Abstract

When expanding digital signal processing of mobile communications terminals toward the antenna while making the terminal more wideband in order to be able to cope with different mobile communications standards in a software-radio-based terminal, the designer is faced with strong requirements such as bandwidth and dynamic range. Many publications claim that only reconfigurable hardware such as FPGAs can simultaneously cope with such diversity and requirements. Starting with considerations of the receiver architecture, we describe key functionalities of the digital front-end and highlight how signal characteristics of mobile communications signals and commonalities among different signal processing operations can be exploited to great advantage, eventually enabling implementations on an ASIC that, although not reconfigurable, would empower the software radio concept.

The Digital Front-End of Software Radio Terminals

TIM HENTSCHEL, MATTHIAS HENKER, AND GERHARD FETTWEIS
DRESDEN UNIVERSITY OF TECHNOLOGY

One of the fundamental ideas of software radio is the expansion of digital signal processing toward the antenna, and thus to regions where analog signal processing has been dominant so far. It is straightforward to realize that the hardware platform is a most prominent enabling component of a software radio terminal. Of special interest is the very part of the terminal where analog signal processing is replaced by digital signal processing, namely the *digital front-end*.

Having its naming derived from *radio frequency (RF) front-end* and *digital signal processing*, the digital front-end is a part of the receiver realizing front-end functionalities digitally that were formerly realized by means of analog signal processing (i.e., downconversion and channel filtering).

A main reason for replacing analog with digital signal processing is the possibility to "softly" reconfigure the system, thereby enabling the implementation of different air interfaces on a given hardware platform. Although baseband processing has been realized digitally for some time, parameters related to the digital front-end such as channel spacing/bandwidth and carrier frequency could not be changed by means of software. Therefore, such terminals, even though often built with high-performance signal processors, are sentenced to "hardwired-looking" terminals.

Apparently a standard design of mainly digital filters, the digital front-end turns out to be one of the most power- and time-critical functionalities of the software radio terminal. This is due to the combination of large bandwidth and high dynamic range of the signals to be processed. Consequently, the digital signals have large word lengths and high sample rates. Besides increasing power consumption, high sample rates make the use of time-shared hardware infeasible. Despite these facts, we show below that even under strong constraints software radio is feasible today, specifically the digital front-end as one of the most critical parts.

Generally, the receiver is regarded as the more demanding design problem. Therefore, and since all considerations are principally valid for both the transmitter and receiver of a software radio terminal, we shall stick to the receiver in the following.

Starting with a description of receiver architectures and important signal characteristics, the digital front-end of a software radio receiver, with its main functionalities, channelization and sample rate conversion, is the object of this article.

Receiver Architecture

The Ideal Software Radio Receiver

The architecture of the ultimate software radio receiver [1] with a minimum of analog components (analog front-end) is given in Fig. 1. Since all channelization tasks (discussed later) are performed digitally, the analog front-end and analog-to-digital converter (ADC) have to process the complete signal bandwidth for which the terminal is designed.

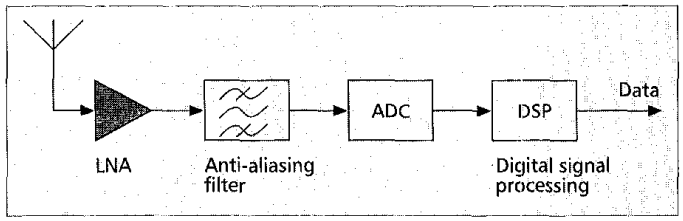
Before evaluating this architecture, the signals to be processed by such a terminal must be assessed. Typical characteristics of mobile communications signals — fading, shadowing, and so on (caused by RF signal propagation characteristics) — in conjunction with potentially strong blocking and interfering signals (due to the coexistence of several transmit signals) lead to a very high dynamic range. This high dynamic range of mobile communications signals is reflected in the receiver characteristics in the definition of the different standards (e.g., [2]). Several parts of the receiver have to process a large number of channels simultaneously if the channel selection functionality is shifted from the analog to the digital domain. Such "narrowband signals at wideband reception" yield a dynamic range far above that with which conventional terminals have to cope. For instance, a wideband signal in the the Global System for Mobile Communications (GSM) band has a dynamic range in the vicinity of ~100 dB. Extending to signals of different standards that might appear simultaneously at the receiver, the dynamic range can get even larger.

Supposing it is feasible to design the small amount of analog components that process such signals, all-digital signal processing makes no sense if the signals cannot be converted to the digital domain. Thus, the ADC is a key component of a software radio terminal [3]. As the analog front-end, it has to

1070-9916/99/$10.00 © 1999 IEEE

cope with signals of large bandwidth and high dynamic range. Basically, the dynamic range of ADCs can be increased by increasing the number of bits and/or the oversampling ratio, that is, the ratio of sample rate and channel bandwidth of the currently received signal spectrum. The dynamic range of memoryless ADCs rises with 3 dB per doubling the oversampling ratio and 6 dB per additional bit of quantizer resolution.

Obviously, oversampling does not yield a high gain in dynamic range. Therefore, memoryless ADCs require a large number of bits in order to meet the high dynamic range requirements of wideband mobile communications signals. Although there are ADCs available that sample at rates of nearly 100 million samples per second (MSPS) and quantize the signal with 14 bits, their dynamic range does not reach the desired level. Advanced ADCs applying noise-shaping techniques (e.g., $\Sigma\Delta$ converters) promise to provide extreme dynamic range at relatively low expense. However, their principle permits high dynamic range only in certain frequency bands. This property limits their applicability, at least as long as parameterizable versions are not available. Finally, it should be mentioned that besides the dynamic range, the sample rate having to fulfill the Nyquist criterion also limits the bandwidth that can be digitized in practice.



