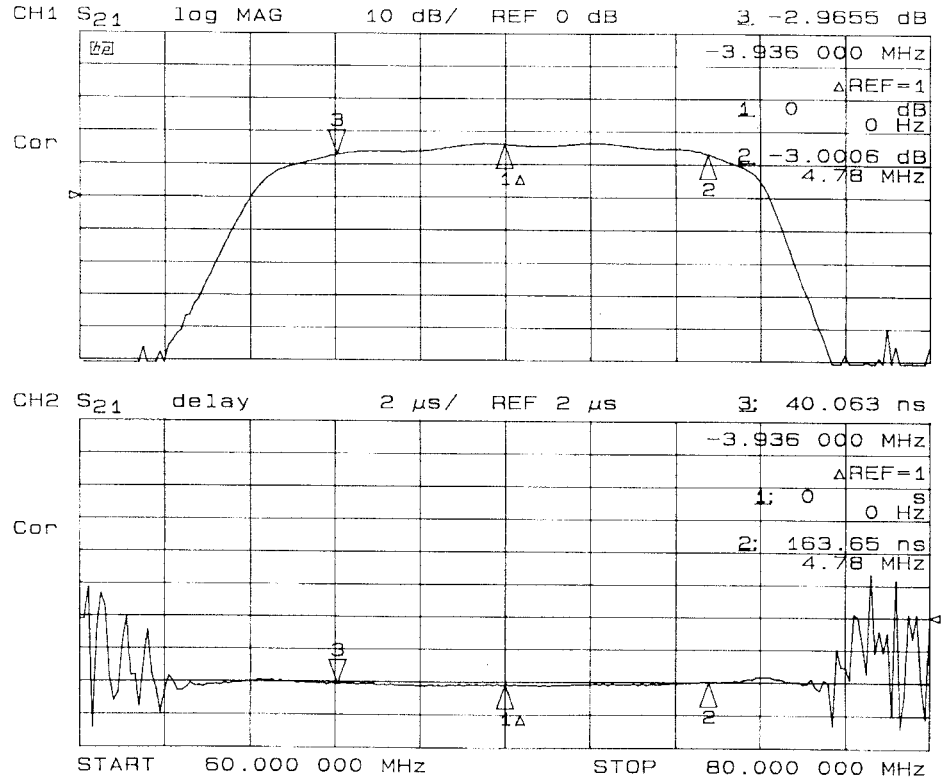

Fig. 12. Transfer function of the cascaded RF interfaces.

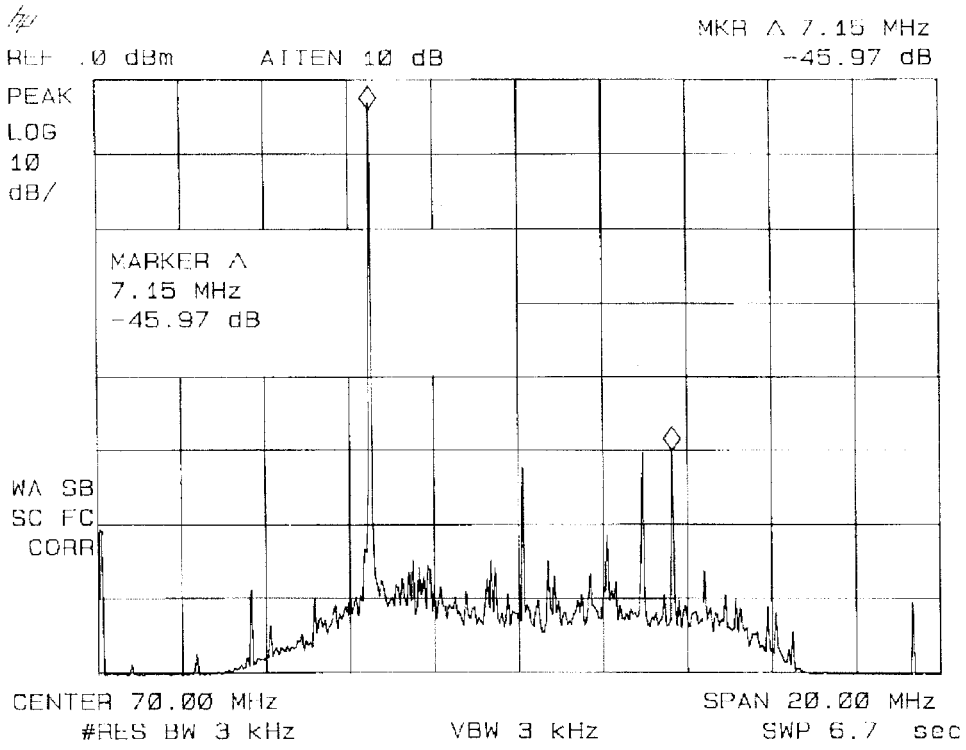

Fig. 13. Spurious signals levels measurement.

stop command from the user. This is equivalent to have a mobile station that runs a prefixed path within an environment and then it goes back following the same path, an so on. This allows a long emulation time without discontinuities in the channel impulse response that could affect the behavior of synchronization devices.

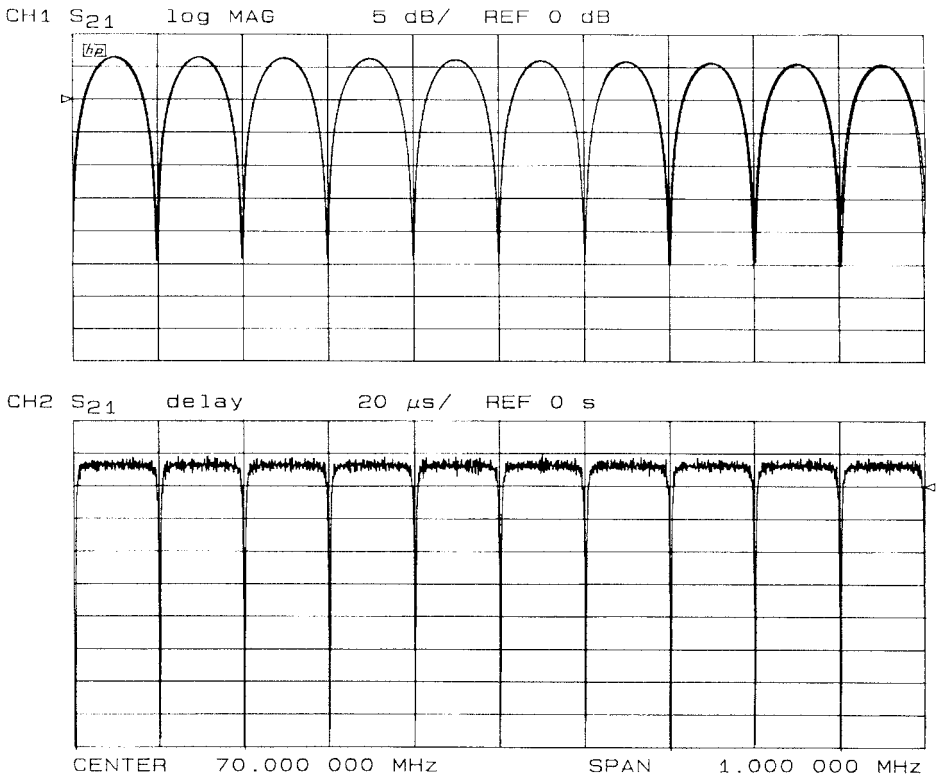

Fig. 14. Two-ray model with minimum phase fading.

