A Software Radio Platform for New Generations of Wireless Communication Systems

Christian Bonnet, Giuseppe Caire, Alain Enout, Pierre A. Humblet, Giuseppe Montalbano, Alessandro Nordio, and Dominique Nussbaum

Institut Eurécom,*
B.P. 193, 06904 Sophia–Antipolis CEDEX, France
Tel: +33 4 93 00 26 08, Fax: +33 4 93 00 26 27
E-mail: firstname.name@eurecom.fr

Abstract With the proliferation of mobile communication systems, the implementation of Software Defined Radio terminals able to reconfigure themselves to handle several different standards has become a key issue to provide universal seamless connection to the users (Software Radio is a very broad term involving several levels in the protocol stack, see e.g. [1], [2], [3] and references therein).In this context Eurécom and EPFL have started a joint project whose objective is to study and implement a real-time software radio communications platform to validate advanced mobile communication signaling algorithms. Due to various practical design issues, the platform implements the essential physical layer features of the TDD transmission mode of the UMTS standard (air-interface and signal processing) although the Frequency Division Duplex (FDD) mode and even other standards could also be implemented. The platform is a real-time PC based system that gives access to, and allows experimenting with, wide band radio resources. It will be able to emulate both a powerful base station with smart antennas and a simpler mobile terminal with a single antenna. It provides functionality at three levels: hardware, DSP software, and link level software. In this paper we address the major issues related to the design of a real-time software radio system. Within this framework we provide a general description of the platform architecture and functionalities. A special emphasis will be devoted to the description of the signal processing techniques adopted to implement transmission and reception software front-ends. Some indications on the demonstration setting will also be provided. Finally, near future perspectives for the platform evolution will be addressed.

1 Introduction and Motivation

The presence of several different wireless communication standards and the wide variety of services provided by mobile communication operators have posed the prob-

*Eurécom's research is partially supported by its industrial partners: Ascom, Swisscom, Thomson-CSF, IBM France, CEGETEL, Motorola, France Telecom, Hitachi Europe and Texas Instruments

lem of providing universal seamless connection to customers with different service requirements or simply moving through different countries. Software defined radio terminals able to reconfigure themselves to handle several different standards and different services represent a solution to this problem. Also motivated by the intensive world wide activity around the third generation mobile communication systems, Eurécom and EPFL (École Polytechnique Fédérale de Lausanne) have started a joint project with the objective of designing and implementing a real-time software radio communication platform to validate advanced mobile communication signal processing algorithms. The platform is characterized by the following major features:

- Flexibility, achievable by a software driven system.
- Duplex communication.
- Multiple antennas transmit and receive signal processing (i.e. joint spatio-temporal signal processing).

Flexibility remains a key word for a software defined system. In our case it serves on several purposes. For instance one may perform propagation channel measurements and transmitter characterization, evaluate the performance of different signal processing algorithms for both single user and multi-user systems under different operating conditions. *Duplex communication* is also necessary to allow higher layer protocol services, and to analyze more complex system aspects, such as multi-access and power control, and optimize down-link signal processing from up-link measurements. The platform will allow *Multiple antennas signal processing*, or more generally, spatio-temporal signal processing (also known as array processing) since this a very promising ensemble of techniques able to significantly increase the capacity of wireless communication systems. Indeed, the possibility of exploiting both the spatial and the temporal dimensions, jointly when possible, adds further diversity orders to the pure temporal processing improving the capability of canceling the interference and rejecting the noise. Clearly the more the aside information about the user's channels and their statistical properties the more effective these techniques can be.

The main focus of this paper is on the description of the basic platform architecture and signal processing to implement the essential physical layer level procedures of the UMTS standard operating in Time Division Duplex (UMTS/TDD) mode.

At the physical layer level, Software Radio generally requires the development of signal processing algorithms suited to implementation on a general purpose programmable processor (as opposed to analog or digital dedicated hardware). For this reason the position of Analog to Digital (A/D) and Digital to Analog (D/A) conversion must be moved as close as possible to the antennas. The goal is to perform operations like *channel selection* [4, 5], synchronization and detection in the all-digital domain, by using high performance Digital Signal Processors (DSPs). Special care must be dedicated to the transmission and reception front-end architecture. In fact, this should be independent on system-dependent parameters like the signal bandwidth and the symbol (or chip, in a CDMA system) rate. When designing software radio algorithms it is assumed that a multi-band Radio Frequency (RF) section takes care of translating the desired signal from a fixed Intermediate Frequency (IF) carrier to

the required RF carrier, and vice-versa. Then we are concerned with the efficient generation of an IF analog signal from a baseband digital signal (transmitter front-end) and with the reverse operation (receiver front-end). Once the end-to-end transmitter receiver architecture is defined the next step consists in synchronizing the receiver, estimating the channels associated with the users, and eventually detecting the transmitted symbols. All these operations are performed in real-time over our platform.

