TRANSMISSION AND RECEPTION FRONT-END ARCHITECTURES FOR SOFTWARE RADIO

Giuseppe Caire, Pierre A. Humblet, Giuseppe Montalbano, and Alessandro Nordio

Institut Eurécom, B.P. 193, 06904 Sophia–Antipolis CEDEX, France
E-mail: `firstname.name@eurecom.fr`

***Abstract*– We present low-complexity algorithms for transmitter and receiver front-end suited to the implementation of Software Radio (SR) terminals. The proposed algorithms make the processing sampling frequency independent of the symbol rate of the digitally modulated signal and use the "IF-sampling" technique for D/A and A/D conversion. As a case-study, we consider training-based joint multiuser channel estimation and we show that our front-end algorithms work nicely when coupled with an efficient FFT-based joint channel estimator. A real-time PC based SR testbed with fully programmable Digital Signal Processors (DSP) has been successfully implemented based upon the concepts illustrated in this paper.**

1. INTRODUCTION

Software-defined Radio (SR) terminals able to reconfigure themselves to handle several different standards represent a really attractive solution to provide universal connection to the users of wireless communication systems. At the physical layer, SR requires signal processing algorithms suited to implementation on a programmable CPU, as opposed to analog or digital dedicated hardware. The goal is to perform operations like *channel selection* [1, 2], *up- and down-conversion* [3], synchronization and detection in the all-digital domain, by using high performance DSPs. Both transmit (Tx) and receive (Rx) front-end algorithms should be independent of system-dependent parameters like the signal bandwidth and the symbol (or chip, in a CDMA system) rate. Here we propose some front-end algorithms for the class of linearly-modulated digital signals. We assume that at both the Tx and Rx there is an RF/IF conversion stage. Then, we are concerned on the Tx side with the efficient generation of an IF analog signal from the baseband digital signal, while on the Rx side with the efficient generation of a digital signal from the IF analog signal. Different options are compared in terms of their complexity and performance. Complexity is measured in real operations per sample. Performance is given in terms of the signal-to-interference plus noise ratio (SINR) at the output of the receiving filter matched to the modulation elementary pulse, assuming that the transmission channel is pure additive white (not necessarily Gaussian) noise. The SINR is expressed as a function of E_s/I_0, where E_s is the average symbol energy and I_0 denotes the (frequency-flat) noise power spectral density, in the complex baseband equivalent model. The proposed algorithms performances are also validated on the training-based multiuser joint channel estimation scheme of UMTS-TDD (see e.g. [4], [5], and [6]) for a particular multipath test channel in terms of the actual measured matched-filter bound (MFB) at the output of the estimated channel matched filter versus the MFB of an ideal system. The concepts exposed in this paper have been implemented and tested on a real-time SR testbed developed at Eurécom [7], where the UMTS-TDD standard proposal has been experimented in practical field trials.

2. TRANSMITTER FRONT-END

2.1. Linearly Modulated Signals

The continuous-time complex envelope of a linearly modulated signal is given by

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

where $a[k]$ is a sequence of modulation symbols belonging to some complex alphabet (e.g., PSK, QAM [8]), $\psi(t)$ is the symbol-shaping pulse, bandlimited over $[-W/2, W/2]$, and T the symbol interval. In several cases of interest (e.g. TDMA, DS-CDMA, multi-user and multi-antenna systems) (1) can be used to model the transmitted signals. In a digital Tx, 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 I and Q baseband components are generated by low-pass filtering the output of two separate D/A converters, and the IF signal $y(t) = \mathrm{Re}\{x(t)e^{j2\pi f_{\mathrm{IF}} t}\}$ is produced by mixing the I and Q components of $x(t)$ with IF carrier signals in phase and quadrature and by summing the modulated real signals [8]. This approach requires two D/A converters, two low-pass filters, two analog mixers and one adder. Another approach [9], [3], consists of producing a sampled version of the IF modulated signal $y(t)$ by using a sampling rate $f_s > 2f_{\mathrm{IF}}$. The continuous-time signal is obtained by bandpass filtering the output of a single D/A converter. Since the IF typically ranges between 20 and 100 MHz this approach is extremely computationally intensive and unpractical. In the following, we propose a Tx front-end algorithm allowing a 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}.