IV. Results

As a result of the laboratory tests done to the channel emulator, the main characteristics and performances of the channel emulator are reported in this section. Two kinds of tests have been performed. First, the input and output RF interfaces of the emulator were linked together in order to measure its global features. The analyzed parameters include: bandwidth and group delay, input and output impedances, spurious level, carrier leakage, and I/Q image and signal to noise ratio. The second kind of tests involves the complete emulator. A network analyzer has been used to measure the simulated time-variant channel impulse response and transfer function for several test situations.

A. RF Interfaces

Fig. 12 shows a plot of the network analyzer measure of the overall transfer function considering input #1 interface cascaded with output #1 interface. Magnitude and group delay characteristics of the assembly are presented. From the magnitude plot, we can see that the 3-dB bandwidth is 8.75 MHz and with respect to the group delay plot we can verify that, within the 3-dB bandwidth, the group delay is approximately constant. Similar results have been obtained considering each one of the five remaining input–output connections.

In order to test the spurious signal levels, input #1 was again cascaded with output #1 and a −20-dBm in-band tone was injected. The output was measured by a spectrum analyzer spanning 20 MHz centered at 70 MHz. Fig. 13 shows a plot of this measure. It can be seen that the input signal frequency is 66.425 MHz. The carrier leakage, at 70 MHz, is 50 dB below the wanted output signal and the I/Q image rejection, at 73.575 MHz, is 46 dBc. The noise floor is 60 dB below the signal level and the strongest in-band spurious signal is 46 dB below the carrier. With respect to the out of band spurious signal levels (not shown in Fig. 13) all of them were found to be at least 55 dB below the carrier level.

B. Emulator Tests

The complete emulator tests were organized in five categories: two-ray model (static channel), time-variant flat fading channel, suburban mobile channel, shadowing fading test, and attenuation patterns test.

1) Two-Ray Model: This assumption provides a simple method to check the emulator ability to produce a well known transfer function. The measures can be easily compared with the theoretical predictions. A two-ray model is defined by the following impulse response and transfer function:

$$\begin{aligned} h_C(t) &= \delta(t) + \rho e^{j\phi} \cdot \delta(t-\tau) \\ H_C(F) &= 1 + \rho e^{j\phi} \cdot e^{j2\pi f r} \end{aligned} \tag{4}$$

where $\delta(t)$ is the Dirac delta function, τ is the delay between the two rays, and ρ and ϕ are, respectively, the magnitude and phase of the second ray with respect to the first one. We assume that τ and ρ are positive numbers. From expression (4) it is easy to verify that $H_C(f)$ is periodic with magnitude

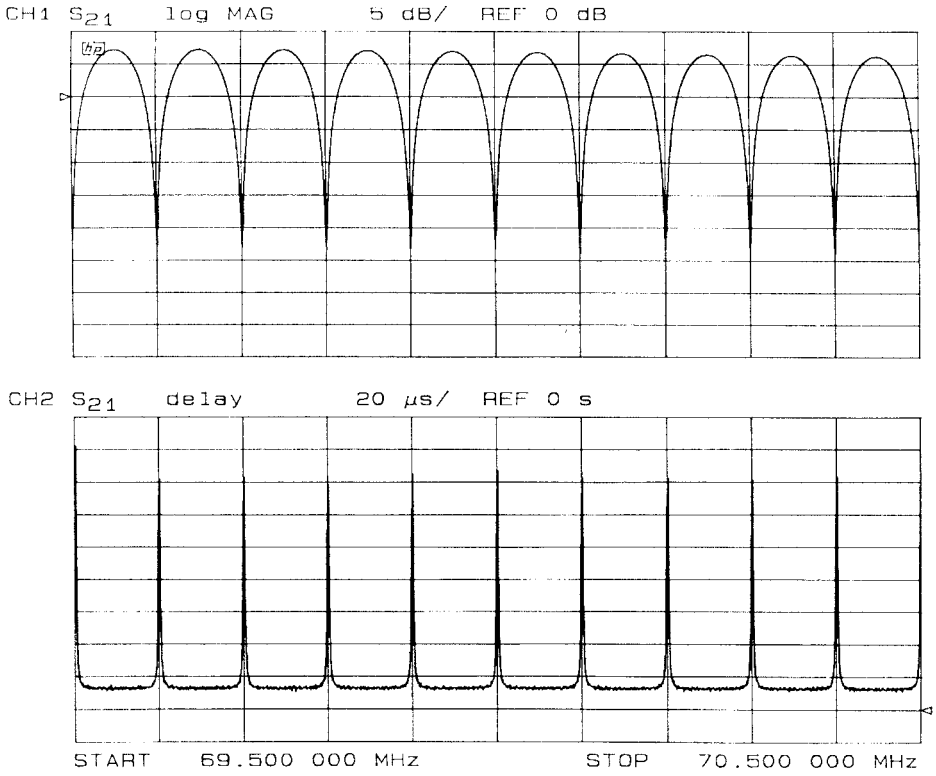

Fig. 15. Two-ray model with nonminimum phase fading.

ranging between $1+\rho$ and $|1-\rho|$ and group delay

$$\frac{\tau\rho}{\rho-1} \le t_g(f) \le \frac{\tau\rho}{\rho+1}, \qquad \text{if } (\rho<1)$$
$$\frac{\tau\rho}{\rho+1} \le t_g(f) \le \frac{\tau\rho}{\rho-1}, \qquad \text{if } (\rho>1). \tag{5}$$

The situation $\rho>1$ corresponds to a nonminimum phase channel with positive peak group delay. Figs. 14 and 15 show, respectively, the magnitude and group delay of the simulated transfer function as a function of the frequency. In both cases the delay between the two rays is $\tau = 10\ \mu$s. In Fig. 14 the second ray level is 0.5 dB below the first one $(\rho = 0.94)$ while Fig. 15 is an example of nonminimum phase fading since the second ray level is 0.5 dB above the first one $(\rho = 1.06)$. It can be verified in both figures that, in accordance with the programmed value of τ, the attenuation notches and the peak values of the group delay appear every 100 KHz. Also, in both figures the notch depth is approximately equal to 25 dB, while the maximum gain is near 5.76 dB in Fig. 14 and 6.27 dB in Fig. 15. These are in fact the values of $1+\rho$ and $|1-\rho|$ in decibels for the programmed values of ρ. Finally, it is also important to remark that the sign of the group delay peak values is in accordance with the type of channel: minimum or nonminimum phase. The theoretical values to be reached are $-156.7\ \mu$s in Fig. 14 and $+176.5\ \mu$s in Fig. 15. Approximately, these values are reached in both figures, but it must be taken into consideration that the represented group delay functions correspond to the addition of the delay of the emulated channel plus the group delay associated to the RF interfaces (mainly introduced by the filters). For this reason, the measured values do not correspond exactly with the computed ones, but they are close enough.

