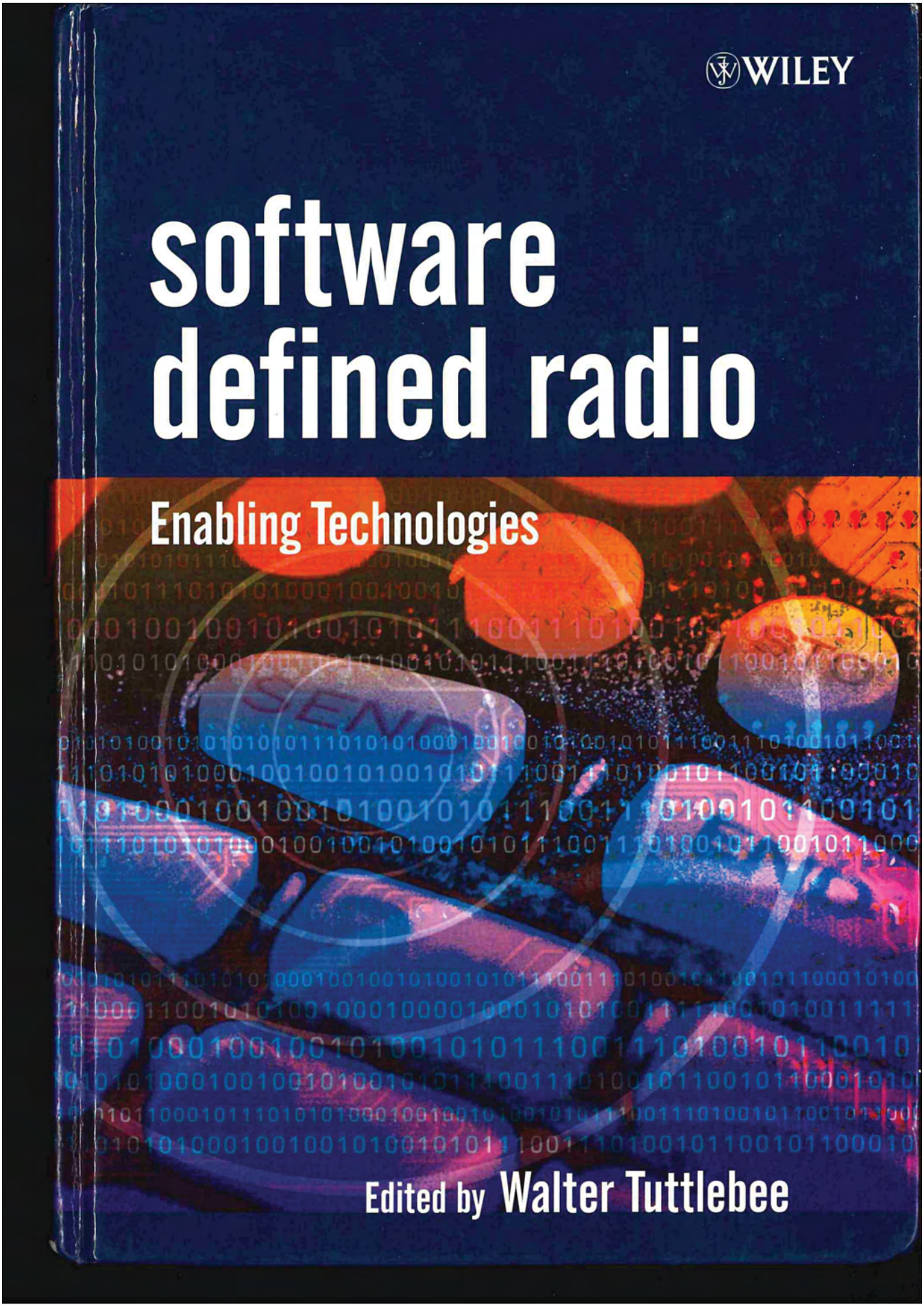

software defined radio
Enabling Technologies
Tuttlebee
(Editor)
WILEY

**Wiley Series in Software Radio**

Series Editor: Dr Walter Tuttlebee, Mobile VCE, UK

The Wiley Series in Software Radio aims to present an up-to-date and in-depth picture of the technologies, potential implementations and applications of software radio. Books in the series will reflect the strong and growing interest in this subject. The series is intended to appeal to a global industrial audience within the mobile and personal telecommunications industry, related industries such as broadcasting, satellite communications and wired telecommunications, researchers in academia and industry, and senior undergraduate and postgraduate students in computer science and electronic engineering.

*Mitola:* Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering, *0471384925, 568 Pages, October 2000*
*Mitola and Zvonar (Editors):* Software Radio Technologies: Selected Readings: *0780360222, 496 Pages, May 2001*
*Tuttlebee:* Software Defined Radio: Origins, Drivers and International Perspectives, *0470844647, £55, 350 pages*
*Tuttlebee:* Software Defined Radio: Enabling Technologies, *0470843187, £55,304 pages*

# Software Defined Radio

## Enabling Technologies

Edited by

**Walter Tuttlebee**
*Virtual Centre of Excellence in Mobile & Personal Communications (Mobile VCE)*

JOHN WILEY & SONS, LTD

 John Wiley & Sons Ltd, Baffins Lane, Chichester,
West Sussex PO19 1UD, England

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on www.wileyeurope.com or www.wiley.com



***Other Wiley Editorial Offices***

John Wiley & Sons Inc., 605 Third Avenue, New York, NY 10158-0012, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Pappelallee 3, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 33 Park Road, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 22 Worcester Road, Etobicoke, Ontario, Canada M9W 1L1

***British Library Cataloguing in Publication Data***
A catalogue record for this book is available from the British Library

ISBN 0-470-84318-7

Typeset in 10/12pt Times by Deerpark Publishing Services Ltd
Printed and bound in Great Britain by Biddles Ltd, Guildford and King's Lynn

This book is printed on acid-free paper responsibly manufactured from sustainable forestry in which at least two trees are planted for each one used for paper production.

# Contents

For a 10 dB gain in the output amplifier, the TOI of the buffer block would only need to be 10 dB less than that of the high power amplifier in order to have the same distortion contribution in the output to the antenna. Recognition of this is important, as power control is often affected by nonlinear elements such as PIN diodes.

#### 2.3.3.5 Mixers

Mixers may also have a crucial impact on transceiver linearity. The higher the signal level that is input into the mixer, the more critical the distortion performance of the mixer becomes to the overall distortion performance of the transceiver. Diode mixers do not have especially good TOI and other solutions may be needed. The use of highly linear resistive FET mixers [22], or mixer linearization techniques [23–25] could provide a solution. Some successful work has been done using predistortion to linearize mixers, for example, with results to date reported in [26].

The nonlinear behavior of the transmitter mixers not only results in the generation of IMD products but also of related spurious signals that arise from the combination of the LO and IF signals.

## 2.4 Candidate Architectures for SDR

This section critically discusses techniques that could provide ways forward with SDR hardware design. The techniques are analyzed, and their deficiencies highlighted.

### *2.4.1 Zero IF Receivers*

Superheterodyne receivers have a number of problems as far as their application in an SDR is concerned. These problems largely revolve around the complexity of the architecture, and the fact that this architecture must deal with image signals.

These problems could be substantially solved if *direct conversion* architectures could be employed [27]. A typical direct conversion receiver is shown in Figure 2.2. As well as reducing the complexity of the receiver circuit, a direct conversion architecture has image suppression properties and thus the design of image suppression circuitry is not as critical.[12] We investigate this property before going on to discuss in detail the downside of the zero IF architecture (a broad outline of these problems has already been presented earlier in the chapter).