2.2. IF-Sampling and Up-Conversion

We address a very simple method to obtain an IF signal from a discrete-time baseband signal. For this purpose we choose the sampling rate f_s according to the expression

$$f_s = \frac{f_{\mathrm{IF}}}{\ell \pm 1/4} \qquad \text{for a positive integer } \ell \quad (2)$$

Then, we can generate the discrete-time real signal

$$x'[n] = \mathrm{Re}\left\{x[n]e^{j2\pi(f_{\mathrm{IF}}/f_s)n}\right\} = \mathrm{Re}\left\{j^{\pm n}x[n]\right\} \quad (3)$$

whose periodic spectrum has a replica centered at f_{IF}. After ideal D/A conversion, a pass-band filter centered at f_{IF} removes the other replicas, generating the desired IF modulated signal. The operation $\mathrm{Re}\left\{j^{\pm n}x[n]\right\}$ in (3) is simply obtained by alternately changing the sign of the real and imaginary part of $x[n]$. In order to avoid aliasing the sampling rate must satisfy the condition $f_s \geq 2W$.

The authors appear in alphabetical order.

Eurécom's research is partially supported by its industrial partners: Ascom, Cégétel, France Télécom, Hitachi, IBM France, Motorola, Swisscom, Texas Instruments, and Thomson CSF

2.3. D/A Conversion

In the above description we assumed an ideal D/A converter with flat frequency response. Actual D/A converters exhibit a low-pass frequency response (approximately) of the form $\mathrm{sinc}(f/f_s)$ that does not extend to IF. A way to extend the D/A converter response so as to reduce the attenuation at IF consists of clocking the converter at rate $f_d = L_{\mathrm{D/A}} f_s$, where $L_{\mathrm{D/A}}$ is a positive integer such that $f_d \gg f_{\mathrm{IF}}$, and up-sampling $x'[n]$ by the factor $L_{\mathrm{D/A}}$. Unfortunately, this approach reduces the average signal energy per-sample by a factor $L_{\mathrm{D/A}}$. Thus, high amplification gains may become necessary in the IF/RF stage, giving rise to significant non-linear distortion. To overcome this problem one may pre-compensate the linear response of the D/A converter by introducing a pass-band FIR filter between the up-sampler and the D/A converter (see [10]). The filter must be designed in order to enhance the spectrum replica at IF while attenuating the other replicas.

2.4. Transmit Signal Resampling

In the previous approach the sampling frequency f_s depends on f_{IF} and on the signal bandwidth W. In a SR system, we can think of W as the maximum signal bandwidth over all handled signal formats, while f_{IF} is fixed and depends on the analog hardware. In any case, f_s should be independent of the symbol rate $R_s = 1/T$ of the particular signal format. When $f_s \neq R_s$, the Tx must produce the signal $x[n]$ at rate f_s from the discrete-time symbol sequence $a[n]$ at rate R_s. We refer to this general problem as *resampling*. Resampling by rational factors is widely discussed in the literature (see e.g. [11, 12]). A general sampling rate conversion scheme by a rational factor L_u/L_d consists of an up-sampler by a factor L_u, a low-pass filter and a down-sampler by a factor L_d. The low-pass filter can be efficiently implemented by a polyphase filter bank [11] composed by L_u phases, where each filter phase is sampled at rate R_s. Unfortunately, in general the ratio f_s/R_s is not a rational number, therefore f_s can be only approximated by $\frac{L_u}{L_d} R_s$ by a careful choice of the integers L_u and L_d. Since the shaping filter $\psi(t)$ of the modulation scheme is bandlimited, it can be used as low-pass filter in the resampler. In this way, shaping and resampling are integrated in a single step with considerable saving in computational complexity. Assume that $\psi(t)$ is approximated by an FIR spanning N_ψ symbol intervals (e.g., by windowing [12]) and let $\psi_\ell[i] = \psi((iL_u + \ell)/(L_u R_s))$ with $\ell = 0, \ldots, L_u - 1$ and $i = 0, \ldots, N_\psi - 1$, denote the i-th sample of the ℓ-th phase of the polyphase filter bank implementation of the shaping discrete-time filter. Then, the output at rate $L_u R_s$ (i.e. oversampled by a factor L_u w.r.t. the symbol rate) of the polyphase pulse-shaping filter bank is given by $v[k] = \sum_{i=0}^{N_\psi - 1} \psi_\ell[i] a[m-i]$ where $m = \lfloor k/L_u \rfloor$ and $\ell = k$ modulo L_u. A low cost approach for achieving the desired non-rational sampling rate conversion consists of using a suboptimal interpolation method (e.g., nearest neighbor, linear, cubic or cubic spline interpolation) at the output of the polyphase filter bank. The computation of the output sample value depends on the interpolation method. Nearest neighbor (NN) interpolation is the simplest technique, since it approximates the desired output sample to the time-nearest input sample. In fact using NN interpolation the overall cost of the resampler reduces to the storage in memory of the coefficients of the shaping filter bank. The NN interpolator produces the output sequence $x[n]$ at sampling rate f_s as $x[n] = v[k]$ where