2) Time-Variant Flat Fading Channel: With the above test we have proven the emulator ability to simulate frequency selective static channels. However, in the mobile communications environment we also find time-variant dispersive channels. In these cases, the Doppler spectrum must be measured. Fig. 16 shows the power spectral density of the emulator output when a tone signal at approximately 73 MHz is introduced at the input. The emulator is programmed to produce a flat fading channel with a "classical" Doppler spectrum, that is [9]

$$S(f) = \frac{1}{\pi f_m}\left[1-\frac{(f-f_0)^2}{f_m^2}\right]^{1/2}; (|f-f_0| \le f_m)$$
$$S(f) = 0; (|f-f_0| > f_m) \tag{6}$$

where f_0 is the carrier frequency and $f_m = v/\lambda$ is the maximum Doppler frequency corresponding to a vehicle moving at v m/s and with $\lambda = c/f_0$. In Fig. 16, the Doppler shift is $f_m = 1000$ Hz.

3) Mobile Channel Corresponding to a Suburban Environment: This test is done to prove the emulator ability to reproduce, in real time, the wide-band mobile channel models assumed in the CODIT project. First, using high-level language simulation, a file containing the time-variant channel impulse response representative of a suburban environment was obtained. The impulse response has six complex rays with increasing delay. The delays are kept constant throughout the real-time emulation, and its values, in microseconds with

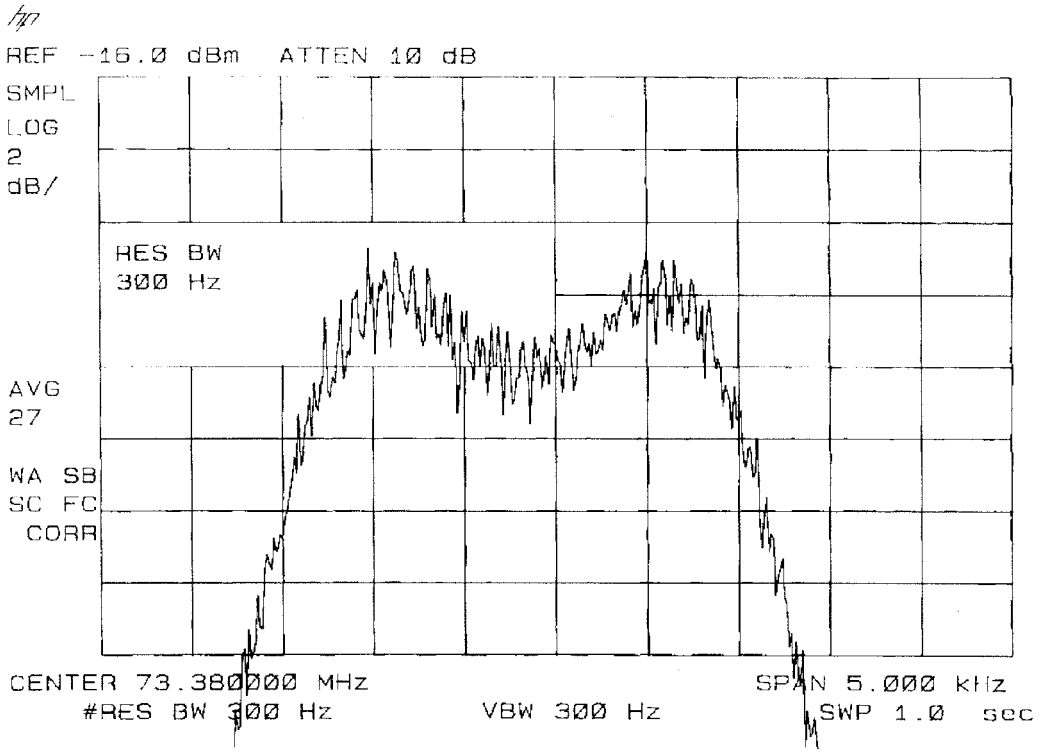

Fig. 16. Doppler spectrum of a time-variant dispersive channel.

respect to the first ray, are respectively, 0, 0.5, 5.25, 5.75, 6.75, and 9.1 μs. This impulse response file was loaded into the emulator and started to run. A network analyzer covering the whole 10-MHz bandwidth was connected to the emulator output to simultaneously measure the magnitude of the transfer function versus frequency and the magnitude of the impulse response versus time. Figs. 17 and 18 show two different moments of the time evolution of the measured magnitudes. It can be seen how both the transfer function and the impulse response are time changing, producing a wide-band channel with many attenuation notches very close in frequency that change depth very fast.

4) Shadowing Fading Test: In addition to the time-variant impulse response, the emulator has digitally controlled attenuators placed at each of the two outputs. The DSP processor controls them to simulate the effect of the shadowing fading found in real environments. The attenuation versus time function is obtained by low-pass filtering of a Gaussian white noise. Since the rate of change of the fading is very slow, the noise and the filter are implemented by software that runs in the DSP in real time. The first-order statistics of the attenuation is lognormal with a standard deviation that can be adjusted up to a maximum of 10 dB. The second-order statistics can also be adjusted by changing the bandwidth of the low-pass filter (the maximum bandwidth is 14 Hz). Fig. 19 shows an example of the generated attenuation. It was measured using a network analyzer centered at 70 MHz and set to zero span. The parameters of the example shown in Fig. 19 are a standard deviation of 5 dB and a bandwidth (fading rate) of 5 Hz.