#### 2.4.1.1 Image Rejection Properties

The image signal is separated from the wanted signal by twice the IF frequency. It can therefore be assumed that if the IF frequency were to be set to zero, there would be no image problems. This is not quite the case, as the image signal turns out to be a frequency-inverted image of the wanted signal itself and, although some image rejection advantages can be achieved, the image cannot be ignored. To achieve direct down-conversion of a complex

[12] By the same token, the preselect filter as a tool for removing the image signal is not available to the designer of a zero IF receiver.

signal requires a local oscillator that is itself a complex signal, with components in $I$ and $Q$ branches that are in precise phase quadrature and precise amplitude balance. If this is achieved, then the image signal will be eliminated. If precise phase quadrature and amplitude balance for the local oscillator is not achieved, however, then a small sideband will be superimposed on the wanted sideband. This will result in magnitude and phase errors for the received signal constellation. Because the image signal in a direct conversion receiver is derived from the wanted signal itself, and not from a high power blocking signal, the requirements for image rejection (and by implication amplitude balance and phase quadrature of the local oscillator) are not quite as great as would be the case in superheterodyne architecture. The question of what is a reasonable amplitude and phase balance is taken up again in Section 2.4.2.

### 2.4.1.2 Problems with Zero IF architecture

*DC Offset* A significant problem with direct conversion architecture is the introduction, by the process, of a DC offset. This DC offset can arise from a number of sources. One typical source is illustrated in Figure 2.24. It can be seen that leakage of a local oscillator signal to the input of the LNA, or the mixer, can occur either through imbalance in the mixer, or via capacitive or inductive (substrate or bond wire) coupling within the LNA. Once these signals are in the system, they are then mixed with themselves to produce a DC voltage at the input to

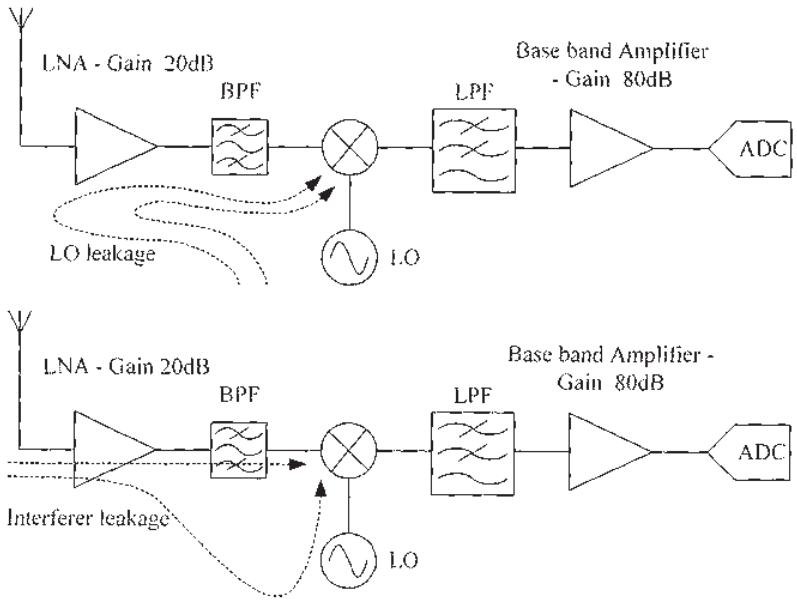


**Figure 2.24** Sources of DC offset

the baseband amplifier. This phenomenon is known as self-mixing. It can be seen that a high power interfering signal can also generate a self-mixed DC offset. This process is shown in the lower diagram of Figure 2.24.

Figure 2.25 puts some illustrative figures on this process to provide an appreciation of this DC offset level compared to the wanted signal. Assuming that the local oscillator is at a level of 0 dBm, and the attenuation this signal experiences in reaching the input of the LNA is 80 dB, this results is an effective LO signal of $-80$ dBm at the input of the receiver. This will result in a self-mixed DC signal level of $+20$ dBm at the input to the ADC. If it is assumed that the receiver is designed to have a wanted power level at the input to the ADC of 0 dBm, then the self-mixed signal will be 20 dB greater than the wanted signal.

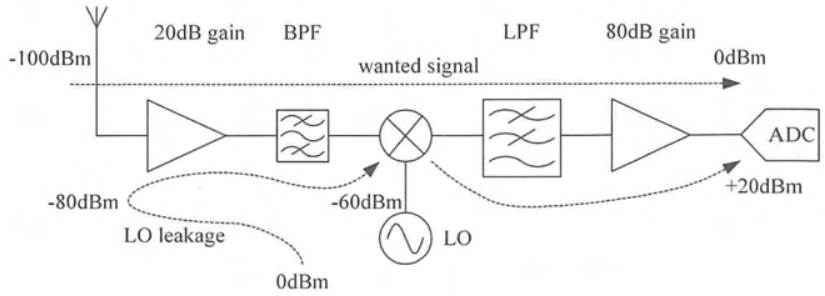


**Figure 2.25** Some typical signal levels for self-mixed signals

In such a situation, a DC offset greater than the wanted signal will place a limit on the amount of amplification possible with a direct conversion receiver. This happens because the offset (rather than the wanted) signal will determine the power level at which the receiver amplifier saturates.

Because the antenna impedance varies with the surroundings in which the receiver is placed, it is possible that the reflection of the LO signal from this antenna will also vary with time, as the mobile station moves. This will mean that the DC offset will also vary with time and, while it is possible to cancel DC offsets that are stationary, cancelling offsets that are time varying is more difficult.

The most obvious way to remove a DC offset is to AC couple the signal. This introduces new problems, however, in that the spectrum of a number of binary (or m-ary) modulation schemes has a peak at DC. It has been found through simulations that most of this peak needs to be preserved. Thus, any high pass filter used to filter the data requires a long time constant. This in turn makes the circuit slow to respond to time varying offsets.

An alternative is to use a modulation scheme with no DC component at all. Clearly such a design choice is not available to the designer of an SDR receiver.

A second alternative is to adaptively cancel the offset. TDMA systems are particularly well suited to this technique since the time period between bursts can be used to measure the offset, which may then be subsequently subtracted from the signal during the active part of the signal.

***1/f Noise*** A direct conversion receiver has a large amount of its gain placed at baseband frequencies. In the previous illustrative example in this chapter we have assumed a gain of 20 dB in the LNA, and the remaining 80 dB at baseband. The overall noise figure of a cascade of amplifiers is given by Equation (2), which is restated here:

$$F = F_1 + \frac{F_2 - 1}{G_1} \tag{10}$$

where $F$ is the overall noise figure of the receiver, $F_1$ is the noise figure of the LNA, $F_2$ is the noise figure of the baseband amplifier and $G_1$ is the gain of the LNA. It can be seen that the noise contribution of the second stage is reduced by the gain of the first stage. However, the second stage will have an excess noise figure due to its operation at baseband, which incorporates a $1/f$ (flicker noise) component. This can mean that the noise performance of the second stage becomes just as critical as the first.

***Second-Order Distortion*** Third-order distortion is an important parameter in the design of a conventional heterodyne receiver. A two-tone test performed with frequencies $f_1$ and $f_2$ will generate third-order distortion components, $2f_1 - f_2$ and $2f_2 - f_1$, which are in-band, and will stay in-band throughout the complete down-conversion process. Second-order distortion components $f_1 - f_2$ and $f_2 - f_1$ are out-of-band for conventional heterodyne structures and thus need not be considered. In a direct conversion receiver the components $f_1 - f_2$ and $f_2 - f_1$ are still out-of-band in the RF part of the receiver but fall within the passband of the baseband amplifier. These components may reach baseband by:

- direct leakage through the mixer
- the second harmonic of the RF signal mixing with the second harmonic of the local oscillator signal

Second-order distortion[13] will demodulate any amplitude modulation present on the signal, so variations on the signal amplitude caused by, for example, fading may result in a spurious term being present at the receiver output. Second-order distortion, like third-order distortion, is characterized by an intercept point $IP_2$ (cf. $IP_3$ or alternatively TOI, for third-order distortion). Second-order distortion is usually tackled by using a differential circuit structure. This type of structure is ideally suited to realization in integrated circuit format.

***Retransmission of Local Oscillator Signal*** The local oscillator signal not only causes problems by being *reflected* off the antenna and self-mixed down to baseband, but it can also cause problems by being transmitted from the antenna. Because direct conversion is being used, the local oscillator is 'in-band,' and may thus cause interference to adjacent receivers.