$$|t_n R_s - k/L_u| \leq 0.5 \qquad (4)$$

and where $t_n = t_0 + n/f_s$ is the desired sampling epoch (t_0 is a fixed time-offset). The computational complexity is then the cost of an inner product per output sample. Normally the shaping filter coefficients are real while, in general, the symbols $a[k]$ are complex. Then, the resulting computational complexity is $2N_\psi$ multiplications and $2(N_\psi - 1)$ sums per output complex sample. However, as previously indicated, only the real signal $x'[n]$ is needed in our Tx front-end. This is equivalent to computing alternately the I and Q components of $x[n]$. Therefore, the actual computational complexity per output sample reduces to N_ψ multiplications and $N_\psi - 1$ sums. The polyphase filter-bank coefficients are pre-computed. The memory occupation of the filter is proportional to the number of filter phases L_u. It is clear from (4) that the accuracy of NN interpolation critically depends on L_u. However, since memory is a really cost-effective resource, it is advantageous to choose L_u large enough in the Tx and insist on NN interpolation rather than considering a lower L_u and implementing a more complex interpolation method (e.g., linear, cubic, etc.).

3. RECEIVER FRONT-END

3.1. IF-Sampling and Down-Conversion

The IF received analog signal $r_{\mathrm{IF}}(t)$ is sampled by an A/D converter at rate f_s. If $f_s \geq 2W$ is chosen according to (2), 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 Rx can be different from f_{IF} and f_s at the Tx, for simplicity we use the same notation).

3.2. Receive Signal Resampling

To simplify synchronization and data detection, the received signal should be re-sampled at rate R'_s, integer multiple of the symbol rate (or chip rate, in the case of CDMA). As in the Tx, the system should be able to choose R'_s independently of f_s (within a certain range of supported signal formats), where R'_s/f_s is in general non-rational. In order to have a perfect transparent resampling system a general approach consists of resampling the I and Q components of the real passband signal $r[n]$ at rate R'_s and then remodulate the signal at frequency $R'_s/4$ by simple sign change. In this way a real passband signal sampled at the desired rate R'_s is obtained. In the following we consider two resampling approaches.

Approach 1. This is the most classical resampling scheme

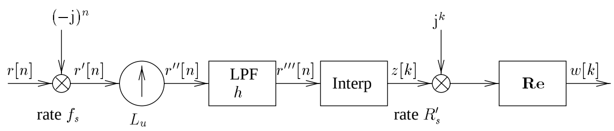

Figure 1: Resampling approach 1.