5) Attenuation Patterns Test: In order to test the power control loops behavior, the emulator has the possibility of controlling the output attenuators to produce ramp and/or step attenuation versus time functions. The attenuation versus time functions can only be defined within one "scan interval." The scan interval is the time it takes for the emulator to read the whole memory contents. If we are in single scan mode, the whole emulation takes a single scan interval. If we are in continuous mode the emulator repeatedly reads the memory up and down until it receives the order to stop. As this is equivalent to have a mobile station that runs a prefixed path and then it goes back by the same way, the attenuation versus time function, simply takes on the same values reversed in time. In order to program the desired attenuation pattern, the user must supply four points of the function, then the DSP program will join these points with straight lines. For example, lets assume that the scan interval is equal to 3.35 s. The first point is always the attenuation at $t = 0$ (beginning of the scan). We supply a value of 0 dB for this point. Then we program a second point at $t = 1$ s, with an attenuation also of 0 dB. The third point is at $t = 2$ s. with an attenuation of 35 dB, and the fourth point is at $t = 3.35$ s with an attenuation of 10 dB. The resulting attenuation versus time function is shown in Fig. 20. This figure is a result of the measure taken with a network analyzer using zero span. Notice that, since the emulator was in continuous mode, the function is reversed in time after each scan of 3.35 s. To program a step function we should supply two consecutive points with different attenuations separated by 12.8 μs. (minimum separation). The commutation time is in fact much smaller that 12.8 μs, since it only depends on the attenuator transient response which is below 5 ns. The maximum programmable attenuation is 43 dB.

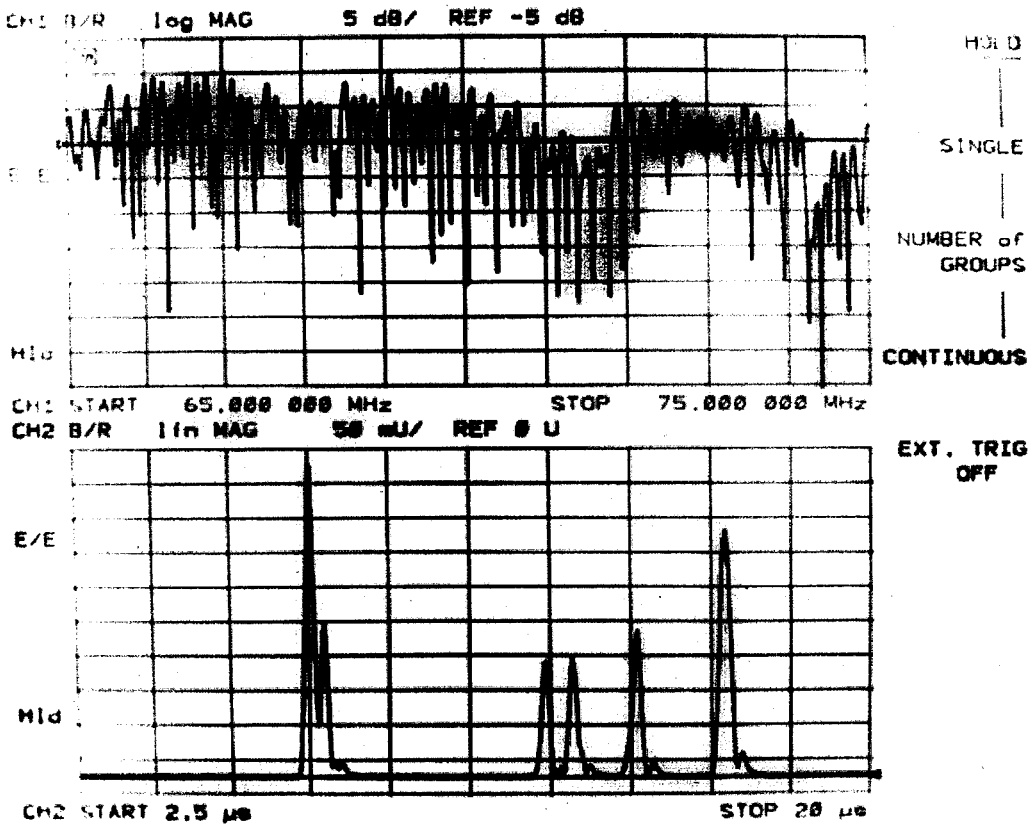

Fig. 17. Frequency and impulse response of a time-varying suburban environment channel (first snapshot).

V. Conclusions

A new wide-band real-time mobile channel emulator has been designed, built, and tested. Our efforts have been directed to achieve a flexible design suitable for testing in the laboratory the mobile communications equipment of the CODIT project. As CDMA is the multiple-access technique used in the CODIT project, it was specified that the emulator should have good delay resolution as well as long impulse response duration. It also had to be able to implement, in real time, the new wide-band propagation models developed within CODIT.

As a result, the new emulator has approximately 10-MHz RF bandwidth, a delay resolution of 50 ns, an 80-μs impulse response duration, and it can handle three independent input signals and two independent outputs. The architecture of the emulator is based on a fully configurable fir filter with time-variant coefficients. Up to six independent mobile channels can be emulated at the same time. With respect to the channel impulse response, its samples are stored in memory previously to the emulation start. To avoid using a very large memory, two cascaded interpolation processes are carried out. The design of the interpolation has been exposed in detail in Appendix B. By using interpolation we are able to operate with short files, containing the CODIT channel models for every environment, which are sent from a PC to the emulator during the set up time.

With respect to hardware implementation, the emulator is built on nine printed circuit boards connected by a VME bus. The design of the more relevant hardware blocks has been explained. Finally, the laboratory tests of the emulator have been presented. They show a good performance of the emulator and a good agreement with the specified goals.

Appendix A
Mobile Channel Modeling

Within the CODIT project a new approach for wide-band channel characterization has been used, [7]. The model is intended for the 2-GHz band assigned to FPLMTS. The environments are classified into urban, suburban, rural, suburban hilly, rural hilly, microcell (LOS and NLOS), and indoors. Each of these environments has different wide-band characteristics. For every environment a scattering function exists that relates it to a (statistical) number of scatterers located in certain positions and having some coherence characteristics, [10].

Under the WSSUS assumption, the channel impulse response will be given by (1). In (1), the contribution of each scatterer is given by the process $E_i(t)$. The time variation of the processes $E_i(t)$ is obtained as follows [11], [12]:

$$E_i(t) = a_{i0} \cdot e^{j(\varphi_{i0} - (2\pi/\lambda) v t \cos(\alpha_{i0}))} + \sum_{l=1}^{N} a_{il} \cdot e^{j\varphi_{il} + (2\pi/\lambda) v t \cos(\alpha_{il}))} \qquad \text{(A.1)}$$

where N is the number of dispersed waves that conform an scatterer (suggested values range between 30–100), v is the vehicle speed, and λ is the wavelength. The rest of parameters