The adopted modulation scheme (BPSK and QPSK), the spread signal bandwidth (5 MHz), the spreading gain (DS-CDMA with allowed spreading gain of $4, 8, 16$) and the frequency band (around 2.1 GHz) are the same as the one of the UMTS/TDD. The TDD mode has been chosen because under certain circumstances it allows the exploitation of the channel reciprocity between up-link and down-link in duplex communication, and also reduces the DSP computational load for a given bandwidth.

The solutions proposed in this paper can also be applied to a wide class of linearly-modulated digital signals (including most today's and future mobile communications standards, like GSM, IS-54, IS-136, IS-95, and DECT (see e.g. [6] and references therein), UMTS (both FDD and TDD modes) [7], CDMA2000 (see e.g. [8], [9]) and EDGE (see e.g. [10] and references therein)).

2 Platform architecture

In a first phase, the development of the platform has been based on a single antenna architecture for the Mobile Terminal (MT) as well as for the Base Station (BTS). The retained architecture permits to enhance easily the capabilities of the platform without redesigning the essential hardware and software parts.

The BTS and the MT are based on a similar hardware platform. The different hardware parts of the platform are highly partitioned in order to be able to use as many standard cards and components as possible. The platform is divided into two subsystems: a signal processing subsystem and a radio subsystem. The signal processing subsystem is comprising

- a reconfigurable data acquisition system based on Field Programmable Gate Array (FPGA) and PCI bus technology
- a PCI bus-based DSP system employing a combination of embedded DSPs and workstations
- data management software (data routing, framing, synchronization)
- signal processing software (digital transceiver algorithms, multiple-access protocols, error coding/decoding)

These elements may be replicated in a parallel fashion to allow for the possibility of multiple-antenna systems (both at the BTS and mobile terminal).

The radio transmitter/receiver subsystem is composed of

- a radio module that interfaces the signal processing system at IF level with the antenna

- a clock frequency generation card that distributes clock signals references to the whole system

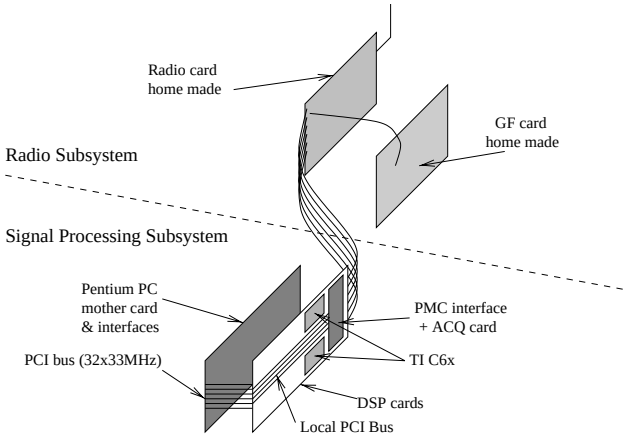


Figure 1: Mobile Terminal architecture

Figure 1 depicts the actual hardware setup:

- a radio card capable to handle a 5MHz bandwidth radio signal
- an Analog/Digital Digital/Analog conversion card (ADAC)
- a data acquisition card (ACQ) that is put as a mezzanine card (PMC) via a local PCI bus
- a 2-processor DSP card based on (Texas Instruments) TI C6x technology that supports the signal processing at chip level
- a Pentium PC card that supports the processing at symbol level and the higher protocols (main CPU)

The target architecture of the BTS (figure 2) will include the following elements:

- 8 radio frequency cards
- a clock and frequency generation card (GF card)
- 8 data acquisition cards (ACQ) with 8 A/D D/A cards (ADAC)
- 8 DSP cards

It is important to notice that the same architecture can be used for different applications and according to different modes of operation. These flexibility is achieved by using as much as possible a software approach. Software is playing an essential role at each stage of the digital processing chain: at the acquisition level via the use of programmable FPGAs as well as at the signal processing level via the use of DSPs. Passing from one mode to another is a matter of downloading the corresponding support software in the different hardware components.