### 2.4.2 *Quadrature Local Oscillator*

One of the predominant reasons designers of SDR receivers or transmitters are likely to reject direct conversion architecture has to do with the difficulty of generating a wideband quadrature local oscillator that can cover the complete frequency band over which the SDR will be

[13] Or in fact any form of even order distortion.

required to operate. The local oscillator will track the center frequency of the signals being received, and will have to maintain precise phase quadrature, and precise amplitude balance, over that range of frequencies.

It can be shown [27] that the image rejection ratio (IRR) of an image reject mixer is given by

$$\mathrm{IRR} = \frac{1 - 2(1+\epsilon)\cos\theta + (1+\epsilon)^2}{1 + 2(1+\epsilon)\cos\theta + (1+\epsilon)^2} \tag{11}$$

where $\epsilon$ is the unbalance in the magnitude of the local oscillator signals, and $\theta$ is the phase deviation from 90°. This equation is plotted in Figure 2.26, from which it can be seen that in order to maintain an image suppression of better than $-40$ dB, the phase of the LO needs to be matched to better than 1° and the amplitude to better than 0.1 dB. Such tight tolerances are very difficult to achieve over the wide frequency range that an SDR may be asked to cover.

There have been a number of attempts to produce a wideband quadrature local oscillator [28–32]; the most promising would appear to be the wideband quadrature oscillator described in [28]. This circuit is a combined resistor capacitor lowpass-highpass network with amplitude stabilization being effected by feedback. The circuit, realized in GaAs technology, produces a precise quadrature output with balanced amplitude. Performance of the IC is impressive – it operates over a frequency range extending from 250 MHz to 4 GHz. Sideband rejection exceeds 60 dB at 1 GHz, falling to 50 dB at 4 GHz.

An important downside is that the power consumption of the device is 1.5 W making it unsuitable for portable applications as yet. It has, however, been used in the Agilent company's measurement instrumentation products and, as such, it tends to be an 'in-house' chip; to the authors' knowledge, the chip is not at present available to the wider technical community. Its existence does indicate, however, that viable commercial solutions

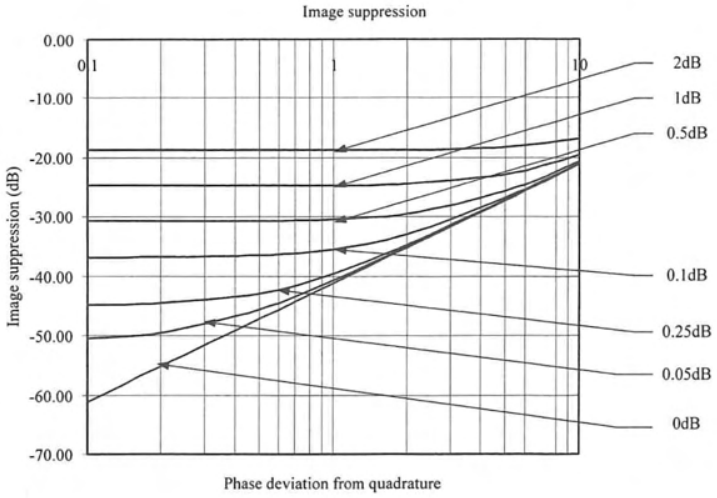


**Figure 2.26** Image suppression as a function of LO phase and amplitude imbalance

are possible, and the possibility of having wideband, accurate, quadrature oscillators available within the next 5 years is very high. The availability of such devices, at lower power consumption, will result in direct conversion becoming a much more viable architecture for the SDR application.

Some work using transmission line stubs, and path differentials, to set up quadrature signals, has been reported by Warr et al. (see [30,31]). Phase balance within 1.0° with amplitude balance of 0.3 dB, over a bandwidth approaching one octave, is reported on an essentially 'open loop' system.

### *2.4.3 Variable Preselect Filters*

As an alternative to image reject mixing, it is possible to employ electronically variable preselect filters. Traditionally tuned filters have relied on varactor tuning; however, to meet the requirements of SDR, new approaches such as tunable dielectric properties or the use of MEMS switching are likely to be needed.

#### 2.4.3.1 Design Issues

The frequency range of wanted signals for the range of the receiver air interface standards that we have been considering can be represented graphically, as shown in the upper line of Figure 2.27. The second line of Figure 2.27 shows the coverage required of preselect filters. It can be seen that four filters are required. The table at the bottom of this diagram lists the performance required of these filters in terms of the upper and lower cut-off frequencies. This table assumes that the bandwidth of the preselection filter is limited to about 5% of the filter center frequency, and that any bandwidth required over and above this amount can be obtained by sweeping the filter characteristics. Five percent bandwidth tends to be a reasonable figure as, when designing distributed component filters, realization starts to become difficult when the filter bandwidth approaches 1%.

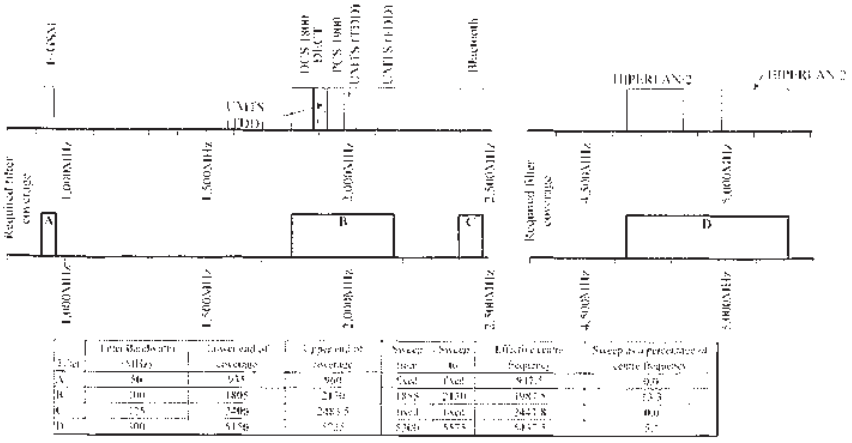


**Figure 2.27** Frequency of operation of an SDR image reject filter

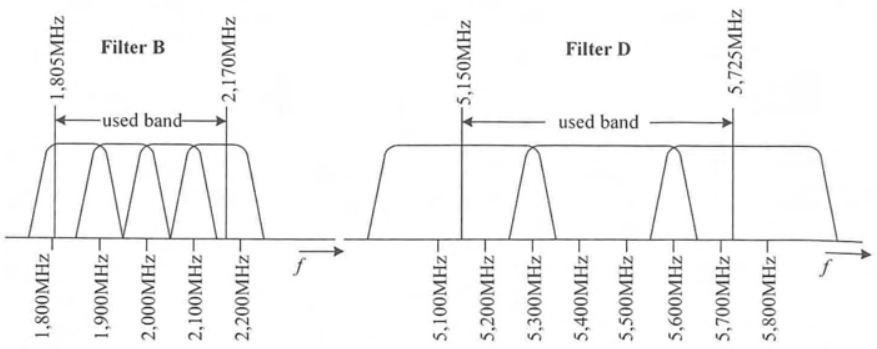


**Figure 2.28** Filter coverage required for the UMTS, DCS1800, and DECT band and for the HIPERLAN/2 band

One possible arrangement for the variable filters is shown in Figure 2.28, where a constant bandwidth filter is proposed (100 MHz in the case of the lower frequency filters, and 300 MHz in the case of the HIPERLAN/2 filters). The low frequency filter will be swept in four steps to cover the UMTS, DCS1800, and DECT band, and the high frequency filter in three steps to cover the HIPERLAN/2 band.

To give some practical basis to the discussion of design issues, let us assume that the first IF frequency is chosen to be 160 MHz, a reasonable choice for the first IF of the 2 GHz frequency band. Now, if we assume that the wanted signal is sitting at the low frequency edge of the filter pass band, then the worst case would be if the image signal is above the filter pass band in frequency (high side conversion), at a frequency of 320 MHz (i.e. twice the IF) above the wanted signal. The image reject filter does not commence 'cutting off' until 100 MHz. The amount it cuts off in the remaining 260 MHz will determine the image rejection performance of the receiver, as illustrated in Figure 2.29.