and it is basically the same proposed for the Tx (see section 2.4). Notation refers to the scheme of figure 1. The low-pass interpolation filter h is implemented as a polyphase FIR filter bank with L_u phases ($h_\ell[i]$ denotes the ℓ-th phase i-th coefficient, for $\ell = 0, \ldots, L_u - 1$ and $i = 0, \ldots, N_h - 1$). Let $r'''[n]$ denote the output of the polyphase filter bank, given by $r'''[n] = \sum_{i=0}^{N_h - 1} h_\ell[i](-\mathrm{j})^{m-i} r[m-i]$ where $m = \lfloor n/L_u \rfloor$ and $\ell = n$ modulo L_u. In case a NN interpolator is employed, as in the Tx, the output sequence $z[k]$ at sampling rate R'_s is generated as $z[k] = r'''[n]$ where

$$|t_k f_s - n/L_u| \leq 0.5 \qquad (5)$$

and where $t_k = t_0 + k/R'_s$ is the desired sampling epoch (t_0 is a fixed time-offset). Finally, the resampled passband signal modulated at frequency $R'_s/4$ is given by $w[k] = \mathrm{Re}\{r'''[n]\mathrm{j}^k\}$. The interpolation filter can be chosen to be real in order to have a complex symmetric frequency response. In this case $w[k]$ can be written as

$$w[k] = \begin{cases} +\sum_{i=0}^{N_h-1} h_\ell[i]\mathrm{Re}\{(-\mathrm{j})^{m-i}r[m-i]\} & k=4l \\ -\sum_{i=0}^{N_h-1} h_\ell[i]\mathrm{Im}\{(-\mathrm{j})^{m-i}r[m-i]\} & k=4l+1 \\ -\sum_{i=0}^{N_h-1} h_\ell[i]\mathrm{Re}\{(-\mathrm{j})^{m-i}r[m-i]\} & k=4l+2 \\ +\sum_{i=0}^{N_h-1} h_\ell[i]\mathrm{Im}\{(-\mathrm{j})^{m-i}r[m-i]\} & k=4l+3 \end{cases}$$

for integer l, where m and ℓ depend on n, which is a function of k through the previous resampling epoch relation (5). Then the computation of $w[k]$ reduces to computing alternately the values $w_0[k] = \sum_{i=0}^{N_h/2-1} \tilde{h}_{\ell,0}[i]r[m-2i]$ and $w_1[k] = \sum_{i=0}^{N_h/2-1} \tilde{h}_{\ell,1}[i]r[m-2i-1]$ where the modified (real) filter coefficients $\tilde{h}_{\ell,0}[i] = (-1)^i h_\ell[2i]$ and $\tilde{h}_{\ell,1}[i] = (-1)^i h_\ell[2i+1]$ can be precomputed.[1] The overall complexity of this resampling approach amounts to an inner product of length $N_h/2$ (i.e., $N_h/2$ multiplications and $N_h/2-1$ sums) per output sample at rate R'_s. One may notice that we could use as interpolation filter the same shaping filter ψ used in the Tx. In this way, resampling and pulse shaping matched filtering could be implemented jointly. However, while this approach gives a complexity saving in the Tx, it is not suited to the Rx because better low-pass FIR responses h, with the same number of taps, can be sythetized by using appropriate filter design methods (see e.g. [11]).

Approach 2. An alternative approach consists of exploiting the fact that in our system the baseband signal $r'[n]$ is already oversampled w.r.t. the Nyquist rate. Thus the interpolation filter h can be avoided and more computationally intensive interpolation methods can be implemented without significant increase of complexity. Figure 2 shows the block diagram of this resampling system. Since the baseband signal $r'[n]$ is alternately purely real or imaginary, the real and imaginary parts can be interpolated (without any prior filtering) separately at half the input rate and eventually recombined. Instead of using a LPF h as done in the previous approach, here the high frequency components of the discrete-time signal are removed by decimating by a factor 2 its real and imaginary part. Then the complex signal $z[k]$ is remodulated and the real part is taken, to obtain $w[k]$. We shall remark that the whole processing of remodulating $z[k]$ at frequency $R'_s/4$ and taking the real part, reduces to computing alternately the real and the imaginary part of $z[k]$ and alternately changing the sign to obtain $w[k]$. Therefore, the overall complexity amounts to one interpolated real value per output sample at rate R'_s. The computational cost of this approach depends on the interpolation method, namely none for NN interpolation, 2 sums and 1 multiplications for linear interpolation, 11 sums and 8 multiplications for cubic interpolation, and 9 sums and 11 multiplications for cubic spline interpolation per output sample.