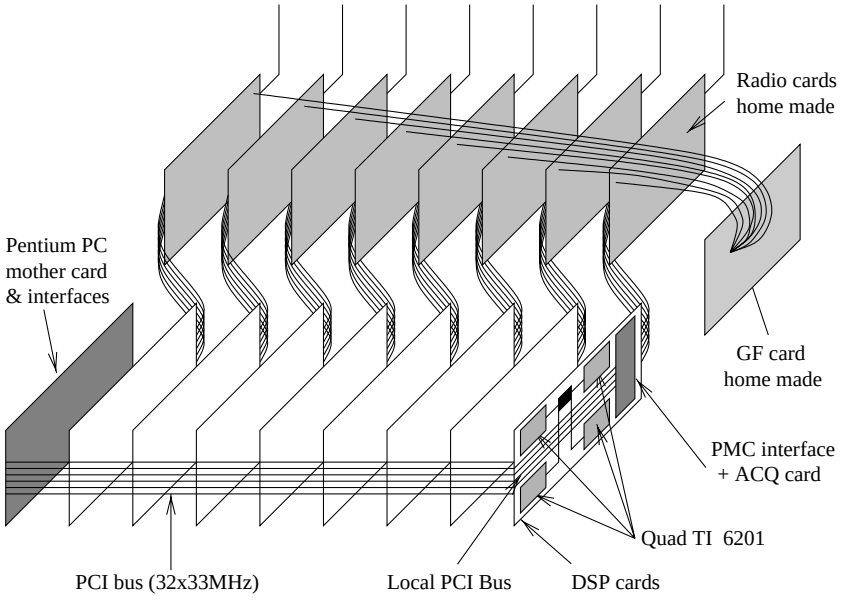


Figure 2: BTS structure

3 Operating modes

As mentioned before, the demonstrator should be capable of implementing the essential aspects of UMTS (in TDD mode). However, some others functionality will be developed (other systems), with the same hardware. This implies that we have to imagine several ”operating modes”. Operating modes are relevant to specific applications (for example UMTS), but not focussed exclusively. The modes 1 and 2 are real time, mode 3 is a recording mode, and modes 4 and 5 are specialized modes.

3.1 Operating mode 1: Real time signal pre-detection (UMTS like processing)

In this mode, each DSP card implements the receiver for its own antenna and provide a symbol-rate sequence of samples output from the receiving filter. Low-rate output sequences are transferred to the main PCI bus (This is possible if the aggregate data rate of the 8 receivers is less than the bus throughput, which should be true for many applications). This mode does not allow joint space-time processing, in the sense that each DSP card process only the output of its own antenna. In the literature [11], this approach is called pre-detection, as opposed to post-detection, in which the outputs of the antennas are combined at the chip rate. In the pre-detection approach, further interference cancelation and antenna combining can be done at the symbol rate, by the main CPU, possibly not in real time. On the transmitter side, the symbols are transferred by burst from the main CPU (Pentium) to the DSPs, which provide the

samples that are sent to the Acquisition Card. This mode allows to handle essential aspects of UMTS in real time.

3.2 Operating mode 2: Real time processing for narrowband signals

This operational mode is very similar to the first one. On the receiver side, each DSP card process the output of its own antenna. Instead of doing pre-detection, such as matched filtering, we do only low-pass filtering and re-sampling, and we transfer the output sequences from all the cards on the main PCI bus. This applies to the case where we want to do space-time joint processing with narrowband (i.e., non-full 5 MHz) bandwidth signals. The bandwidth can be chosen arbitrarily, so it is possible to implement in real time arbitrary processing in the Host. For example, the DSPs can be used to implement a low-pass filter and to down-sample the signal so that the aggregate data rate can be fit on the PCI main bus. In this mode, it will be possible to receive existing real signals (i.e. GSM, FM).

3.3 Operating mode 3: Non-real time processing (recording mode)

In this mode, we use the memory on the DSP cards to record a burst of the received signal sampled at rate f_s synchronously from all the antennas in parallel. When the signal is recorded for the desired length (typically one second), the main control unit transfers in a given ordered way the content of the memory of all cards one after the other to a file or to the main PC memory, for further non-real time arbitrary processing. This mode allows non-real time joint space-time processing, since the signals are recorded synchronously in parallel from all the antennas before any filtering and pre-detection by the DSPs. Also, this mode basically does not use the DSPs and could be implemented on a PC equipped only by the RF and ACQ cards, provided that the ACQ cards are equipped with large on-board memory.