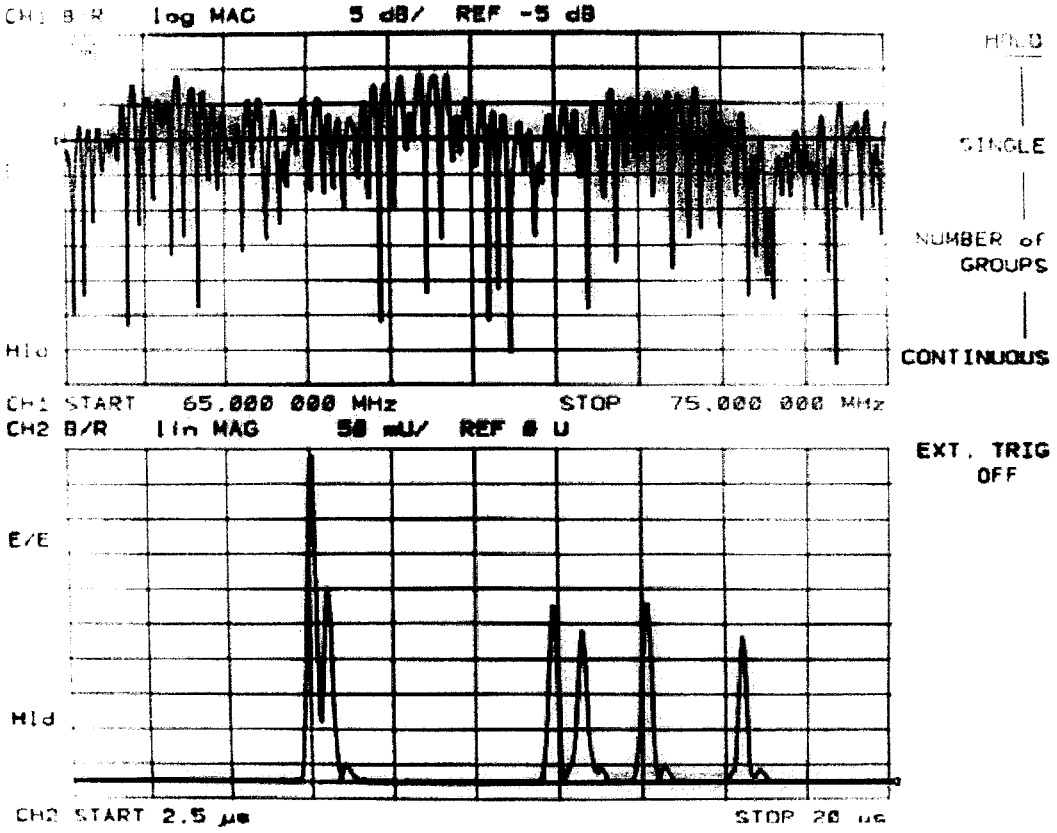

Fig. 18. Frequency and impulse response of a time-varying suburban environment channel (second snapshot).

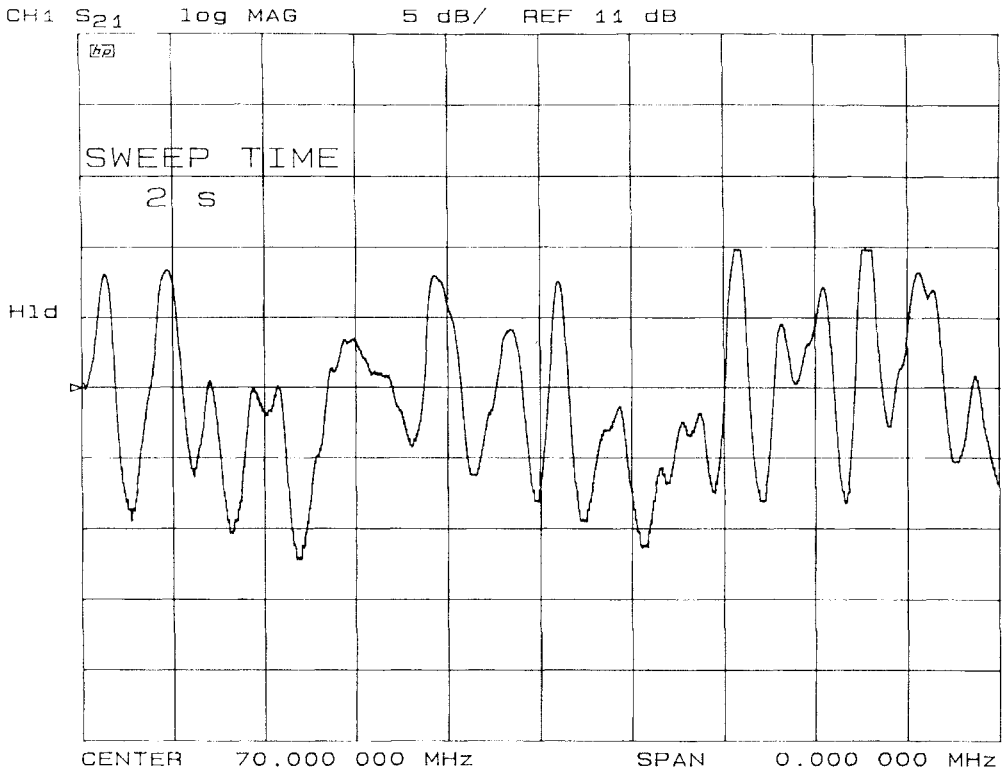

Fig. 19. Shadowing fading emulation.

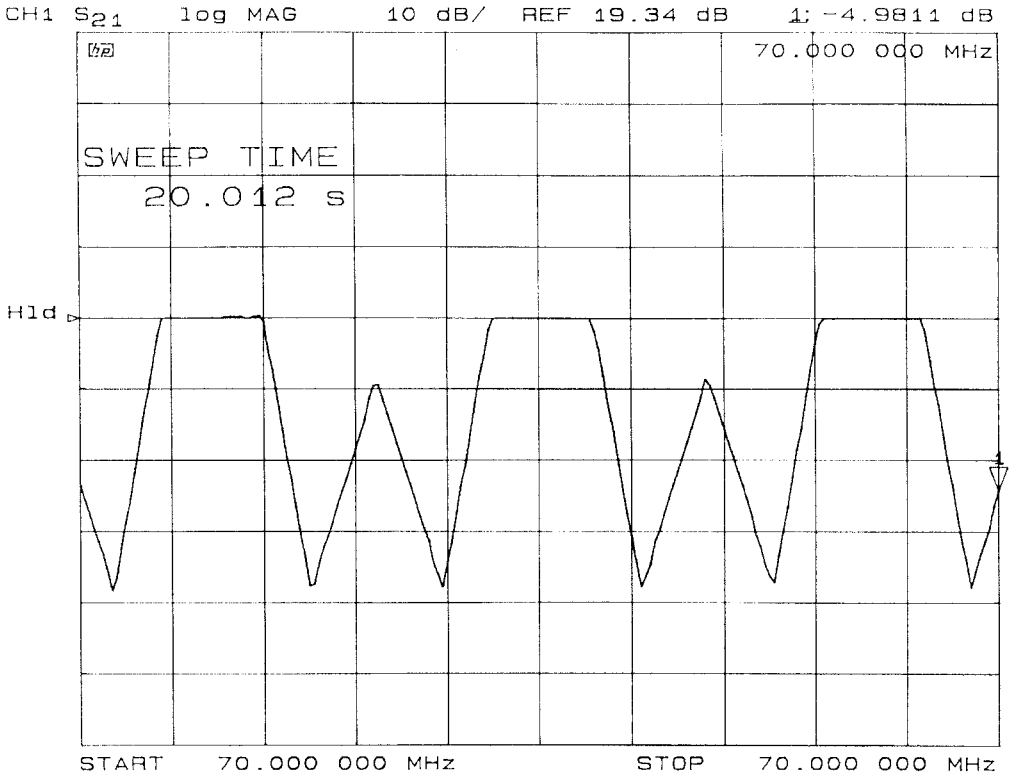

Fig. 20. Attenuation patterns to test power control loops.