■ **Figure 1.** *The ideal software radio receiver.*

The Feasible Software Radio Receiver

The conclusion we can draw from the above discussion is that the ideal software radio architecture of Fig. 1, digitizing the bandwidth of all services to be supported by the terminal, is not feasible today. Therefore, the bandwidth the ADC has to digitize and the digital front-end has to process must be reduced. This idea leads to the concept of *partial band digitization* in contrast to *full band digitization*. The latter reflects the ideal software radio approach described above, while the first can be described as follows:

In order to cover all services to be supported by the software radio terminal, a limited band has to be selected out of the full band by means of analog conversion and intermediate frequency (IF) filtering.

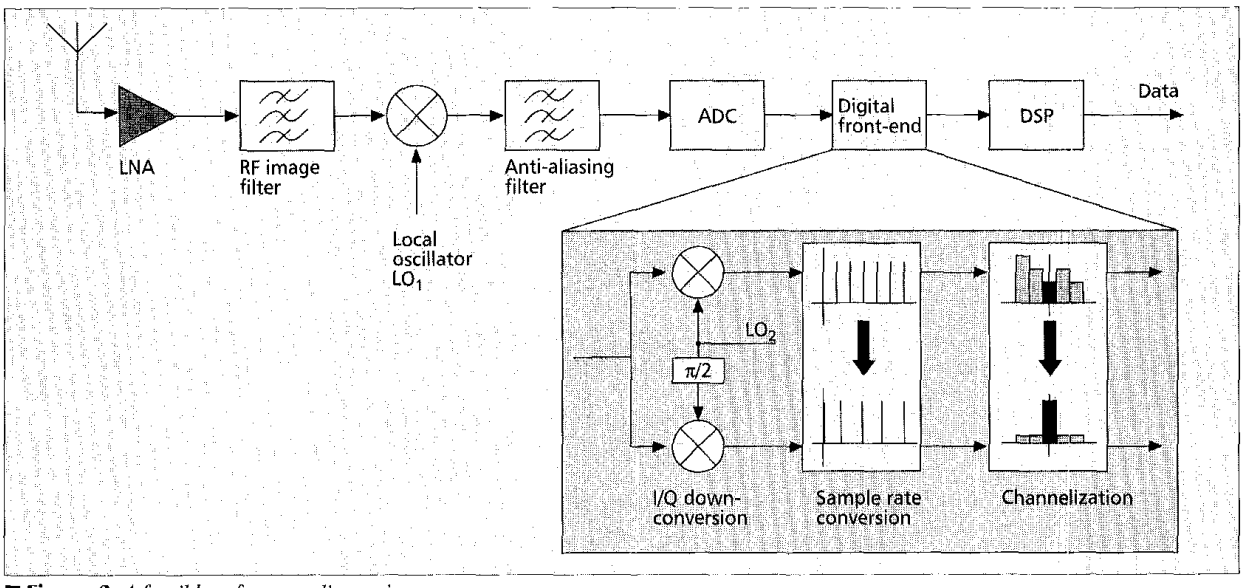
This concept results in an architecture employing IF sampling (Fig. 2).

We would like to point out that in the context of software radio, the notion of *direct conversion* has lost its meaning in some respects, since it is always related to a certain channel of interest to be selected. However, the analog front-end of a software radio receiver selects a bundle of channels. Thus, we cannot determine which channel is converted directly. On the other hand, it must be noted that the idea of direct conversion (i.e., conversion to a complex signal) has not lost importance. Hence, the designer has to decide between complex and real conversion. While the first is more expensive with respect to the mixer, it circumvents filters that would be necessary for the suppression of RF images in the second case. In this article real conversion is assumed.

The digital front-end performs channelization and sample rate conversion (SRC). Channelization comprises all tasks necessary to select the channel of interest. This includes conversion to baseband, channel filtering, and possibly despreading. SRC is a functionality that comes from the idea that it is surely sensible to sample the analog signal at a fixed rate. This simplifies clock generation for the ADC, which would otherwise have to be parameterizable. However, the signals generally have to be processed at symbol or chip rates dictated by the different standards. Both facts lead to the necessity to digitally convert the digitization rate to the rate of the current standard of operation.

The Bandwidth–Dynamic Range Trade-off

Selecting the digitization bandwidth so that channels of only one standard are received simultaneously offers an advantage, that can be derived from fundamental characteristics of mobile communications signals, and that could be named the *bandwidth–dynamic range trade-off*. It is not a characteristic of the signals themselves, but results rather from the typical adjacent-



■ **Figure 2.** *A feasible software radio receiver.*

channel interference and blocking characteristics defined by the different standards. From this the following can be claimed: Given a fixed digitization bandwidth, the dynamic range of a mobile communications signal diminishes as the channel bandwidth increases. This might need some explanation.

A fixed digitization bandwidth means that independent of the current standard of operation (i.e., the current channel bandwidth), the sample rate and anti-aliasing filter[1] are fixed. Thus, the number of channels that are simultaneously digitized depends on the channel bandwidth of the standard of operation. The narrower the channels, the more channels are digitized. Still, more channels mean more adjacent channel interferers and eventually an increasing dynamic range. If the digitization bandwidth equals the channel bandwidth, the digital signal does not comprise any adjacent channel interferer, which minimizes the dynamic range of the signal. This behavior is shown in Fig. 3, where the typical frequency masks of different mobile communications standards are given. The channel of interest is centered around $f_{carrier}$. While the adjacent channel interferer power depends on the very standard, the in-band dynamic range δ of the channel of interest is supposed to be of the same order in all three cases.