3.4 Operating mode 4: Direct partial data transfer with non-real time processing

If we need to record longer blocks of signal, so that the on-board memory is not sufficient, this third mode will allow to transfer data from the ACQ, sampled at rate f_s, directly on the main PC memory via the main PCI bus. This method will allow long-term channel measurements (e.g., the time-evolution of a fading channel over, say, more than 1 second interval).

3.5 Operating mode 5: Hardware simulator

We must be able to use the considerable computational power of the DSP cards to implement powerful simulations, without actually transmitting and receiving signals. In this operational mode, data are transferred from the main PC memory to the DSP cards, and vice versa, in a non-real-time way, and we exploit the DSPs just as computers, to test offline our algorithms without bothering with transmission/reception.

3.6 Synthesis

Obviously, the above modes have various levels of importance, since we focus on real time processing, and essentially with UMTS parameters (i.e. mode 1). However, the other modes are also important, since it allows a more important scope of applications in research (off-line processing with arbitrary complexity), education and partnerships. They are also used for tests and debugging.

4 End to end signal processing

This section gives an overview of some theoretical principles on which the platform software has been implemented. We start the analysis considering the transmitted data flow (for example a video stream) already coded and mapped in the QPSK or BPSK alphabet.

4.1 Transmitter

Let $a[k]$ denote a sequence of symbols belonging to the QPSK alphabet [11], let $\psi(t)$ denote the symbol pulse-shaping, bandlimited over $[-W/2, W/2]$, and T the symbol interval. The corresponding continuous-time complex base-band equivalent linearly modulated signal is given by

$$x(t) = \sum_k a[k]\psi(t - kT) \tag{1}$$

Several cases of interest can be expressed in the form (1). For Direct-Sequence CDMA system [11], using a spreading gain N, T should be interpreted as the chip interval and $a[k] = b[\lfloor k/N \rfloor]c[k]$, where $b[m]$ is the m-th modulation symbol and $c[k]$ is the k-th chip and N is the spreading gain (this generalizes trivially to systems with several spreading layers, like IS-95).

In general, for digital transmitters, the signal $x(t)$ is the output of a D/A converter which takes as input the discrete-time signal

$$x[n] = x(n/f_s)$$

with sampling frequency $f_s \geq W$. In classical I-Q modulators, the continuous base-band components $\mathrm{Re}\{x(t)\}$ and $\mathrm{Im}\{x(t)\}$ are generated by low-pass filtering the output of two separate D/A converters, and the IF signal

$$y(t) = \mathrm{Re}\{x(t)\exp(\mathrm{j}2\pi f_{\mathrm{IF}}t)\} \tag{2}$$

is produced by mixing $\mathrm{Re}\{x(t)\}$ and $\mathrm{Im}\{x(t)\}$ with IF carrier signals in phase quadrature and by summing the modulated real signals [11]. This approach requires two D/A converters, two low-pass filters, two analog mixers and one adder.

Another approach consists of producing a sampled version of the IF modulated signal $y(t)$ by using a sampling rate f_s greater than $2f_{\mathrm{IF}}$. The continuous-time signal is obtained by bandpass filtering the output of a single D/A converter. This solution

(for the receiver front-end) is described in [12] and [13]. However, since intermediate frequencies usually range between tens of MHz up to 100 MHz, this approach is extremely computationally intensive as it requires the generation of signal samples and the multiplication by the carrier signal at extremely large sampling frequency.

Next, we propose a transmitter front-end architecture that allows a working sampling frequency of the order of the baseband signal bandwidth (and not of the order of the IF carrier), no explicit multiplication by the carrier signal and a single D/A converter and analog filter centered at f_{IF}.

4.1.1 Transmitter architecture

We choose the sampling rate f_s according to the classical expression

$$f_s = \frac{f_{\mathrm{IF}}}{\ell \pm 1/4} \qquad \text{for a positive integer } \ell \tag{3}$$

Then, we generate the discrete-time real signal

$$x'[n] = \mathrm{Re}\{x[n]\mathrm{e}^{\mathrm{j}2\pi(f_{\mathrm{IF}}/f_s)n}\} = \mathrm{Re}\{\mathrm{j}^{\pm n}x[n]\} \tag{4}$$

In this way, the periodic spectrum of $x'[n]$ shows a spectrum replica centered at f_{IF} (see figure 3(a)). After ideal D/A conversion, a pass-band filter centered at f_{IF} removes the other replicas, generating the desired IF modulated signal.

The discrete-time modulation by $f_s/4$ in (4) requires a negligible computational cost since it corresponds to change alternatively the signs of $x[n]$ as can be noticed in expanding equation (4). In order to avoid aliasing when taking the real and imaginary part, the sampling rate must satisfy also the condition $f_s \geq 2W$.