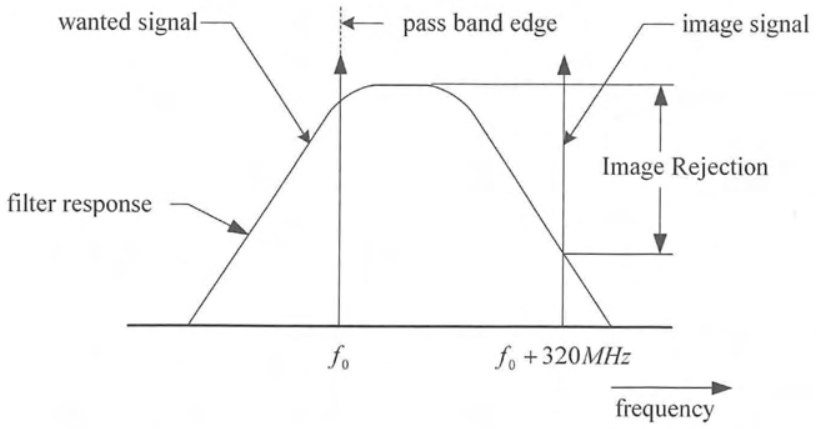


**Figure 2.29** Effect of filter response on receiver image rejection performance

Using this approach we can calculate the order of the filter we would require for our image reject filter. The standard formulae for the order of a Chebyshev filter required to give a particular pass band loss is given by Equation (12) (see [33]).

$$n = \frac{\cosh^{-1}\left[\left(10^{\frac{\alpha_{\min}}{10}} - 1\right)\left(10^{\frac{\alpha_{\max}}{10}} - 1\right)\right]^{0.5}}{\cosh^{-1}\omega_s} \tag{12}$$

where $n$ is the order of the filter, $\alpha_{\max}$ is the maximum permitted attenuation in the pass band (which is approximately the image rejection expressed as a linear ratio), and $\omega_s$ is the normalized frequency that defines the beginning of the stop band, and $\alpha_{\min}$ is the minimum permitted attenuation in the stop band.

### 2.4.3.2 Practical Design of Image Reject Filter

An image reject filter will always operate at RF; therefore the option for the design of a flexible preselect filter is at present limited to realization as either a distributed component design or as a MMIC.

There have been a few variable MMIC filter designs reported in the literature, notably that of Katzin, reported in [34], produced as a prototype MMIC for the Hittite Corporation. Two versions were produced, both exhibiting $\approx$ 100 MHz bandwidth. One had a center frequency that could be swept from 1500 to 2,000 MHz and the other a center frequency sweepable from 1980 to 2650 MHz. The filter unfortunately never progressed beyond the prototype stage, due to lack of sufficient dynamic range [35]. Nonlinearity is introduced into MMIC designs from two sources. First, it occurs because a varactor is often used to tune the device, and second,a nonlinear active device is often used to compensate losses in MMIC filter components.[14]

There are several classic types of distributed component microwave filters. If we restrict consideration to those filters which could conceivably be realized in microstrip, or similar, technology, then we are left with the following list.

- end coupled microstrip
- edge coupled microstrip with open circuit or short circuit termination
- interdigital microstrip
- combline microstrip
- hairpin microstrip.

Most of these filter architectures were developed in the late 1950s and early 1960s [36,37]. Figure 2.30 illustrates the essential geometry of these filter types.

All of these filter architectures are designed for a fixed center frequency, fixed bandwidth application; the question thus remains as to how they might be electronically tuned. A number of suggestions are listed below.

- varactor diode tuning at some strategic point on the filter structure;
- constructing the filter on a substrate, whose dielectric constant could be electrically varied;
- switching parts of the transmission line so that the physical characteristics of the filter structure could be changed.

[14] By introducing, for example, negative resistance across an inductor.

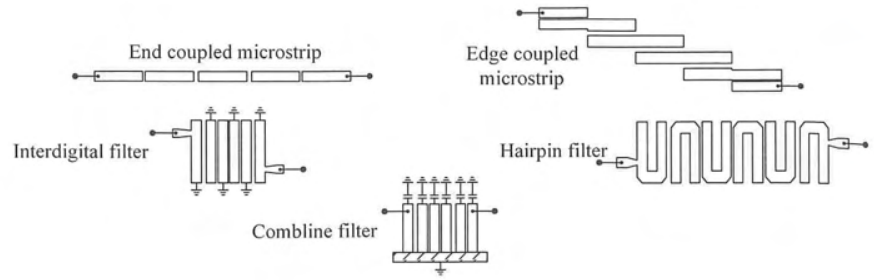


**Figure 2.30** classical microstrip filters

Varactor diode tuning has been investigated with Combline filters [38]. Filter designs are reported in which the center frequency can be swept from 3200 MHz to 4800 MHz with a bandwidth of about 200 MHz. Reported insertion loss for such a filter is of the order of 5 dB. Such a filter structure is likely to exhibit distortion problems, because of the presence of the nonlinear, varactor diodes.

It would be possible to sweep the filter characteristic by sweeping the effective dielectric constant of the substrate. As the electrical length of a transmission line is inversely proportional to the square root of the dielectric constant, this will cause the center frequency of the filter to vary. The substrate would allow the dielectric constant to change in response to variation in an electrical bias. Such a substrate material has been developed by a research laboratory in the United Kingdom. This technology has been subsequently sold on to a third party, and its commercial future is presently uncertain.

Switching the component parts of a filter, in and out of circuit, using micro-electro-mechanical structures (MEMS) seems to offer an alternative solution [39]. The use of electro-mechanical switches will mean that the filter is composed entirely of linear components, and therefore the dynamic range of the filter should not be prejudiced. The major problem with electrically switching a filter is to preserve the filter geometry as the center frequency is translated, while at the same time utilizing an essentially simple switching arrangement. Structures, such as edge coupled lines, or interdigitized filters, have geometry problems as switching extends lines.

At the time of writing, the simplest arrangement for tuning the filter characteristic would appear to be the modified hairpin structure shown in Figure 2.31 [40–42]. This filter has a coupled line, which loads the top of the hairpin and forms part of the filter resonator. Inter-stage transformer action is brought about by edge coupling of the U-shaped structures.

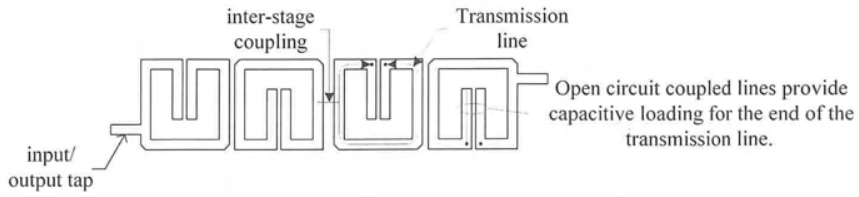


**Figure 2.31** Switching filter center frequency using MEMS

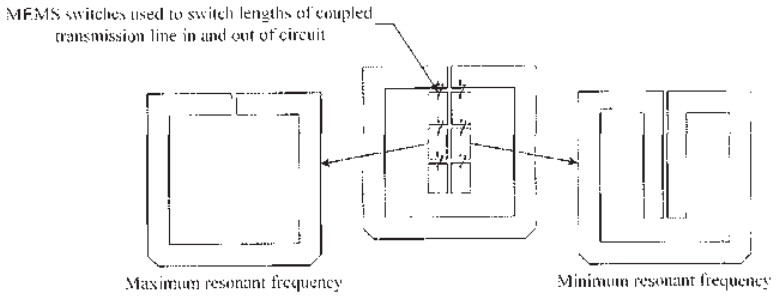


**Figure 2.32** Switching filter center frequency using MEMS

Tuning of this filter using MEMS can potentially be achieved by shortening the top loading coupled line, as shown in Figure 2.32. A simple filter of this type has been simulated on advanced design systems (ADS); the results are shown in Figure 2.33.