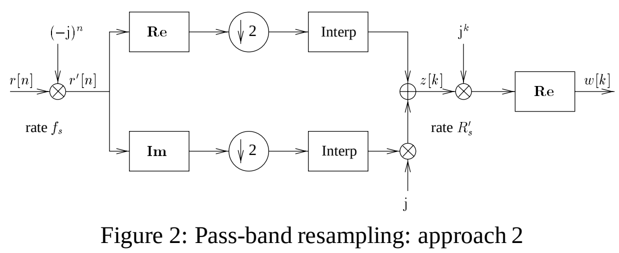

Figure 2: Pass-band resampling: approach 2

4. FRONT-END PERFORMANCES

In this section we illustrate the impact of the various system parameters on the performance of the proposed front-end algorithms. As performance measure we use the SINR measured at the output of Rx front-end vs. the nominal E_s/I_0 ratio. The SINR loss w.r.t. the nominal E_s/I_0 is due to two effects: intersymbol interference (ISI) and energy decrease of the useful samples. The useful samples energy decrease is only due to the distortion introduced by resampling while the ISI is also due to the truncation of shaping filter. We assume a root-raised cosine shaping filter with roll-off factor $\alpha = 0.22$, approximated by an FIR filter spanning 10 symbol intervals, an IF carrier of 70 MHz and a symbol rate of $R_s = 3.84$ Mbaud (or Mchip/s, in the case of CDMA). The Tx and Rx sampling frequency is $f_s = f_{\mathrm{IF}}/(5+1/4) = 13.33$ MHz, yielding $f_s/R_s \approx 3.4722$. Figure 3 shows the performance of the Rx front-end with resampling approach 2 using NN, linear, cubic and cubic spline interpolation at the Rx. Each sub-figure shows the performance with Tx up-sampling factor $L_u = 4, 8, 16, 32, 64$. For large L_u cubic spline interpolation at the Rx yields negligible SINR degradation over the range 0–30 dB of nominal E_s/I_0. Figure 4 shows the performance of the Rx front-end with resampling approach 1 where NN interpolation is employed without oversampling of the signal $r'[k]$ before filtering. The low-pass filter is a linear phase FIR filter designed by using the Remez algorithm [12] in order to minimize the distortion in the signal bandwidth. The filter length is $N_h = 6, 10$ sample periods. With NN interpolation the absence of up-sampling of the received signal yields severe degradation of the output SINR with both approaches. Moreover, in the case of approach 1, larger filter lengths do not yield any significant improvement. Note also that the receiver matched filter only account for ψ and not for the whole cascade $\psi \star h$, although this does not yield any significant signal distortion.