Supposing a digitization bandwidth of 5 MHz, the signal comprises 25 channels in the case of GSM. If the digitization band is located at the edge of the GSM band, these 25 channels include distortions from outside the GSM band. Only three 1.25 MHz channels of IS-95 fit the same 5 MHz bandwidth in practice [4]; in Universal Mobile Telecommunications System (UMTS) it is just one channel. Hence, the GSM channel of interest has 24 interfering adjacent channels, while the IS-95 channel of interest has only two adjacent interferers. The UMTS channel is not interfered with by adjacent channels. Consequently, the latter signal has minimum dynamic range, while the GSM signal has the highest dynamic range of the three signals. In order to show the worst case, the channel of interest has been placed at an edge of the 5 MHz frequency band.

The *bandwidth–dynamic range trade-off* suggests that a very high dynamic range is only necessary in relatively narrow bands where the channels of interest lie. This property can be exploited by nearly all signal processing steps which the signal has to undergo:

- The mentioned noise-shaping ΣΔ ADCs seem to be perfectly tailored to this characteristic [5, 6].
- Filters for SRC can be simplified.
- Channelization filters can be implemented as multirate filters with relaxed constraints.

Implementation Issues

The reader might be used to finding such a section at the end of an article. However, since this discussion has strong influences on the design criteria for the digital front-end, implementation issues are to be dealt with first.

[1] *The anti-aliasing filter is supposed to sufficiently attenuate all signal components outside the digitization bandwidth.*

The Worst Case – A Brute-Force Approach

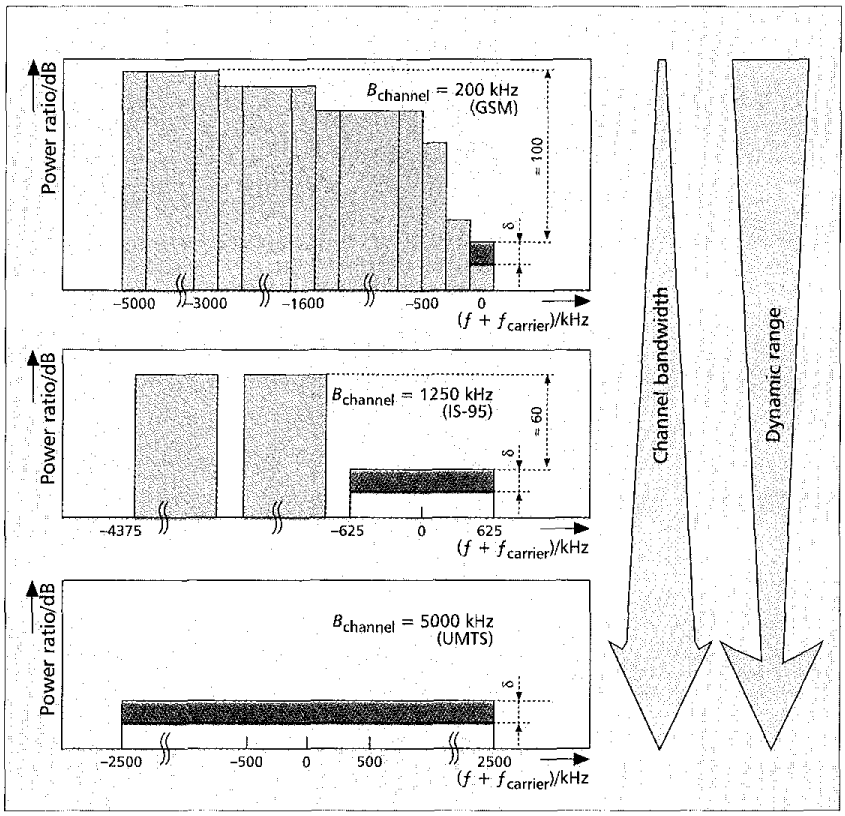
The worst case is if we say that each standard to be processed is something totally different. Therefore, the digital front-end must be changed completely for each standard of operation. As mentioned above, the clock rate and word length are very high. Therefore, today's digital signal processors are not applicable to the digital front-end. Still, with FPGAs there is a means of realizing a signal processing platform that can run at relatively high clock rates and be reconfigured completely [7].

Thus, a dedicated digital front-end can be designed for each standard of operation on a common hardware platform. This approach has several advantages. Not only is the degree of freedom maximized, but also the number of gates currently used is minimized. As an example, a filter could be taken. A reprogrammable filter needs configurable coefficient multipliers, which are usually implemented as standard multipliers. The signal itself acts as one input and the coefficient to be changed as the other. In an FPGA-based solution, not requiring configurable multipliers, a true coefficient multiplier could be used. The number of gates of the latter, however, is on average much lower than that of a standard multiplier of the same word length.

On the other hand, FPGAs can also become a worst-case solution. The degree of freedom FPGAs offer is not necessary. Moreover, it has to be paid for with power consumption and chip area.