in (A.1) depend on the particular scatterer. Each scatterer is characterized by the following parameters (which are, in turn, environment dependent):

- Ω mean scattered power;
- m_i "m" parameter of the Nakagami distribution which models the coherence properties of the wave produced by the i scatterer;
- α_i mean angle of arrival of the waves generated in the i scatterer (assumed Gaussian);
- $\Delta\alpha_i$ standard deviation of the angle of arrival;
- τ_i propagation delay.

So, returning to expression (A.1), a_{i0} is a constant

$$a_{i0} = \sqrt{\Omega_i \sqrt{1 - \frac{1}{m_i}}} \tag{A.2}$$

while $a_{il} (l = 1, \cdots, N)$ are zero-mean Gaussian random variables with variance

$$E\{a_{il}^2\} = \frac{\Omega_i}{N} \sqrt{1 - \left(1 - \frac{1}{m_i}\right)}. \tag{A.3}$$

The time-independent phases φ_{il} are uniform random variables in $[-\pi, \pi]$ and the angles of arrival of the waves α_{il} are assumed to be Gaussian with mean α_i and standard deviation $\Delta\alpha_i$.

APPENDIX B
INTERPOLATION DESIGN

As stated in Section II, since the I/Q signals sampling rate is $f_S = T_S^{-1} = 20$ Msamples/s, the coefficients of the FIR filters must be updated every 50 ns. Interpolation is required to avoid having very large RAM memory banks. The envisaged design is as follows.

During the setup phase the PC will send to the emulator 40 independent sets of samples (two for each complex tap) of the channel impulse response. This original samples are stored in the 1Mword RAM of the DSP card. We assume that these samples are taken at a rate of one sample for every $\lambda/n (n \geq 8)$ m of distance run by the mobile. The DSP will then interpolate the original samples, and 40 independent RAM memory banks will be filled with the interpolated samples of each tap. This interpolation is based on zero padding followed by low-pass filtering of the samples. In principle, two cascaded software interpolation stages are assumed to avoid filters with a too large number of coefficients.

During the operating phase of the emulator that is, in real-time, the 40-RAM memory banks are read and the samples pass through a new interpolation in order to generate the channel impulse response samples that feed the FIR filter. The hardware interpolation is also achieved through zero padding plus filtering. Due to limitations in the maximum allowable clock frequency, the used hardware interpolator (PDSP16256/A) cannot work at the full 20-MHz sampling rate, so we drive it with a 20M-Hz/N clock and keep constant the emulated channel impulse response during N signal sampling periods. This is a new hardware interpolation performed by passing the samples through a zero-order hold. The minimum value for N, which we will call from here on $N_{\min}$, depends on the maximum clock frequency constraint of the hardware interpolator. Moreover, we shall show that if we allow N to be

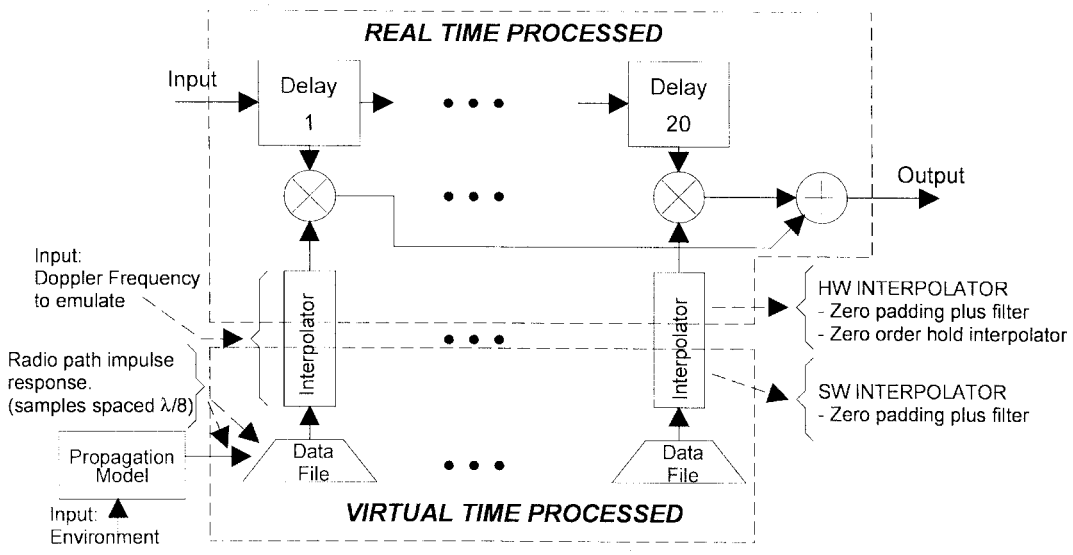

Fig. 21. Interpolation design.

TABLE I
NUMBER OF HARDWARE INTERPOLATOR COEFFICIENTS VERSUS MAXIMUM CLOCK FREQUENCY

N_{min}	$(f_S)_{max}$ $f_S N_{min}$	N_2 ; B_2T_2 (ripple 0.2dB, Atten.–90dB)			
1	20 Ms/s	16 ; -			
2	10 "	16 ; -	32 ; 0.108		
4	5 "	16 ; -	32 ; 0.108	64 ; 0.054	
8	2.5 "	16 ; -	32 ; 0.108	64 ; 0.054	128 ; 0.027

selectable, taking for example $N = N_{\min}2^k$ $(k = 0, 1, 2\cdots)$, the same contents of the RAM banks can be used to simulate a range of different Doppler frequencies. Fig. 21 illustrates the overall interpolation process.

To find out which are the key design factors we define the following notation.

Interpolation ratio of the first interpolation stage (soft)	m.
Interpolation ratio of the second interpolation stage (soft)	l.
Interpolation ratio of the third interpolation stage (hard)	p.
Interpolation ratio of the fourth interpolation stage (hard)	N.