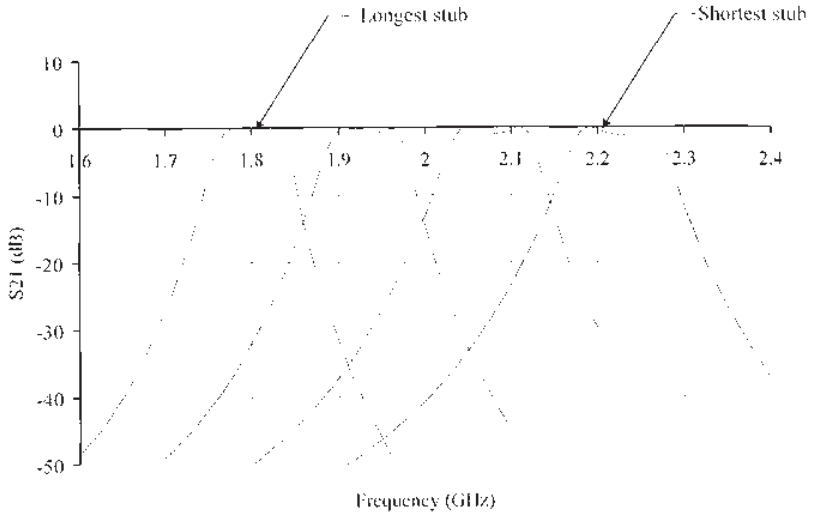


**Figure 2.33** Simulated response of MEMS-tuned hairpin filter

### 2.4.3.3 The Potential Role of MEMS Technology

The potential role of MEMS technology goes beyond the example just discussed. In the recent past there has been an explosive growth of interest in MEMS components [43,44]. It is possible to use MEMS technology not only to realize essentially simple MEMS switches but also to realize quite complex mechanical shapes. Micro sized resonator structures become a possibility. This means that filters and duplexers may be implemented as MEMS components. It may not be necessary to actually tune a filter, but potentially the same effect may be achieved using MEMS switches to configure a path through a large number of fixed frequency 'micro filters.'

As an alternative, it may be possible to vary the geometry of a MEMS filter to effectively tune a filter. It is possible, for example, to tune a MEMS filter by physically moving the fingers of an interdigital capacitor in and out of mesh.

It may be that MEMS components will lead the way in making the breakthroughs required to realize much of the potential of SDR. Certainly this is an exciting area that could allow some longstanding design ground rules to be rewritten.

### *2.4.4 Low IF Receivers*

As discussed earlier, in Section 2.4.1, the principal problems associated with the direct conversion (or zero IF) architecture are the realization of a wideband quadrature oscillator and the DC offset caused by self-mixing. Low IF architecture is an endeavor to separate the wanted signal from the DC offset via the use of a low, but nonzero, final IF frequency, thereby retaining all the advantages of a zero IF stage, while eliminating one of the most significant problems (DC offset).

The signal spectra with a typical low IF down-conversion is shown in Figure 2.34, using illustrative frequencies to give a feel for the nature of the signals with which we are dealing.

It can be seen that both the wanted signal and the image signal are present in the baseband, placing the image signal very close in frequency to the wanted signal (because

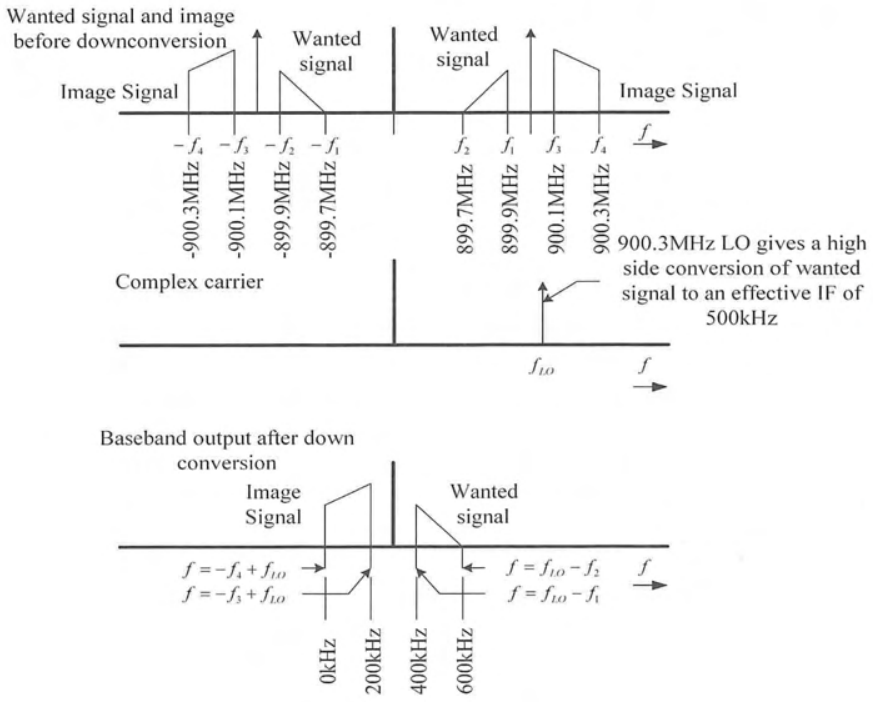


**Figure 2.34** Low IF mixing

the IF frequency is small by definition in a low IF architecture), making the image difficult to remove using conventional filtering. To remove the image requires image reject mixing to be performed. Image reject requirements for a low IF stage are not as stringent as those for a conventional superheterodyne stage, nor are they as low as the zero IF stage.

Consider a final IF of 500 kHz. The blocking specifications for the GSM air interface standard (see Figure 2.43) would permit a blocking signal at a separation of twice the IF frequency (1 MHz) to be at a level of − 33 dBm. If a calculation like that in Section 2.2.3.3 is performed; it can be shown that the required image rejection is 78 dB. The lower the IF, the lower will be the required image rejection performance.

### 2.4.4.1 Complex Filtering

An alternative to image reject mixing is use of a complex filter. A conventional filter, operating on real signals, is only capable of realizing complex poles in pairs [45,46]. If the signal is complex, however, then a filter that contains a single complex pole may be realized. Figure 2.35 shows such a filter.

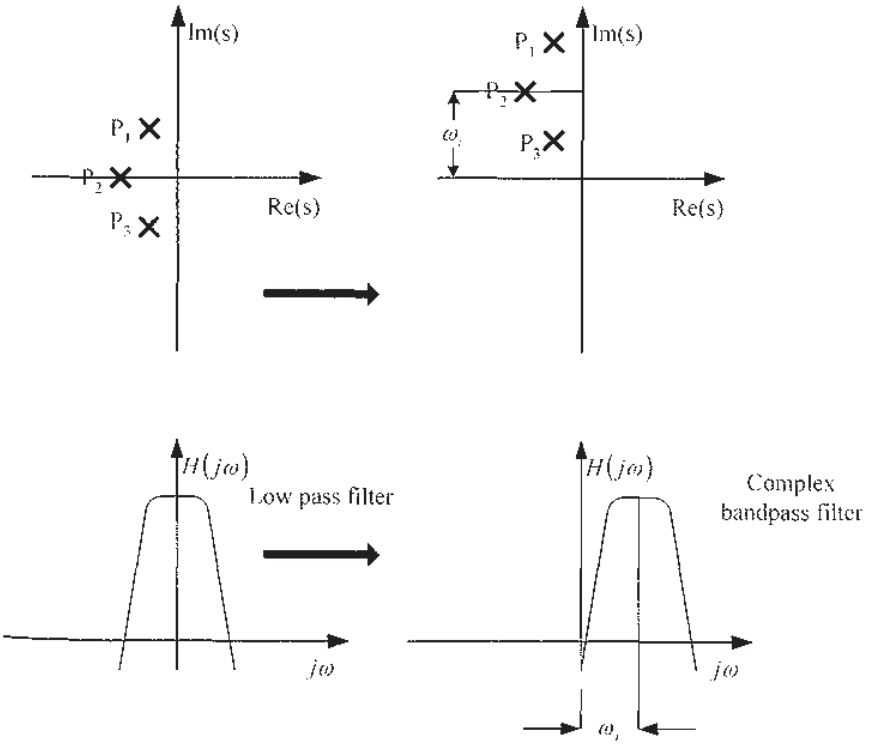


**Figure 2.35** S plane and frequency response representation of the translation of the poles of a lowpass to a bandpass filter that responds only to positive frequencies