Exploiting Commonalities: A Sophisticated Approach

The great challenge when designing a digital front-end of a software radio terminal is the exploitation of commonalities of the different standards of operation. This is especially true if



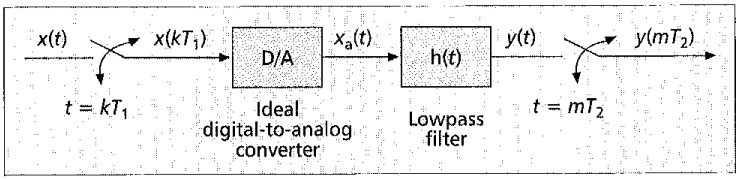
■ **Figure 3.** *An illustration of the bandwidth–dynamic range trade-off.*

the different standards do not seem very similar, for example, spread-spectrum systems in contrast to frequency/time-division multiple access (FDMA/TDMA) systems. In order to avoid the disadvantage of using standard multipliers instead of fixed coefficient multipliers, special algorithms realizing the front-end functionalities have to be found, besides the opportunity for an FPGA-based solution. These are mainly multiplier-free filters, which, as will be seen in the next section, have to be time-varying. Obviously, such algorithms and systems can also be implemented on FPGAs. Still, they also provide the opportunity for implementation on an ASIC, which would considerably empower the software radio concept.

A Smart Approach

As a compromise solution, reconfigurable logic blocks are currently arising as a possible candidate hardware platform for the digital front-end. The limitation of the application to typical functionalities of the digital front-end leads to a limited choice of signal processing principles. This enables the use of optimized logic blocks (e.g., multipliers, coefficient multipliers, integrators) whose interconnections are reconfigurable like the low-level logic blocks of FPGAs.

Having the implementation issues in mind, in the following the main functionalities of the digital front-end are dealt with. These are SRC and channelization.



■ **Figure 4.** *Resampling of a reconstructed time-discrete signal.*

Sample Rate Conversion

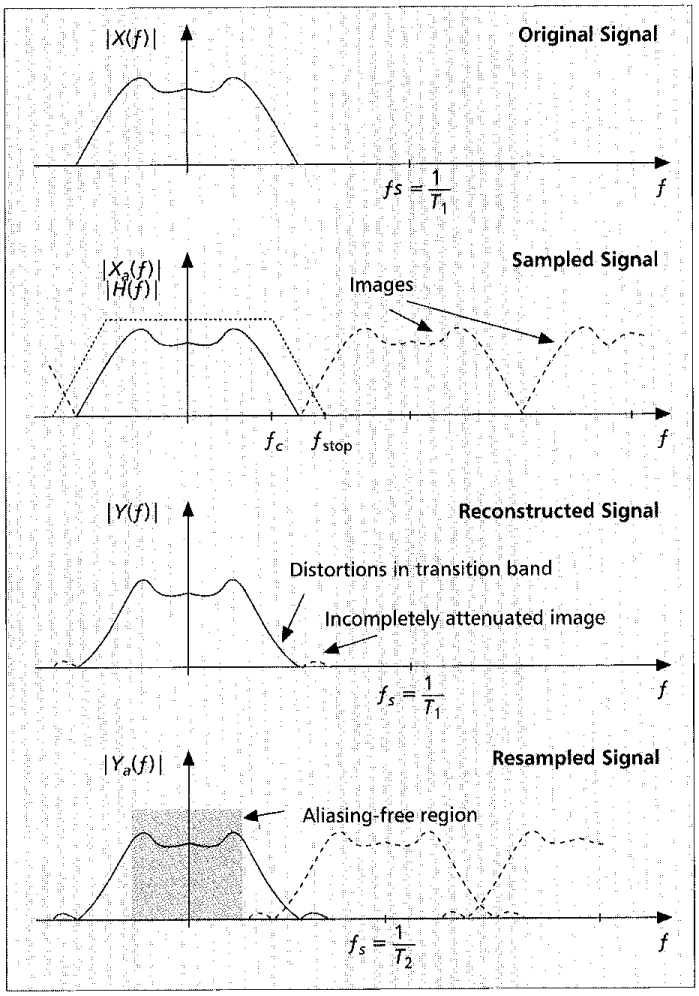
Basic Considerations

The problem of SRC arises when one time-discrete signal has to be calculated from another time-discrete signal, while the sampling positions of both signals differ. Thereby the information of the original signal is to be preserved with respect to the application. For equidistant sampling it can be written in other words: a signal $x(mT_2)$ is to be calculated from signal $x(kT_1)$, which itself is the result of sampling a time-continuous signal $x(t)$ at period T_1. Eventually, signal $x(mT_2)$ should equal the result of hypothetically sampling $x(t)$ with a period of T_2. Obviously, this is a target that can only be achieved under certain conditions, which are to be derived and investigated in this section.

In the following only equidistant sampling periods are considered. Hence, SRC can be characterized by a factor relating the sample rate of both the original (input) signal and the new (output) signal. Since SRC is a process of sampling, the sampling theorem of aliasing-free signal reconstruction has to be obeyed.

The most obvious approach to SRC is the reconstruction of the input signal followed by a resampling process with a new sampling period. A system realizing this approach is sketched in Fig. 4.

Ideally, $y(t) = x(t)$. This is generally true if T_1 is small enough to prevent aliasing and, second, h is an ideal low-pass filter. The whole system of Fig. 4 can be implemented digitally. In general, the resulting digital filter h is time-varying with its coefficients taking sample values of the time-continuous impulse response $h(t)$ [8]. Obviously, there are cases when h does not need to be an ideal low-pass filter (happily, since an ideal low-pass filter cannot be realized!). This is the case if T_1 is much smaller than the sampling theorem dictates, or a certain amount of aliasing is acceptable. The latter is of particular interest since, exploiting the bandwidth–dynamic range trade-off, aliasing components have to be highly attenuated only in a relatively narrow frequency band. Moreover, filter h can be combined with the matched filter (e.g., root raised cosine filter) which is necessary in most receivers. Thus, design constraints for filter h can be relaxed. This might be clearer when interpreting sample rate conversion spectrally.