If we call T to the time interval in which the mobile runs λ/n m (at constant speed), the simulated speed of the vehicle will be $v = (\lambda/n)/T$, and T is given by

$$T = T_S \cdot Nplm = T_2 \cdot plm = T_1 \cdot lm = T_0 \cdot m \qquad \text{(B.1)}$$

where $T_2 = N \cdot T_S$ is the hardware FIR filter clock period, $T_1 = p \cdot T_2$ is the time interval between two consecutive RAM read events, and $T_0 = l \cdot T_1$ is the output sampling period of the first software interpolation. The simulated Doppler frequency will be given by

$$f_d = \frac{v}{\lambda} = \frac{1}{nT} = \frac{f_s}{nmlpN}. \qquad \text{(B.2)}$$

We assume that the bandwidth of the original analog process W is less than $1/2T$. That is, we assume that the original samples are slightly oversampled. This is necessary for the first filter to be feasible in practice. In our design we have tried to handle a WT product as close as possible to 0.5. Once the interpolation ratios are known, we derive the parameters needed to design the filters, that is, the normalized cutoff frequencies

$$WT_0 = \frac{WT}{m} \quad WT_1 = \frac{WT}{ml} \quad WT_2 = \frac{WT}{mlp} \qquad \text{(B.3)}$$

and the normalized transition bandwidths of the software and hardware interpolators

$$B_0T_0 = \frac{1 - 2WT}{m} \quad B_1T_1 = \frac{m - 2WT}{ml}$$
$$B_2T_2 = \frac{ml - 2WT}{mlp}. \qquad \text{(B.4)}$$

The number of coefficients needed for a FIR filter to fulfill a given set of specifications can be obtained from [13]. We call N_2 to the number of coefficients in the hardware filter. For a pass-band ripple of 0.2 dB, a stop-band attenuation of

90 dB and $N_2 \geq 32$ the following inequality applies:

$$B_2T_2 \geq \frac{3.456}{N_2}. \tag{B.5}$$

Although the number of coefficients in the software filters can be chosen freely, the number of coefficients of the hardware filter depends, due to a hardware constraint, on the maximum clock frequency $[(f_2)_{\max}$, where $f_2 = T_2^{-1}]$ that we are planning to use. Table I is a summary of the constraints introduced by the interpolator. In each row of Table I we can read, for a given maximum clock frequency, the number of coefficients available in the PDSP16256 as well as the corresponding minimum values of B_2T_2 as obtained from (B.5). In fact, for $(f_2)_{\max} = 20$ Ms/s ($N_{\min} = 1$) only 16 coefficients can be used. This option is discarded, since the stop-band attenuation reduces to 40 dB. So $N_{\min}$ must be selected from the set $\{2, 4, 8\}$ and, in accordance with Table I, we must have $N_{\min} \geq N_2/16$. By substitution of (B.5) into (B.4), we find this lower bound for the global soft interpolation factor

$$ml \geq \frac{2 \cdot WT \cdot N_2}{N_2 - 3.456p}. \tag{B.6}$$

Obviously, for any two given values of WT and N_2, the hard interpolation factor p must guarantee that expression (B.6), is greater or equal than one. This is a lower bound for p, the upper bound is $p < N_2/3.456$ since this would require $ml \to \infty$. In order to have the maximum efficiency in the use of the amount of RAM available we should take the value of p as great as possible. By increasing the hard interpolation factor, the emulator simulation time (from here on t_p), which is the time that takes the emulator to read the available memory, is also increased. For these reason, and in order to simplify the hardware design, we consider that p and N_2 must take hardware fixed values and p should be a power of two.

Now, looking at (B.2) we see that, in order to adjust the Doppler frequency that we wish to simulate we have two parameters: the global software interpolation ratio (ml) and the zero-order hold interpolation ratio (N). Notice that N can be adjusted by dividing the hardware filter clock by a suitable power of two.

According to expression (B.2), once $f_s, n, (ml)_{\min}, p$ and $N_{\min}$ are fixed, there is an upper bound for the Doppler frequency that can be simulated $(f_d|_{\max})$. The minimum value that we must consider for the product $ml, (ml)_{\min}$, will be chosen in order to have $f_d|_{\max}$ close to the maximum Doppler frequency of interest ($\approx$ 1000 Hz). The maximum value $(ml)_{\max}$ will be $(ml)_{\max} = 2(ml)_{\min} - 1$, since taking $ml = 2 \cdot (ml)_{\min}$ is equivalent to take $ml = (ml)_{\min}$ and divide the clock by two. For each value of the product ml between $(ml)_{\min}$ and $(ml)_{\max}$, taking $N = N_{\min} \cdot 2^k$ $(k = 0, 1, 2, \cdots)$, we generate a set of different Doppler frequencies.

The hardware and software digital low-pass filters can be designed to have a stop-band attenuation of 90 dB. So the noise inside the signal band due to zero padding and low-pass filtering interpolation is negligible. In turn, the zero-order hold interpolation produces residual side lobes in the power spectral density of the simulated process that are not negligible. Taking into account that the signal that we are delivering to the D/A converter is a sampled version (at rate T_S^{-1}) of the output of the sample and hold device, we may calculate the in-band signal-to-noise ratio due to the rectangle interpolation as (B.7), given at the bottom of the page, where $G_x(f)$ is the spectral density of the real (or imaginary) part of one of the (up to) 20 independent complex random processes that compose a channel impulse response. In order to take into account all the design parameters, a FORTRAN program has been written.

A. Final Design

The main goal of our design is to ensure the maximum efficiency in the use of the available memory in the tap circuits. For this reason the hardware interpolation ratio must be as high as possible. We assume that the size of the available RAM memory is $R = 256$ Kwords (that is, 512 Kwords for each complex tap), and we take $WT = 0.4425$. The memory size R guarantees that t_p is always greater than 1.78 s, and the value for WT is the greatest that still allows a single-stage software interpolation, that is, m equal to one in all the cases, and with software filters with less than 512 coefficients.

In order to maximize the hardware interpolation we take $N_2 = 128$ and $p = 32$. Then it results $N_{\min} = 8$. To generate the Doppler frequencies of interest we must take $9 \leq ml \leq 17$. That is, only nine different set of coefficients are needed for the software interpolation. For $ml = 9$ we generate a Doppler frequency of 1085.07 Hz, while for $ml = 10$ we generate $f_d = 976.56$ Hz. That is, the worst resolution is about 100 Hz. For $ml = 17$ we generate $f_d = 574.45$ Hz. To produce the Doppler frequency immediately below this value we take again $ml = 9$ and $N = 2 \cdot N_{\min} = 16$. For small values of f_d the resolution is very good. The original channel information reduces to 15 420 samples (1928 wavelengths) and the signal to noise ratio is above 50 dB. The RAM access time is large enough so we can use slow static RAM. Fig. 22 shows the set of simulable Doppler frequencies.