In this filter the translation of the low pass poles, to the band pass poles, is effected by the transformation listed in Equation (13):

$$H_{BP}(s) = H_{LP}(s - j\omega_i) \tag{13}$$

where $H_{BP}(s)$ is the frequency response of the band pass filter, $H_{LP}(s)$ is the frequency response of the low pass filter and $\omega_i$ is the frequency offset of the band pass poles. If the transfer function of a low pass filter is described by the equation

$$H_{LP}(s) = \frac{1}{1 + \frac{s}{\omega_0}} \tag{14}$$

which has a pole at $-\omega_0$, the transfer function of the corresponding band pass filter is given by

$$H_{BP}(s) = \frac{1}{1 + \frac{s - j\omega_i}{\omega_0}} \tag{15}$$

This transfer function has a pole at

$$s = -\omega_0 + j\omega_i \tag{16}$$

So the pole is shifted up the imaginary axis by a distance of $\omega_i$.

The transfer function of Equation (15) can be rewritten as

$$H_{BP}(s) = \frac{1}{1 - 2jQ + s/\omega_0} \tag{17}$$

where

$$Q = \frac{\omega_i}{2\omega_0}$$

This type of operation can be performed the circuit shown in Figure 2.36. The output of the summing node of this filter is given by

$$z(t) = x(t) + (-1 + 2jQ)y(t) \tag{18}$$

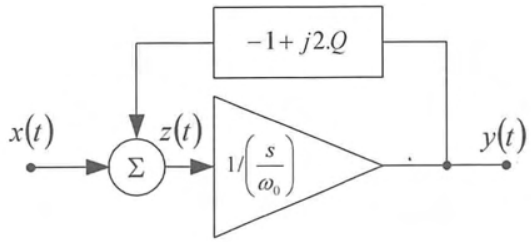


**Figure 2.36** Basic complex filter

Therefore the output of the filter can be shown to be

$$y(t) = \frac{1}{s/\omega_0} z(t) = \frac{1}{s/\omega_0}(x(t) + (-1 + 2jQ)y(t))$$

$$y(t)(s/\omega_0 - (-1 + 2jQ)) = x(t)$$

$$H(t) = \frac{y(t)}{x(t)} = \frac{1}{1 - 2jQ + s/\omega_0} \tag{19}$$

which corresponds with Equation (17).

The circuit that will allow complete realisation of this filter is shown in Figure 2.37. $x_R(t)$ and $x_I(t)$ represent the real and imaginary parts of the input signal. $y_R(t)$ and $y_I(t)$ represent the real and imaginary parts of the output signal. Using this filter, either the positive or negative frequency components of the filter may be selected. Residual response at the undesired sideband will be determined by the accuracy and balance of the filter components and the accuracy of the quadrature local oscillator.

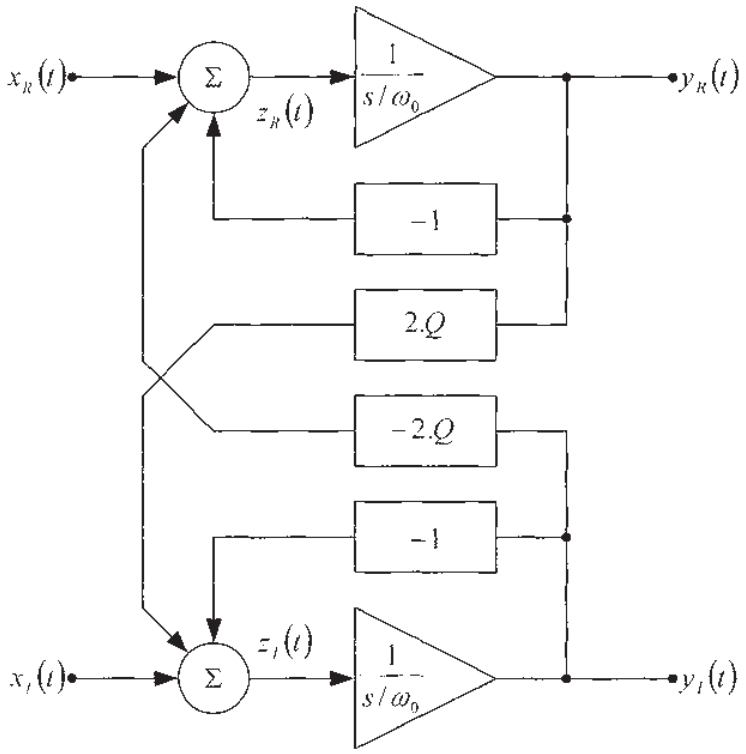


**Figure 2.37** Complete realization of a complex filter

# 6

# The Digital Front End – Bridge Between RF and Baseband Processing

Tim Hentschel and Gerhard Fettweis

*Technische Universität Dresden*

## 6.1 Introduction

### *6.1.1 The Front End of a Digital Transceiver*

The first question that might arise is: What is the digital front end? The notion of the digital front end (DFE) has been introduced by the author in several publications (e.g. [13]). Nonetheless it is useful to introduce the concept of the DFE at the beginning of this chapter.

Several candidate receiver and transmitter schemes have been presented by Beach et al. in Chapter 2. They all have in common that they are different from the so-called ideal software radio insofar as the signal has to undergo some signal processing steps before the baseband processing is performed on a software programmable digital signal processor (DSP). These signal processing stages between antenna and DSP can be grouped and called the *front end* of the transceiver.

Historically, the notion of a front end was applied to the very part of a receiver that was mounted at or near the antenna. It delivered a signal at an intermediate frequency which was carried along a wire to the back end. The back end was possibly placed apart from the antenna. In the current context the notion of the front end has been undermined a bit and moreover extended to the transmitter part of a transceiver. The functionality of the front end can be derived from the characteristics of the signals at its input and output. Figure 6.1 shows the front end located between the antenna and baseband processing part of a digital receiver.

Its input is fed with an analog wide-band signal comprising several channels of different services (air interfaces). There are $N_i$ channels of bandwidth $B_i$ of the $i$th service (air interface). Integrating over all services $i$ yields the total bandwidth $B$ of the wide-band signal. It includes the channel-of-interest that is assumed to be centered at $f_c$.

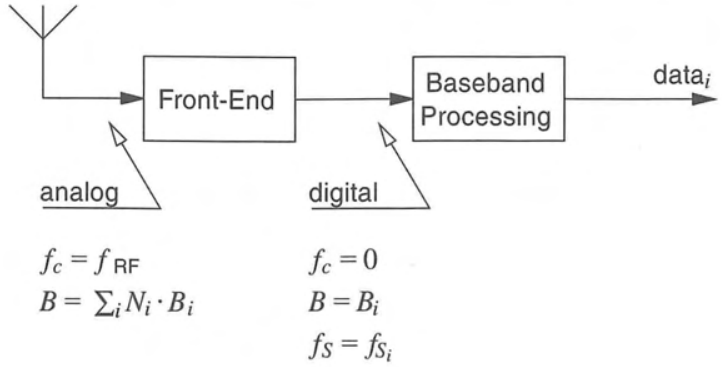


**Figure 6.1** A digital receiver

The output of the front end must deliver a digital signal (ready for baseband processing) with a sample rate determined by the current air interface. This digital signal represents the channel-of-interest of bandwidth $B_i$ centered at $f_c = 0$. Thus, the front end of a digital receiver must provide a *digital* signal

- of a certain *bandwidth*,
- at a certain *center frequency*, and
- with a certain *sample rate*.