■ **Figure 5.** *A spectral interpretation of resampling a reconstructed signal.*

Spectral Interpretation of Sample Rate Conversion

In Fig. 5 the spectral interpretation of the different steps of signal processing from sampling signal $x(t)$ to resampling $y(t)$ are sketched.

The reconstruction filter h cancels all signal components resulting from spectral repetition (image components), eventually yielding a band-limited signal. However, a perfect reconstruction filter cannot be realized. For practical applications the constraints

THE INTERPOLATION PROBLEM

$n + 1$ time instances $\tau_a \leq \tau_0 < \tau_1 < \tau_2 < \cdots < \tau_n \leq \tau_b$ are given in the interval $[\tau_a, \tau_b]$. Moreover, $n + 1$ real numbers $y_i (i = 0, 1, \ldots, n)$, for example, representing the samples of a function $y(\tau)$, are given. The interpolation problem is then generally defined as finding a polynomial $I_n(\tau)$ of maximum degree n so that $I_n(\tau_i) = y_i$ for $0 \leq i \leq n$, thus finding a polynomial that takes the originally given samples at the time instances $\tau_i (i = 0, 1, \ldots, n)$. Under these constraints exactly one polynomial $I_n(\tau)$ exists.

on h can be relaxed if a certain amount of aliasing is allowable. As mentioned, only the channels of interest really have to be kept free from aliasing. This directly leads to the following remark.

A Most Important Remark

Anti-aliasing is the most prominent constraint to be obeyed by any sample rate conversion system.

The reconstruction filter can generally be regarded as a filter that attenuates distorting components of $x_a(t)$. These distorting components are those signal components potentially causing aliasing. Depending on the application, the frequency band that has to be kept free from aliasing can be defined. Eventually, the *anti-aliasing* filter h can be designed. Finally, all aliasing components not falling in the frequency band which has to be kept aliasing-free can be removed by filtering after SRC.

Some Notes on Interpolation

The process of calculating a set of samples out of another set of samples at different positions is generally called *interpolation*. Historically, interpolation is a mathematical problem of calculating in-between values of tabulated functions. Based on the notation in [9], the difference between interpolation and anti-aliasing is elaborated with respect to the application to SRC.

The interpolation problem can be formulated as shown in the box on this page.

Interpolation of equidistantly sampled signals can be realized by piecewise finding the interpolation polynomial, which can be interpreted as a filtering operation. If interpolation is realized by means of filtering, a fundamental property of the impulse response of interpolation filters can be derived from the definition of interpolation:

$$h(kT_1) = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{if } k = \pm 1, \pm 2, \ldots \end{cases} \quad (1)$$

This, however, no longer limits the impulse response to be derived by polynomial interpolation. It is interesting to note that the ideal low-pass filter and any derivatives of it obtained by means of windowing are interpolation filters.

Although not only polynomial filters fulfill Eq. 1 (e.g., raised-cosine filters do also) interpolation in the digital domain is usually connected to the application of polynomial filters. This is due to the fact that the sampled impulse response of polynomial filters can be calculated with relatively low effort in real time. In applications where this is not necessary, the range of applicable filters is not limited to polynomial ones.

Polynomial interpolation filters have zeros of the transfer function that tend to be clustered about integer multiples of $2\pi/T_1$ [10]. Among those filters are generalized polynomial interpolation filters. These filters can be designed by building the impulse response from weighted basis polynomials. The weights are obtained by some optimization procedure incorporating the interpolation constraint of Eq. 1 [11]. The typical clustering of the zeros of the transfer function makes polynomial interpolation filters applicable only if the signal to be interpolated is already oversampled and band-limited. This means that the sample rate is several times higher than the Nyquist rate. In this case the relatively small notches of the transfer function of the interpolation filter attenuate the images sufficiently to not cause severe aliasing.

The target of SRC is a signal at a new sample rate with the essential information preserved. Moreover, severe restrictions regarding bandwidth and oversampling ratio of the signal to be converted in sample rate are not acceptable. Therefore, it seems sensible to refrain from the interpolation constraint of Eq. 1 and stick to the spectral interpretation given above and the *Most Important Remark* concluding it. However, there are certainly applications where pure interpolation suffices.

Realizing SRC in Software Radio Terminals

Finally, we shall give some suggestions of how SRC could be realized in software radio terminals:

- An obvious solution is to oversample and band-limit the signal, which eventually enables the application of the above described interpolation. The disadvantage of this approach is the necessity of high sample rates and anti-imaging filters for attenuating the images caused by oversampling. If this approach is to be used anyway, polynomial interpolation filters can advantageously be implemented on the Farrow structure [12].
- A second approach is the approximation of arbitrary impulse responses by piecewise polynomials fulfilling the constraints of the current application. Such impulse responses can also be implemented on the Farrow structure, but require sophisticated controlling [13, 14].
- Finally, comb filters attenuating potential *aliasing* components implemented as cascaded-integrator-comb (CIC) filters [15] are suggested. These filters are highly efficient multiplier-free filters for sample rate increase and decrease by integer factors. Combining both filters for up- and downsampling yields a system performing rational factor SRC. The drawback of a cascade of interpolator and decimator is generally the high intermediate sample rate at which the filters have to operate. However, by exploiting the fact that the input signal to the filters is zero-padded by the upsampling process, implementations of CIC filters are possible that run at the input sample rate rather than the intermediate rate. The resulting *time-variant CIC filters* are a very efficient means of SRC, and were introduced in [16, 17].