4.1.2 D/A Conversion

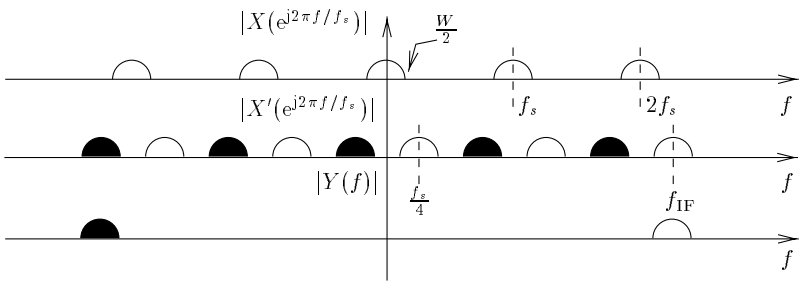
In the above description we assumed an ideal D/A converter with flat frequency response. Real D/A converters have an impulse response $p(t)$ that can be approximated as a rectangular pulse of duration $1/f_s$ with frequency response of the form of $\mathrm{sinc}(f/f_s)$. If the discrete-time input of the D/A converter is the pass-band signal $x'[n]$, then the spectrum of the output signal is given by

$$Y(f) = \sum_i X'(f - if_s)\mathrm{sinc}(f/f_s) \tag{5}$$

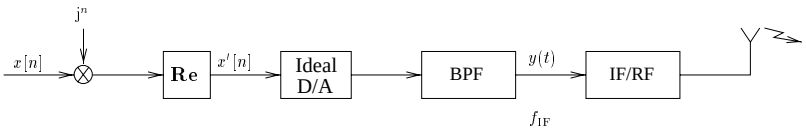
where $X'(f)$ is the discrete-time Fourier transform of $x'[n]$, defined by

$$X'(f) = \sum_n x'[n]e^{\mathrm{j}2\pi nf/f_s}$$

Hence, the spectrum replica located at frequency f_{IF} is attenuated and distorted by the D/A impulse response as shown in figure 4(a). A way to reduce the attenuation consists of using a D/A converter working at rate $f_d = L_{\mathrm{D/A}}f_s$, where $L_{\mathrm{D/A}}$ is a suitable integer, and up-sampling $x'[n]$ by the factor $L_{\mathrm{D/A}}$. By choosing $L_{\mathrm{D/A}}$ such



(a) Spectrum of $x[n]$, $x'[n]$, and $y(t)$ with the integer $\ell = 2$ and the sign $(+)$ chosen in (3)



(b) System model

Figure 3: Alternative approach for IF up-conversion

that $f_d \gg f_{\rm IF}$, the spectrum replica around $f_{\rm IF}$ falls inside the first lobe of the D/A frequency response since its first zero is located at f_d (see figure 4(b)). Moreover it is possible to compensate in part the distortion of the D/A converter by introducing a pass-band FIR filter between the up-sampler and the D/A converter. The filter must be designed in order to enhance the spectrum replica at IF while attenuating the other replicas.

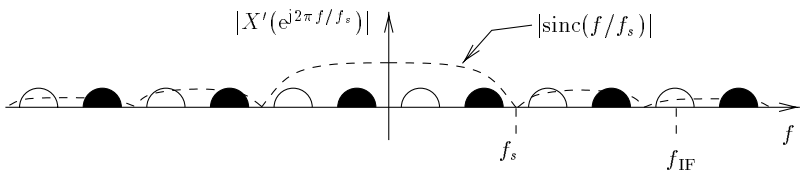
Low complexity implementation. Denote by $x''[n]$ the up-sampled version of $x'[n]$, given by

$$x''[n] = \begin{cases} x'[k] & \text{for } n = L_{\rm D/A}k \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

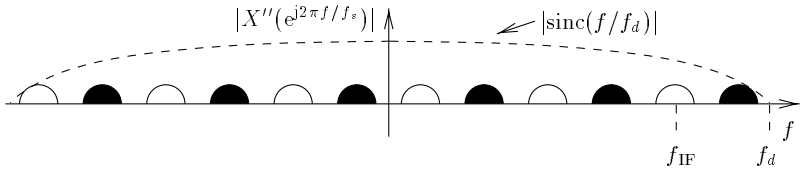
and by $h_{\rm D/A}[n]$ the filter impulse response. The filter output is given by

$$x'''[n] = \sum_m h_{\rm D/A}[m]x''[n-m] \tag{7}$$

If the filter impulse response has length $L_{\rm D/A}$, there is only a single non-zero term in the sum in the RHS of the above equation. Then, $x'''[n] = h_{\rm D/A}[m]x'[k]$ where $k = \lfloor n/L_{\rm D/A} \rfloor$ and $m = n$ modulo $L_{\rm D/A}$. Therefore, the computational cost of the filtering operation consists of one real product per output sample at rate f_d. Since f_d is large, this might still be too complex for practical applications. Then, the filter coefficients can be constrained to be in the set $\{0, \pm 1\}$. In this way products are avoided (change of sign comes basically at no cost). The problem of optimizing