Hence, the functionalities of the front end of a receiver can be derived from the four emphasized words as:

- channelization,
  - down-conversion of the channel-of-interest from RF to baseband, and
  - filtering (removal of adjacent channel interferers and possibly matched filtering),
- digitization,
- sample-rate conversion, and
- (synchronization).

Should synchronization belong to the front end or not? If the front end is equivalent to what Meyr et al. [19] call the *inner receiver*, synchronization is part of the front end. Synchronization basically requires two tasks: the estimation of errors (timing, frequency, and phase) induced by the channel, and their correction. The latter can principally be realized with the same algorithms and building blocks as the channelization and sample-rate conversion. The estimation of the errors is extra. In the current context these estimation algorithms should not be regarded as part of the front end. The emphasis lies on channelization, digitization, and sample-rate conversion.

Having identified the front end functionalities, the next step is to implement them. The question arises of where channelization should be implemented, in the analog or digital domain. As the different architectures in Chapter 2 suggest, some parts of channelization can be realized in the analog domain and other parts in the digital domain. This leads to

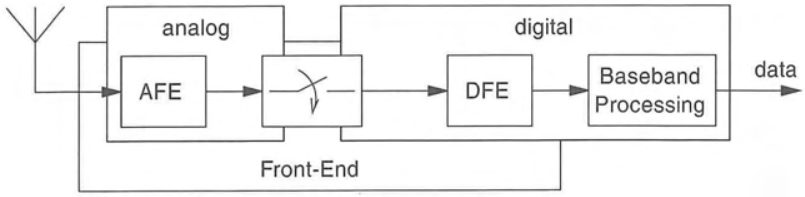


**Figure 6.2** The front end of a digital receiver

distinguishing the *analog front end* (AFE) and the *digital front end* (DFE) as shown in Figure 6.2. Thus, the digital front end is part of the front end. It performs front end functionalities digitally. Together with the analog-to-digital converter it bridges the analog RF- and IF-processing on one side and the digital baseband processing on the other side.

The same considerations that exist for the receiver are valid for the transmitter of a software defined transceiver. In the following, the receiver will be dealt with in most cases. Only where the transmitter needs special attention will it be mentioned explicitly.

In order to support the idea of software radio, the analog-to-digital interface should be placed as near to the antenna as possible thus minimizing the AFE. However, this means that the main channelization parts are performed in the digital domain. Therefore the signal at the input to the analog-to-digital converter is a wide-band signal comprising several channels, i.e. the channel-of-interest and several adjacent channel interferers as indicated by the bandwidth $B$ in Figure 6.1. On the transmitter side the spurious emission requirements must be met by the digital signal processing and the digital-to-analog converter. Hence, the signal characteristics are an important issue.

### *6.1.2 Signal Characteristics*

Signal characteristics means what the DFE must cope with (in the receiver) and what it must fulfill (in the transmitter). This is usually fixed in the standards of the different air interfaces. These standards describe, e.g. the maximum allowed power of adjacent channel interferers and blockers at the input of a receiver. From these figures the maximum dynamic range of a wide-band signal at the input to a software radio receiver can be derived. These specifications for the major European mobile standards are given in the Appendix to Chapter 2.

The maximum allowed power of adjacent channels increases with the relative distance between the adjacent channel and the channel-of-interest. Therefore, the dynamic range of a wide-band signal grows as the number of channels that the signal comprises increases. In order to limit the dynamic range, the bandwidth of the wide-band signal must be limited. This is done in the AFE. By this means the dynamic range can be matched to what the analog-to-digital converter can cope with. Assuming a fixed filter in the AFE, the total number of channels inside the wide-band signal depends on the channel bandwidth. This is sketched in Figure 6.3 for the air interfaces, UMTS (universal mobile telecommunications system), IS-95, and GSM (global system for mobile communications), assuming a total bandwidth of $B = 5$ MHz, and where $\delta$ stands for the minimum required signal-to-noise ratio of the channel-of-interest which is assumed to be similar for the three air interfaces.

Obviously, a trade-off between total dynamic range and channel bandwidth can be made.

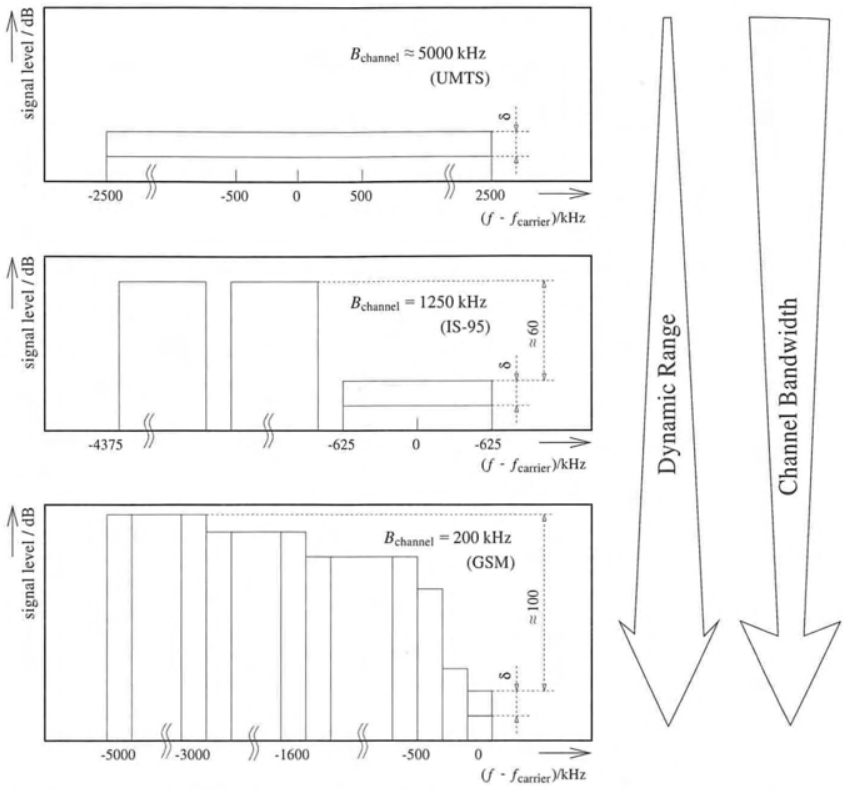


**Figure 6.3** Signal characteristics and the bandwidth dynamic range trade-off (adapted from [13], © 1999 IEEE)

The smaller the channel bandwidth is, the larger is the number of channels inside a fixed bandwidth and thus, the larger is the dynamic range of the wide-band signal. This trade-off has been named the *bandwidth dynamic range trade-off* [13]. It is important to note that only the channel-of-interest is to be received. This means that the possibly high dynamic range is required for the channel-of-interest only. Distortions, e.g. quantization noise of an analog-to-digital converter, must be limited or avoided only in the channel-of-interest. This property can be exploited in the DFE resulting in reduced effort, e.g.

1. the noise shaping characteristics of sigma-delta analog-to-digital converters fit this requirement perfectly [11],
2. filters can be realized as comb filters with low complexity (this is dealt with in Sections 6.4.1 and 6.5.4).

On the transmitter side the signal characteristics are not as problematic as on the receiver side. Waveforms and spurious emissions are usually provided in the standards. These figures must be met, influencing the necessary processing power, the word length, and thus the power

consumption. However, a critical part is the wide-band AFE of the transmitter. Since there is no analog narrow-band filter matched to the channel bandwidth, the linearity of the building blocks, e.g. the power-amplifier, is a crucial figure.

### *6.1.3 Implementation Issues*

In order to implement as many functionalities as possible in the digital domain and thus provide a means for adapting the radio to different air interfaces, the sample rates at the analog/digital interface are chosen very high. In fact, they are chosen as high as the ADC and DAC allow. The algorithms realizing the functionalities of the DFE must be performed at these high sample rates. As an example, digital down-conversion should be mentioned. As can be seen in Section 6.3, a digital image rejection mixer requires four real multiplications per complex signal sample. Assuming a sample rate of 100 million samples per second (MSps) this yields a multiplication rate of 400 million multiplications per second. This would occupy a good deal of the processing power of a DSP, however, without really requiring its flexiblity. Therefore it is not sensible to realize digital down-conversion on a digital signal processor. The same consideration also holds in principle for channelization and sample-rate conversion: very high sample rates in connection with signals of high dynamic range makes the application of digital signal processors questionable. If, moreover, the signal processing algorithms do not require much flexiblity from the underlying hardware platform it is not sensible to use a DSP.