While the first two suggestions can principally perform SRC by any real factor, which could lead to a nonperiodical time-varying filter (in case of irrational factors), the latter can only do rational factor SRC, leading to a periodically time-varying filter. Still, the restriction to rational factors does not really limit the applicability since any real factor can be approximated by rational factors. Moreover, since the symbol/chip rates of the different standards can be represented by integer numbers, there is always a rational factor when SRC between those rates is the issue.

Channelization

Basic Considerations

Channelization is a notion that is not uniformly used in the literature. Therefore, we would like to give a definition of channelization: channelization is the functionality comprising all necessary tasks to extract a single user channel for further processing at baseband, thus involving downconversion, filtering, and possibly despreading. The following considerations are based on this definition.

Digital Downconversion

As in the analog case, downconversion to baseband has to be realized by multiplication of the input signal with a rotating complex phasor. In the digital case the samples of the complex phasor can be stored in memory. Also, online generation of the samples is feasible. A special case is if the center frequency of the digitized channel of interest equals a quarter of the sample rate. This is achieved if the center frequency of the channel of interest before digitization equals an odd multiple of a quarter of the sample rate. In this case, considerable simplifications are possible, since the sine and cosine signals representing the rotating complex phasor degenerate to the sequences {0, 1, 0, –1} and {1, 0, –1, 0} [18].

Channel Filtering

Channel filtering is necessary to extract frequency-divided channels. Thus, the filters have to attenuate adjacent-channel interferers and have to meet the blocking characteristics of the current standard of operation. This leads to similar requirements which the ADC has to fulfill with respect to dynamic range. If the software radio terminal is to be implemented on FPGAs, these filters can be designed in an optimized manner for each standard of operation.

However, if a common parameterizable platform is the issue, conventional approaches such as direct finite impulse response (FIR) filters or polyphase filters are not well suited. This is due to the fixed rather than parameterizable nature of these structures. Actually, a sufficiently large number of coefficients and branches (in case of polyphase filters) could be foreseen, but this approach does not lead to an efficient implementation.

Exploiting the fact that in the context of wideband reception a single channel is considerably oversampled, cascaded multirate filters can be applied. They have benefits especially as to efficiency and effort [8]. As mentioned in the previous section, CIC filters are parameterizable multiplier-free filters for integer factor SRC. Therefore, they can advantageously be applied to channelization in software radio terminals. In this case it must be noted that channelization and SRC are realized by the same entity [5, 18]. The partially poor characteristics of CIC filters can be compensated for by a cascade of two or three half-band filters that follow the CIC filter. Moreover, these half-band filters can simultaneously realize chip/symbol-matched filtering.

Despreading

Spread-spectrum systems such as wideband code-division multiple access (W-CDMA) require despreading for final user channel selection. In the context of multipath propagation present in mobile communications channels, the rake-receiver is a prominent solution. Basically, such a receiver consists of several identical parallel rake fingers which themselves comprise a correlator and a decimator. The correlator and decimator are usually realized by a multiplier for multiplying the spreading code and the signal, followed by an integrate-and-dump circuit. Since the spreading code is usually binary or ternary, the multiplier can be realized by a simple switch. Controlling circuitry and code generators complete the rake receiver.

Multiple-Channel vs. Single-Channel Reception

There is a difference if only one channel or multiple channels are to be selected off the input signal. In case of single-channel reception, the filters necessary for SRC can be combined with the channelization filters. This eventually leads to highly efficient implementations that are vital for mobile terminal applications where power consumption is a major issue.

If multiple channels are to be received, the simplest approach is to use several one-channel channelization units in parallel. Still, by combining downconversion and channel filtering in filter banks the effort can be lowered, especially in narrowband systems. Particularly, if only signals of one standard are to be received, discrete Fourier transform (DFT) filter banks are promising candidates as channelizers, since all channels have the same bandwidth [19, 20]. Polyphase filter banks [8, 21], one subclass of uniform DFT filter banks, have the desirable characteristic that the relative complexity tends to decrease as the number of channels to be separated increases, in contrast to the parallel implementation of one-channel-channelizers. Uniform DFT filter banks split the frequency band $[0, f_S)$ into an integer number of subbands. These subbands should represent the different channels. Thus, the sample rate f_S must be an integer multiple of the number of channels. Still, earlier it was suggested that one choose the digitization rate standard independently fixed. Since, moreover, the channel spacing generally does not equal the symbol/chip rate, SRC is necessary before *and* after channelization. A solution to overcome the limitation of the integer ratio between the sample rate and the number of channels could be the application of nonuniform filter banks.