(a) D/A frequency response without up-sampling



(b) D/A frequency response with up-sampling

Figure 4: D/A frequency response

the FIR filter reduces to exhaustive search for the best frequency response over all possible filter vectors of length $L_{\mathrm{D/A}}$ with ternary elements $\{0, \pm 1\}$ (a total of $3^{L_{\mathrm{D/A}}}$ possibilities). After the D/A conversion, the continuous-time signal $y(t)$ is eventually up-converted to the RF carrier and sent to the antenna.

4.1.3 Resampling

The baseband processing algorithms such as synchronization, channel estimation and data detection assume that the signal is sampled with an integer number, N_c, of samples per chip. In the current implementation we set $N_c = 4$. This solution avoids utilization of resampling techniques at both the transmitter and receiver front-end. These techniques have been studied in [14] and will be implemented in the next version of the software.

4.2 Receiver

In this section we analyze some of the theoretical aspects of the receiver signal-processing. In particular we give a description of the receiver front-end architecture. Then we focus on channel estimation, matched filter synthesis and symbol detection.

4.2.1 IF sampling and down conversion

Once the RF signal incoming from the antenna has been down-converted to IF, it is sampled by an A/D converter at a certain rate f_s (figure 5). Calling $r_{\mathrm{IF}}(t)$ the received IF analog signal and choosing $f_s \geq 2W$ according to (3), because of the periodicity of

the discrete-time signal spectrum, the resulting real sampled signal $r[n] = r_{\mathrm{IF}}(n/f_s)$ is pass-band with a spectrum replica centered at $f_s/4$ (although f_{IF} and f_s at the receiver can be different from f_{IF} and f_s at the transmitter, for simplicity we use the same notation).

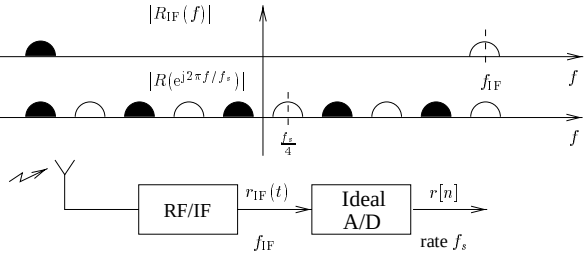


Figure 5: Receiver front-end

Notice that here in order to avoid signal resampling we suppose the rate f_s to be a multiple integer of the chip rate (i.e. $f_s = N_c f_c$ where in our implementation we set $N_c = 4$).

Although a base-band version of the received signal can be obtained by multiplying $r[n]$ by $(-\mathrm{j})^n$ and then by low pass filtering, we will show further how the channel estimation and the data detection processes can be performed in pass-band.

4.2.2 Channel estimation

Here we consider the training-sequence based multiuser channel estimation procedure for block-synchronous CDMA described in the UMTS/TDD norm.

In this scheme users are roughly synchronized to a common time-reference and transmit the training sequence at the same time (user timing errors are included as effect of the channel and taken automatically into account by the estimation procedure). The maximum channel length (including possible timing errors) is Q symbols and the training sequence sent by each u-th user is a cyclic shift of the same common base training sequence $\mathbf{a} = [a[0], a[1], \ldots, a[M-1]]^T$ of length M symbols. This solution allows joint estimation of all user channels if $M \geq QU$, where U is the number of interfering users. It is proposed and described in [15], [16] and with some modifications in [17]. The interested reader is referred to these papers and references therein for more details.