ACKNOWLEDGMENT

The authors wish to acknowledge the support and encouragement from their colleagues at Telefónica I + D, specifically

$$(\mathrm{S/N}) = \frac{\displaystyle\int_{-1/2T_S}^{1/2T_S} G_x(f)\, df}{\displaystyle\int_{-1/2T_S}^{1/2T_S} \left\{ \sum_{i\neq 0}\sum_{k\neq 0} \mathrm{sinc}^2[(fT_S - i)N] G_x\left(\frac{(fT_S - i)N - k}{NT_S} \right) \right\} df} \tag{B.7}$$

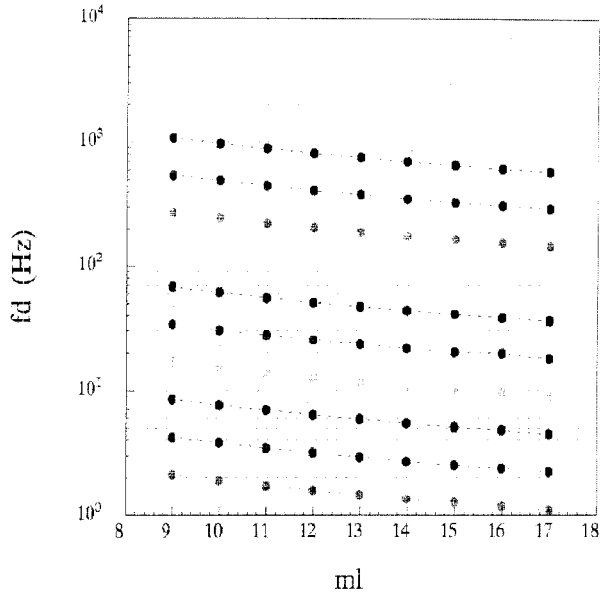

Fig. 22. Simulable Doppler frequencies with $p = 32$.

J. J. Delgado. This acknowledgment is also extended to all their colleagues within the project. The project is represented by Ericsson (Sweden and The Netherlands), Philips (Germany, U.K., and France), Ascom (Switzerland), BT (U.K.), CSELT (Italy), IBM (France, Switzerland, Germany, and U.K.), Matra (France), Telefónica (Spain), Telia Research (Sweden), and Deutsche Forschungsanstalt für Luft und Raumfahrt (Germany).

REFERENCES

[1] P. G. Andermo and G. Larsson, "Code division testbed, CODIT," in *Proc. 2nd Int. Conf. Universal Personal Commun.*, Ottawa, Ont. Canada, Oct. 1993.
[2] P. G. Andermo and L. M. Ewerbring, "A CDMA-based radio access design for UMTS," in *IEEE Personal Commun.*, pp. 48–53, Feb. 1995.
[3] A. Baier, U. C. Fiebig, W. Granzow, W. Koch, P. Teder, and J. Thielecke, "Design study for a CDMA-based third-generation mobile radio system," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 733–743, May 1994.
[4] B. Engstrpöm, S. R. Olofsson, and M. Isaksson, "A stored mobile radio channel simulator and sounder," in *Int. Conf. Signal Processing Applications and Technology*, Boston, MA, Nov. 1992.
[5] COST 207, "Digital land mobile radio communications," Final Rep., ECSC-EEC-EAEC, Luxembourg.
[6] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1989.
[7] J. Jimenez, "CODIT propagation activities and simulations models," in *RACE Mobile Telecommunications Workshop*, Amsterdam, The Netherlands, May 17–19, 1994.
[8] L. E. Pellon, "A double Nyquist digital product detector for quadrature sampling," in *IEEE Trans. Signal Processing*, pp. 1670–1681, July 1992.
[9] W. C. Jakes, Ed., *Microwave Mobile Communications*. New York: Wiley, 1974.
[10] W. R. Braun and U. Dersch, "A physical mobile radio channel model," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 472–482, May 1991.
[11] P. Hoeher, "A statistical discrete time model for the WSSUS multipath channel," *IEEE Trans. Veh. Technol.*, vol. 41, pp. 461–468, May 1992.
[12] U. Dersch, "Physical modeling of Macro, Micro and inhouse cell mobile radio channels," in *Proc. 3rd IEEE Symp. Personal, Indoor and Mobile Radio Comm.*, Boston, MA, Oct. 1992.
[13] J. G. Proakis and D. G. Manolakis, *Introduction to Digital Signal Processing*. New York: Macmillan, 1988.

Juan J. Olmos (M'91) was born in Palma de Mallorca, Spain, on December 6, 1956. He received the Engineer of Telecommunication and Ph.D. degrees from the Universitat Politécnica de Catalunya (UPC), Catalonia, Spain, in 1983 and 1987, respectively.

In 1983, he joined the Escola Técnica Superior d'Enginyers de Telecomunicació de Barcelona, Spain, where he became an Associate Professor in 1990. He has been working in the field of digital radio, both fixed radio relay and mobile communications. He participated in the European research programs COST231 and RACE and is currently working in the ACTS program. His current research interests are personal, indoor and mobile radio communications systems, and wireless LAN.

Antoni Gelonch (M'97) was born in Juneda, Spain, on July 20, 1964. He received the Engineer and Ph.D. degrees from the Universitat Politècnica de Catalunya (UPC), Catalonia, Spain, in 1991 and 1997, respectively.

In 1991, he joined the Signal Theory and Communications Department, UPC, where he became an Assistant Professor in 1992 and an Associate Professor in 1997. He has been working in the field of digital mobile radio communications. He participated in the RACE (CoDiT) European research programs and now is working in the ACTS RAINBOW program.

Fernando J. Casadevall (M'86) received the Engineer of Telecommunication and Ph.D. degrees from the Escola Tecnica Superior d'Enginyeria de Telecomunicacio de Barcelona (ETSETB), Universitat Politècnica de Catalunya (UPC), Catalonia, Spain, in 1977 and 1983, respectively.

In 1978, he joined the ETSETB, where he was an Associate Professor from 1983 to 1991. He is currently a Full Professor in the Signal Theory and Communications Department, UPC. After graduation, he was concerned with equalization techniques for digital fiber optic systems. He has also been working in the field of digital communications with particular emphasis on digital radio and its performance under multipath propagation conditions. In the last ten years, he has mainly been concerned with the performance analysis and development of digital mobile radio systems. In particular, his research interests include cellular and personal communication systems, multipath receiver design, and digital signal processing. He is actively participating in the European research programs COST259 and ACTS.

Guillem Femenias (M'91) was born in Petra, Spain, on August 20, 1963. He received the Engineer and Doctor Engineer degrees in telecommunication engineering from the Universitat Politècnica de Catalunya (UPC), Catalonia, Spain, in 1987 and 1991, respectively.

From 1987 to 1994, he was a Researcher at UPC, where he became an Associate Professor in 1990. In 1995, he joined the Departamento de Ciències Matemàtiques i Informàtica, Universitat de les Illes Balears (UIB), Balearic Islands, Spain. His current research interests and activities span the fields of digital communications theory with applications and wireless personal communication systems, with particular emphasis on trellis-coded modulation, turbo coding, and wireless ATM systems.