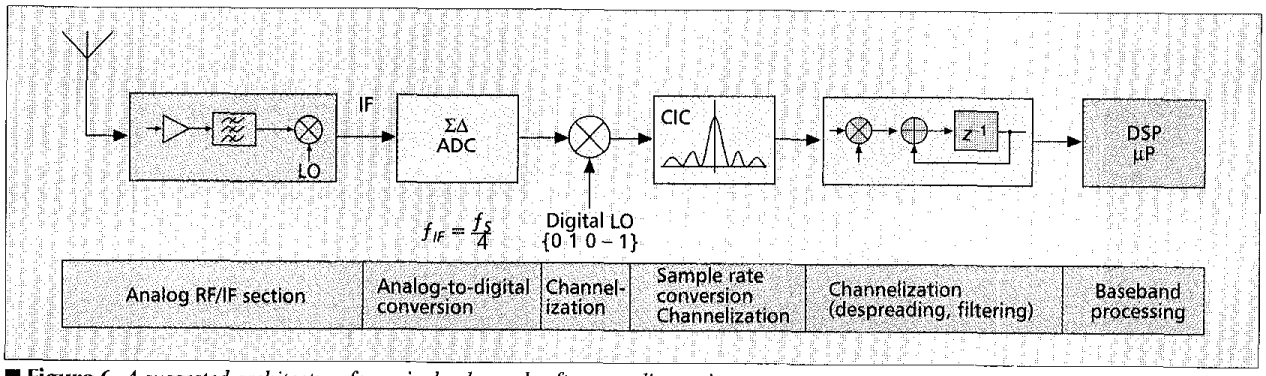
Exploitation of Commonalities

As mentioned earlier, a great challenge is the exploitation of commonalities of the signal processing algorithms required by the different standards of operation of the software radio terminal. In the current section we have seen that narrowband systems require narrowband channel filtering, and spread-spectrum systems, usually having a wide bandwidth, require despreading. While the first task can mathematically be described by a convolution of the signal with the impulse response of the filter, the latter is a correlation. It is well known that convolution can be regarded as a correlation by reversing one of the input arguments. This mathematical commonality between despreading and filtering suggests an implementation of both operations on one common hardware platform. Since spreading codes are possibly very long, it does not seem sensible to implement the despreading by means of a filtering operation, but rather to implement the narrowband channel filter by means of the hardware used for decorrelation, namely the rake receiver [22]. In this case the multiplier each rake finger employs must be able to cope with the word length of the filter coefficients. Still, there are techniques to realize filters having low word length coefficients or even ternary valued coefficients [23, 24]. This would enable one to keep the simple structure of the rake finger. In Fig. 6 a receiver architecture for single channel reception is suggested. This receiver employs ΣΔ ADC at IF, combined SRC and channel-filtering by means of time-varying CIC-filters [16, 17], and combined channel filtering/despreading [22]. The channel filtering task is distributed between the CIC filter and the filter/correlator based on the rake-finger structure.

It should be noted that the proposed combination of filtering and despreading also has great impact on software design. As an example, application-tailored processors are mentioned. Such processors could employ dedicated substructures for each rake finger [25, 26], and thus require dedicated software structures.

Conclusions

The evolution from dedicated mobile communications terminals to software-defined terminals strongly depends on the available hardware platform enabling dedicated software to run, and thereby defining the current mode of operation. Particularly the front-end of the terminal imposes strong requirements regarding dynamic range and bandwidth, and



■ **Figure 6.** *A suggested architecture for a single-channel software radio receiver.*

thus word length and sample rate. The main functionalities of the digital front-end, sample rate conversion and channelization, are realizable on reconfigurable hardware (e.g., FPGAs). However, we have shown that sensible exploitation of characteristics of mobile communications signals and commonalities among the different standards can lead to a generalized hardware platform. Such a platform could be implemented on an ASIC. Being parameterizable and fulfilling the requirements of all intended modes of operation, such an ASIC would empower the software radio concept. A compromise between the high degree of freedom FPGAs offer and the comparatively low complexity of an ASIC solution are reconfigurable logic blocks, which provide the structural flexibility of FPGAs in connection with optimized signal processing blocks.

On the other hand, the reader has certainly encountered several *ifs and buts* indicating that the ideal software radio receiver is still of theoretical nature, but being approached by practice at a rapid pace.

Acknowledgments

This work has been partially supported by the European Commission, ACTS Project *Software Radio Technology*.