Under these assumption we can write the received signal sampled at frequency $f_s = N_c f_c$ during the M symbols spanned by the training sequence as

$$\boldsymbol{w} \quad = \quad \bar{\mathbf{A}}\,\boldsymbol{g} + \boldsymbol{\nu} \tag{8}$$

where

$$\boldsymbol{w} = [w[0], w[1], \ldots, w[M\,N_c - 1]]^T \tag{9}$$

is the received signal,

$$\boldsymbol{g} = [\boldsymbol{g}_1^T, \ldots, \boldsymbol{g}_u^T, \ldots, \boldsymbol{g}_U^T]^T \tag{10}$$

is a vector containing the channel impulse responses of the U users,

$$\boldsymbol{g}_u = [g_u[0], g_u[1], \ldots, g_u[QN_c - 1]]^T \tag{11}$$

is the u-th user channel filter vector and $\boldsymbol{\nu}$ is a vector of interference plus noise samples, assumed white. The $MN_c \times MN_c$ matrix $\bar{\mathbf{A}}$ is defined as

$$\bar{\mathbf{A}} = \mathbf{A} \otimes \mathbf{I}_{N_c} \tag{12}$$

where $(\otimes)$ denotes the Kronecker product and $\mathbf{A}$ is a circulant matrix containing all the possible cyclic shifts (by columns) of the base sequence $\mathbf{a}$. The matrix $\bar{\mathbf{A}}$ is also circulant and it is unitary similar [18] to the diagonal matrix diag$(\bar{\boldsymbol{\alpha}})$, where

$$\bar{\boldsymbol{\alpha}} = [\underbrace{\boldsymbol{\alpha}^T, \ldots, \boldsymbol{\alpha}^T}_{N_c \text{ times}}]^T$$

and where $\boldsymbol{\alpha}$ is the discrete Fourier transform of $\mathbf{a}$. After some algebra, it is possible to show that the Least Squares estimation of the overall channel impulse response $\boldsymbol{g}$ is given by

$$\hat{\boldsymbol{g}} \quad = \quad \text{IDFT}\left\{ \frac{\text{DFT}\{\boldsymbol{w}\}}{\bar{\boldsymbol{\alpha}}} \right\} \tag{13}$$

where DFT and IDFT denote direct and inverse discrete Fourier transforms and the ratio of two vectors should be interpreted as the element-by-element division.

This approach can be applied to both base-band and pass-band signals. In our case since the received signal is real and pass-band, its spectrum shows both the left and right-side replicas. As it is shown in figure 6 where we set $N_c = 4$, the left-side replica occupies the first and the second frequency intervals while the right one occupies the third and the fourth intervals.

The receiver can use the a priori information that the signal bandwidth is limited to W. Then it can limit the computation of the products to the range $[f_s/4 - W/2, f_s/4 + W/2]$, setting to 0 the rest of the channel estimates spectrum. Notice that this operation in the frequency domain corresponds to a low-pass filtering in the time-domain, moreover it reduces the computational cost since only a part of the MN_c products (by the elementwise inverses of $\bar{\boldsymbol{\alpha}}$ in (13)) are computed. Eventually, after the IDFT, the processing gives the estimated channel complex envelope.

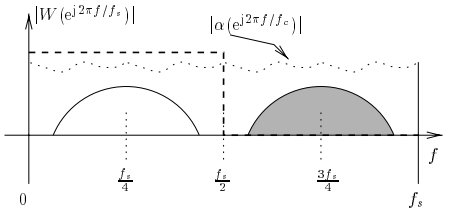


Figure 6: Channel estimation: pass-band

4.2.3 Matched Filter synthesis and data detection

Given the channel estimates, we are then interested in synthesizing a Matched Filter (MF) matched to the cascade formed by the user data spreading sequence, the chip pulse shape filter and the user channel. In other words if the overall impulse response is given by

$$f(t) = \sum_{i=0}^{N-1} s_i \psi(t - i/f_c) * c(t) = \sum_{i=0}^{N-1} s_i g(t - i/f_c) \tag{14}$$

where $\boldsymbol{s} = [s_0, \ldots, s_{N-1}]$ is the spreading sequence, $c(t)$ is the channel response and N is the spreading gain, we synthesize the discrete-time filter $f[k] = f(t)|_{t=t_k}$ matched to the overall response. To this end we compute

$$\hat{f}[k] = s[k] * \hat{g}[k] \tag{15}$$

Eventually in order to extract the data symbols we filter the received signal with the MF obtaining

$$v[k] = r[k] * \hat{f}^*[-k] \tag{16}$$

Notice that the signal after matched filtering is still pass-band and the symbol estimates after sub-sampling are given by

$$\hat{b}[k] = v'[NN_c k] \tag{17}$$

where $v'[k]$ is the complex envelope of the MF output. But since

$$v'[k] = (-\mathrm{j})^k v[k] \tag{18}$$

substituting into (17) and for $N_c = 4$ we get

$$\hat{b}[k] = (-\mathrm{j})^{NN_c k} v[NN_c k] = v[4Nk] \tag{19}$$

In this way the symbol estimates are given by sub-sampling the MF output at symbol rate without taking care of the demodulation.