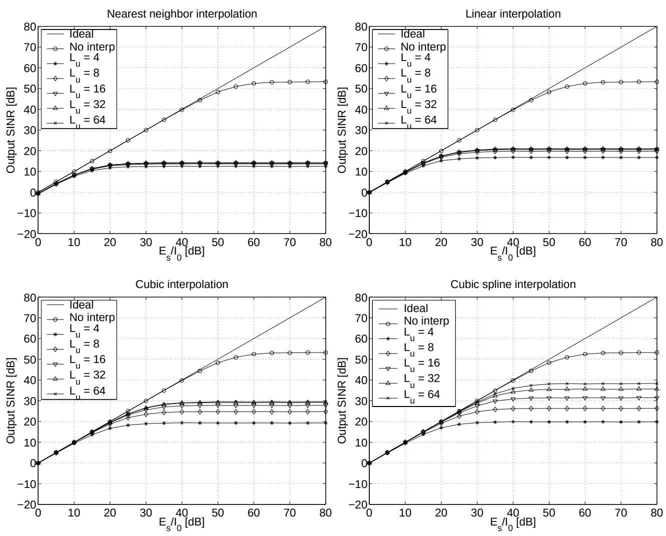

Figure 3: Rx resampling approach 2

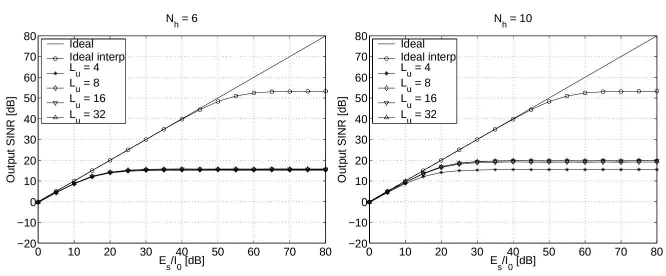

Figure 4: Rx resampling approach 1, $N_h = 6$ and $N_h = 10$

5. EXAMPLE: TRAINING-SEQUENCE BASED CHANNEL ESTIMATION

The Rx front-end output signal, sampled at rate multiple integer of the symbol rate is suitable to be further processed by the Rx. Here the proposed front-end algorithms are tested on a

[1] The actual value of $w[k]$ is given either by $\pm w_0[k]$ or by $\pm w_1[k]$, where the sign depends on the parity of k and m.

training-sequence based multiuser channel estimation procedure for block-synchronous CDMA (e.g., UMTS in TDD mode). In this scheme users are quasi-synchronous and transmit their training sequence at the same time (small timing errors are accounted for by the channel estimation procedure). The maximum channel length (including possible timing errors) is Q chips and the training sequence sent by each user is built from the same common base training sequence of length M chips with a cyclic extension of Q chips. This solution allows a joint estimation of all user channels if $M \geq QU$, where U is the number of interfering users. Due to the particular structure of the training sequence one DFT and one inverse DFT (IDFT) are sufficient to produce a least square channel estimate for all the users [10]. The method applies to passband signals as well as to baseband signals, without complexity increase. Furthermore, passband signals do not need explicit demodulation. Demodulation can be automatically achieved by down-sampling the output of the pass-band symbol matched filter at symbol rate. Figure 5 shows qualitatively the spectrum $W(\mathrm{e}^{\mathrm{j}2\pi f/R'_s})$ of the input sequence $w[k]$, and the spectrum $\alpha(\mathrm{e}^{\mathrm{j}2\pi f/R_s})$ of the training sequence repeated four times due to the oversampling factor $N_c = 4$, with passband signals. The transmitted signal is essentially bandlimited and centered around $R'_s/4$. Thus the passband one-sided (i.e., complex) channel impulse response of all users can be estimated by setting to zero all coefficients of DFT$\{w[k]\}$ but those in the frequency range $[R'_s/4 - W/2, R'_s/4 + W/2]$.

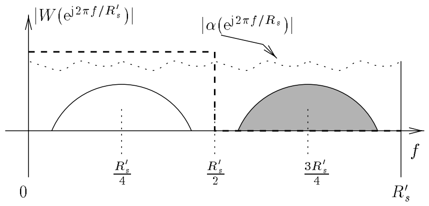

Figure 5: Estimation of the complex passband channel response.