References

[1] W. H. W. Tuttlebee, "Software-Defined Radio: Facets of a Developing Technology," *IEEE Pers. Commun.*, vol. 6, no. 2, Apr. 1999, pp. 38–44.
[2] "Digital cellular telecommunications system (Phase 2+), Radio transmission and reception," GSM 05.05 v. 5.9.0, rel. 1996, pr ETS 300 910, July 1998, draft.
[3] R. H. Walden, "Performance Trends for Analog-to-Digital Converters," *IEEE Commun. Mag.*, Feb. 1999, pp. 96–101.
[4] R. Prasad and T. Ojanpera, "An Overview of CDMA Evolution toward Wideband CDMA," *IEEE Commun. Surveys*, 1998, http://www.comsoc.org/pubs/surveys
[5] T. Hentschel, G. Fettweis, and M. Bronzel, "Channelization and Sample Rate Adaptation in Software Radio Terminals," *3rd ACTS Mobile Commun. Summit*, Rhodos, Greece, June 1998, pp. 121–26.
[6] T. Hentschel and G. Fettweis, "AD Conversion and Channelization for Multi-Mode Terminals," *MTT-S Euro. Wireless '98*, Oct. 1998, pp. 169–74.
[7] M. Cummings and S. Haruyama, "FPGA in the Software Radio," *IEEE Commun. Mag.*, Feb. 1999, pp. 108–12.
[8] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Prentice Hall, 1983.
[9] I. N. Bronstein and K. A. Semendjajew, *Teubner-Taschenbuch der Mathematik*, vol. 1, Leipzig: B. G. Teubner, 1996.
[10] R. W. Schafer and L. R. Rabiner, "A Digital Signal Processing Approach to Interpolation," *Proc. IEEE*, vol. 61, no. 6, June 1973, pp. 692–702.
[11] J. Vesma and T. Sarama ki, "Interpolation filters with arbitrary frequency response for all-digital receivers," *Proc. IEEE Int'l. Symp. Circuits and Sys.*, May 1996.
[12] C. W. Farrow, "A Continuously Variable Digital Delay Element," *IEEE Int'.. Conf. Circuits and Sys. 1988*, pp. 2641–45.
[13] T. A. Ramstad, "Fractional Rate Decimator and Interpolator Design," *Proc. EUSIPCO, 1998*, pp. 949–1052.
[14] L. Lundheim and T. A. Ramstad, "An Efficient and Flexible Structure for Decimation and Sample Rate Adaptation in Software Radio Receivers," *Proc. ACTS Mobile Commun. Summit*, June 1999.
[15] E. B. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation," *IEEE Trans. Acoustics, Speech and Sig. Process.*, vol. ASSP-29, no. 2, Apr. 1981, pp. 155–62.
[16] T. Hentschel, M. Henker, and G. Fettweis, "Sample Rate Conversion in Software Radio Terminals," ACTS Mobile Commun. Summit, Sorrento, Italy, June 1999.
[17] M. Henker, T. Hentschel, and G. Fettweis, "Time-variant CIC-filters for sample-rate conversion with arbitrary rational factors," *Proc. IEEE 6th Int'l. Conf. Elect., Circuits and Sys.*, Paphos, Cyprus, Sept. 1999.
[18] K. Boehm *et al.*, "An IF Digitizing Receiver for a Combined GPS/GSM Terminal," IEEE Radio and Wireless Conf., Colorado Springs, CO, Aug. 1998.
[19] T. Turletti and D. Tennenhouse, "Complexity of a Software GSM Base Station," *IEEE Commun. Mag.*, Feb. 1999, pp. 113–17.
[20] D. B. Chester, "Digital IF Filter Technology for 3G Systems: An Introduction," *IEEE Commun. Mag.*, Feb. 1999, pp. 102–7.
[21] P. P. Vaidyanathan, "Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial," *Proc. IEEE*, vol. 78, no. 1, Jan. 1990, pp. 56–93.
[22] T. Hentschel and G. Fettweis, "Software radio receivers," *CDMA Techniques for Third Generation Mobile Systems*, F. Swarts, P. van Rooyen, I. Opperman, and M. P. Lotter, Eds., Ch. 10, Kluwer, 1999, pp. 257–83.
[23] N. Benvenuto, L. E. Franks, and F. S. Hill, Jr., "Realization of Finite Impulse Response Filters Using Coefficients +1, 0, and -1," *IEEE Trans. Commun.*, vol. COM-33, no. 10, Oct. 1985, pp. 1117–25.
[24] N. Benvenuto, L. E. Franks, and F. S. Hill, Jr., "Dynamic Programming Methods for Designing FIR Filters Using Coefficients -1, 0, and +1," *IEEE Trans. Acoustics, Speech, and Sig. Proc.*, vol. ASSP-34, no. 4, Aug. 1985, pp. 785–92.
[25] G. P. Fettweis *et al.*, "Breaking New Grounds Over 3000 MOPS : A Broadband Mobile Multimedia Modem DSP," *Proc. ICSPAT '98*, Toronto, Canada, Sept. 13–16, 1998.
[26] M. Weiss, F. Engel, and G. P. Fettweis, "A New Scalable DSP Architecture for System on Chip (soc) Domains," *Proc. ICASSP '99*, Phoenix, AZ, Mar. 15–19, 1999.

Biographies

GERHARD FETTWEIS [SM '98] (fettweis@ifn.et.tu-dresden.de) received his M.Sc./Dipl.-Ing. and Ph.D. degrees in electrical engineering from the Aachen University of Technology (RWTH), Germany, in 1986 and 1990, respectively. From 1990 to 1991 he was a visiting scientist at the IBM Almaden Research Center, San Jose, California, working on signal processing for disk drives. From 1991 to 1994 he was a scientist at TCSI, Berkeley, California, responsible for signal processor developments for mobile phones. Since September 1994 he holds the Mannesmann Mobilfunk Chair for Mobile Communications Systems at the Dresden University of Technology, Germany. He is an elected member of the SSC Society's Administrative Commitee, and of IEEE ComSoc Board of governors, since 1999 and 1998, respectively. He has been associate editor for *IEEE Trans. on CAS II*, and now is associate editor of *IEEE Journal on Selected Areas of Communications*' wireless series.

MATTHIAS HENKER (henker@ifn.et.tu-dresden.de) received his M.Sc./Dipl.-Ing. degree from the Dresden University of Technology, Germany, in November 1998. In his thesis he analyzed algorithms for sample rate conversion in software programmable mobile communications receivers. In December 1998 he joined the Mannesmann Mobilfunk Chair for Mobile Communications Systems at the Dresden University of Technology, Germany. His main research interests include software radio, especially dual- and multimode terminals.

TIM HENTSCHEL [S '93] (hentsch@ifn.et.tu-dresden.de) received his M.Sc./Dipl.-Ing. degree in electrical engineering from King's College London, University of London, United Kingdom, and the Dresden University of Technology, Germany, in 1993 and 1995, respectively. From 1995 to 1996 he was with Philips Communications Industries, Nürnberg, Germany. Since May 1996 he is with the Mannesmann Mobilfunk Chair for Mobile Communications Systems, working toward his Ph.D. His current research interests include software radio, specifically the design and investigation of digital signal processing algorithms for reconfigurable front-ends.