4.2.4 Carrier synchronization and decoding

The carrier synchronization is done at symbol rate with a classical decision directed algorithm [11] The algorithm then takes a decision on the symbols and recovers the data flow (in our example a video stream).

5 Validation of the existing platform

The platform described in this paper has been validated by the transmission and the reception of of two user's real-time flows in an indoor environment. Two H263 video streams are transmitted in parallel and decoded in real time. For this we use the following parameters:

- spreading factor 16
- bit rate 384 kbps
- TDD mode: 1 Tx slot is followed by 1 Rx slot (trasmission is done every 2 slots).
- two synchronous user per slot
- RF band: 5 MHz at 2.1 GHz

6 Conclusion

This first demonstration shows that the architecture of the platform is promising. In the next future the platform will be enhanced and opened to industrial and academic collaboration. The enhancements will consist of

- multiple antenna implementation
- more sophisticated signal processing algorithms
- multi user detection

The collaboration will concern

- radio subsystem improvement (flexibility and sensitivity)
- high level protocol stacks

7 Acknowledgments and credits

The platform presented in this paper is the result of a joint collaboration with EPFL (Federal Swiss Institute of Lausanne). The authors want to thank the EPFL team, Bixio Rimoldi, Raymond Knopp and Thomas Hoehne for their valuable contributions. The acquisition and conversion cards (hardware and software) were developed by EPFL. The RF part of the platform is the result of a joint collaboration between STMicroelectronics, EPFL and Eurécom. STMicroelectronics developed the RF and CFG cards.

References

[1] S. Srikanteswara, J. H. Reed, P. Athanas, and R. Boyle, "A soft radio architecture for reconfigurable platforms," *IEEE Communications Magazine*, February 2000.

[2] "Special issue on software radio," *IEEE JSAC*, vol. 4, April 1999.

[3] "Software radio," *IEEE Personal Communications*, vol. 4, August 1999.

[4] H. Tsurumi and Y. Suzuki, "Broadband RF stage architecture for software-defined radio in handheld terminal applications," *IEEE Communications Magazine*, February 1999.

[5] K. Zangi and R. Koilpillai, "Software radio issues in cellular base stations," *IEEE JSAC*, vol. 17, pp. 561–573, April 1999.

[6] T. Rappaport, *Wireless Communications*. Englewood Cliffs, 1996.

[7] 3GPP-TSG-RAN-WG1, "TS-25.2xx series," tech. rep., January 2000.

[8] E. Dahlman, B. Gudmundson, M. Nilsson, J. Sköld, and E. R. S. AB, "UMTS/IMT-2000 based wideband CDMA," *IEEE Communications Magazine*, vol. 36, pp. 70–80, September 1998.

[9] "http://www.itu.int/imt/2-rad-devt/index.html."

[10] A. Furuskär, S. Mazur, F. Müller, H. Olofsson, and E. R. Systems, "EDGE: Enhanced data rates for GSM and TDMA/136 evolution," *IEEE Personal Communications*, vol. 6, pp. 56–66, June 1999.

[11] J. G. Proakis, *Digital Communications*. NY: McGraw Hill, 2nd ed., 1989.

[12] J. Mitola, "The software radio architecture," *IEEE Communications Magazine*, pp. 26–38, May 1995.

[13] J. Razavilar, F. Rashid-Farrokhi, and K. J. R. Liu, "Software radio architecture with smart antennas: A tutorial on algorithms and complexity," *IEEE JSAC*, vol. 17, pp. 662–676, April 1999.

[14] G. Montalbano and A. Nordio, "Front-end software-radio review," tech. rep., Institut Eurécom, July 1999. to be appeared.

[15] B. Steiner and P. Jung, "Optimum and suboptimum channel estimation for the uplink of cdma mobile radio systems with joint detection," *European Transaction on Communications*, vol. 5, pp. 39–49, Jan.-Feb. 1994.

[16] 3GPP-TSG-RAN-WG1, "TS-25.221v3.1.0; physical channels and mapping of transport channels onto physical channels (TDD)," tech. rep., January 2000.

[17] G. Caire and U. Mitra, "Structured multiuser channel estimation for block-synchronous DS/CDMA." Submitted to IEEE Transaction on Communications., July 1999.

[18] G. H. Golub and C. F. V. Loan, *Matrix Computation*. The John Hopkins University Press, 1996.