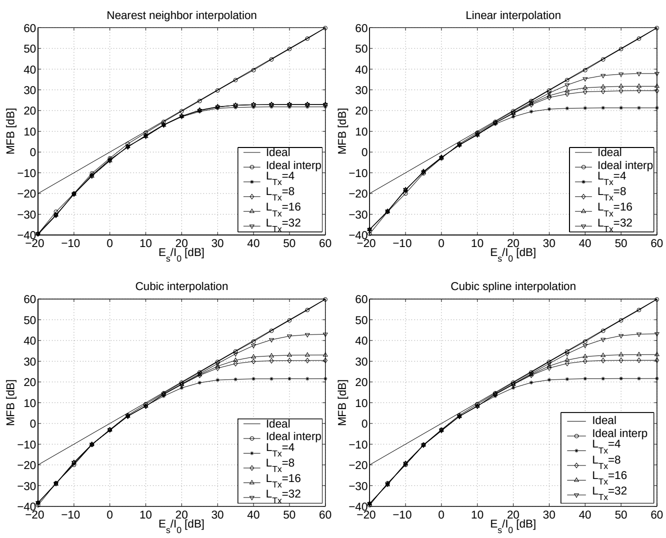

Figure 6: Channel estimation with multipath channel

We used the training sequence defined in the 3GPP proposal for UMTS-TDD mode (burst type 2). The sequence length is $M = 192$ chips and the assumed channel length is $Q = 64$ chips, allowing the simultaneous estimation of $U = M/Q = 3$ users. We considered a channel with delay-intensity profile given by the delays $\tau = (0, 0.9, 2.8, 4.7)$ chip intervals and by the relative path strengths $\sigma^2 = (0, -2, -7, -8.5)$ dB (the sum of the path gains is normalized to 1). Since the SINR would be dominated by the ISI introduced by the channel we use the matched-filter bound (MFB) [8] as performance measure instead. Figure 6 shows the MFB resulting from the estimated matched filter vs. E_s/I_0 (i.e. the MFB of an ideally matched filter). The curves labeled as "Ideal interp." refer to the performance when ideal interpolation is used at both Tx and Rx. Here, the MFB degradation is dominated by the channel estimation errors at low E_s/I_0 and by the resampling distortion at high E_s/I_0.

6. CONCLUSIONS

We presented simple and effective algorithms suited for a flexible implementation of Tx and Rx front-ends based on programmable DSPs, for SR applications. We compared various solutions in terms of complexity and performance. The resulting architectures for Tx and Rx are both based on the concept of IF sampling and resampling. At the Tx, we can generate directly the IF analog signal via the D/A converter, without using mixers, filters and I-Q adders. At the Rx, the IF analog signal is downsampled at a much lower rate than the Nyquist rate in order to generate a passband digital signal with low carrier frequency, suited to further processing. Resampling at both Tx and Rx is needed to support several different symbol (or chip) rates. At the Tx resampling can be integrated with pulse-shaping and can be efficiently implemented by a polyphase filter-bank followed by nearest-neighbor interpolation. At the Rx more accurate interpolation techniques are needed while the pulse-shaping matched filter can be more efficiently integrated with the channel matched filter, provided by a channel estimator. Finally, we tested our front-end algorithms with an actual channel estimation technique. Basing upon these concepts a real-time PC based SR testbed has been developed at Eurécom [7] and the UMTS-TDD standard proposal has been experimented in practical field trials.

7. REFERENCES

[1] H. Tsurumi and Y. Suzuki, "Broadband RF stage architecture for software-defined radio in handheld terminal applications," *IEEE Comm. Mag.*, Febr. 1999.

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

[3] 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.

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

[5] 3GPP-TSG-RAN-WG1, "TS-25.22x series v3.1.0 (TDD)," tech. rep., Jan. 2000.

[6] G. Caire and U. Mitra, "Structured multiuser channel estimation for block-synchronous DS/CDMA," submitted to IEEE Trans. on Comm., July 1999.

[7] C. Bonnet, G. Caire, A. Enout, P. A. Humblet, G. Montalbano, A. Nordio, D. Nussbaum, "A software radio platform for new generations of wireless communication systems," submitted to CNIT'00 12th Tyrrhenian Int. Workshop on Digital Comm., Sept. 2000.

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

[9] J. Mitola, "The software radio architecture," *IEEE Comm. Mag.*, pp. 26–38, May 1995.

[10] G. Caire, P. A. Humblet, G. Montalbano, and A. Nordio, "Transmission and reception front-end algorithms for software radio," submitted to IEEE JSAC Wireless Comm. Series, July 2000.

[11] P. Pirsch, *Architectures for Digital Dignal Processing.* NY: John Wiley & Sons, 1998.

[12] A. Oppenheim and R. Schafer, *Discrete-time signal processing*. Prentice-Hall, 1989.