A solution to this problem is parameterizable and reconfigurable hardware. Reconfigurable hardware is hardware whose building blocks can be reconfigured on demand. Field programmable gate arrays (FPGAs) belong to this class. Up to now these FPGAs have a long reconfiguration time compared to the processing speed they offer. Therefore they cannot be reconfigured dynamically, i.e. while processing. On the other hand, the application in mobile communications systems is well defined. There is a limited number of algorithms that must be realized. For that reason hardware structures have been developed that are not as fine-grained as FPGAs. This means that the building blocks are not as general as in FPGAs but are much more tailored to the application. This results in reduced effort.

If the granularity of the hardware platform is made even more coarse, the hardware is no longer reconfigurable but parameterizable. Dedicated building blocks whose functionality is fixed can be implemented on application specific integrated circuits (ASICs) very efficiently. If the main parameters are tunable, these ASICs can be employed in software defined radio transceivers. A simple example is the above-mentioned digital down-conversion. The only thing that must be tunable is the frequency of the local oscillator. Besides this, the complete underlying hardware does not need to be changed. This is very efficient as long as digital down-conversion is required. In a potential operation mode not requiring digital down-conversion of a software radio, the dedicated hardware block cannot be used and must be regarded as ballast.

However, with respect to the wide-band signal at the output of the analog-to-digital converter in a digital receiver, it is sensible to assume that the functionalities of the DFE, namely channelization and sample-rate conversion, are necessary for most air interfaces. Hence, the idea of dedicated parameterizable hardware blocks promises to be an efficient solution. Therefore, all considerations and investigations in this chapter are made with respect to an implementation as reconfigurable hardware.

Hardware and implementation issues are covered in detail in subsequent chapters.

## 6.2 The Digital Front End

### *6.2.1 Functionalities of the Digital Front End*

From the previous section it can be concluded that the functionalities of the DFE in a receiver are

- channelization (i.e. down-conversion and filtering), and
- sample-rate conversion.

The functionalities of a receiver DFE are illustrated in Figure 6.4. It should be noted that the order of the three building blocks (digital down-conversion, SRC, and filtering) is not necessarily as shown in Figure 6.4. This will become clear in the course of the chapter.

Since the DFE should take over as many tasks as possible from the AFE in a software radio, the functionalities of the DFE are very similar to what has been described in Section 6.1.1 for

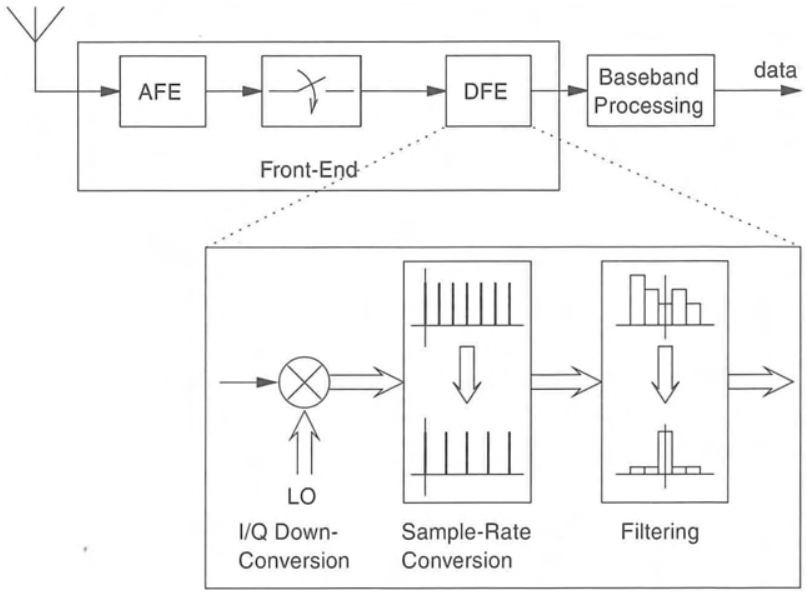


**Figure 6.4** A digital receiver with a digital front end

the front end in general. The digitized wide-band signal comprises several channels among which the channel-of-interest is centered at an arbitrary carrier frequency. Channelization is the functionality that shifts the channel-of-interest to baseband and moreover removes all adjacent channel interferers by means of digital filtering.

Sample rate conversion (SRC) is a relatively 'young' functionality in a digital receiver. In conventional digital receivers the analog/digital interface has been clocked with a fixed rate derived from the master clock rate of the air interface that the transceiver was designed for. In

software radio transceivers there is no isolated target air interface. Therefore the transceiver must cope with different master clock rates. Moreover, it must be borne in mind that the terminal and base station run mutually asynchronously and must be synchronized when the connection is set up.

There are two approaches to overcome these two problems. First, the analog/digital interface can be clocked with a tunable clock. Thus, for all air interfaces the right sampling clock can be used. Additionally, it is possible to 'pull' the tunable oscillator for synchronization purposes. It is clear that such a tunable oscillator requires considerably more effort than a fixed one. For that reason designers favour the application of a fixed oscillator. Nonetheless, the baseband processing requires a signal with a proper sample rate. Hence, sample-rate conversion is necessary in this case for converting between the fixed clock rate at the analog/digital interface and the target rate of the respective air interface.

Very often interpolation (e.g. Lagrange interpolation) is regarded as a solution to SRC. Still, this solution is only sensible in certain applications. The usefulness of conventional interpolation depends on the signal characteristics. In Section 6.1.1, it has been mentioned that the wide-band signal at the input of the DFE of a receiver can comprise several channels beside the channel-of-interest. However, only the channel-of-interest is really wanted. This fact can be exploited for reducing the effort for SRC (see Section 6.5).

Since both channelization and SRC require filtering, it is possible to combine them. This can lead to considerable savings. A well-known example is multirate filtering [1]. This is a concept where filtering and integer factor SRC (e.g. decimation) are realized stepwise on a cascaded structure comprising several stages of filtering and integer factor SRC. Generally, this results in both a lower multiplication rate and a lower hardware complexity.

The functionalities of the transmitter part of a DFE are equivalent to those of the receiver part: the baseband signal to be transmitted is filtered, digitally up-converted, and its sample rate is matched to the sample rate of the analog/digital interface. Although there are no adjacent channels to be removed, filtering is necessary for symbol forming and in order to fulfill the spurious emissions characteristics dictated by the respective standard. Again, filtering and SRC can be combined.

There is a strong relationship between digital down-conversion and channel filtering since they form the functionality channelization. On the other hand, it has been mentioned that there is also a strong relationship between channel filtering and SRC, e.g. in the case of multirate filtering. In the main part of this chapter, a separate section is dedicated to each of the three, digital down-conversion, channel filtering, and sample-rate conversion. Important relations between them are dealt with in these sections.

### 6.2.2 *The Digital Front End in Mobile Terminals and Base Stations*

The great issue of mobile terminals is power consumption. Everything else is less important. Power consumption is the *alpha* and the *omega* of mobile terminal design. On the other hand, mobile terminals usually must only process one channel at a time. This fact enables the application of efficient solutions for channelization and SRC that are based on the multirate filtering concept.

In contrast to this there are no restrictions regarding power consumption in base stations besides basic environmental aspects. Still, in base stations several channels must be processed in parallel.

This fundamental difference between mobile terminals and base stations must be kept in mind when investigating and evaluating algorithms and potential solutions.

## 6.3 Digital Up- and Down-Conversion

### *6.3.1 Initial Thoughts*

The notion of up- and down-conversion stands for a shift of a signal towards higher or lower frequencies, respectively. This can be achieved by multiplying the signal $\underline{x}_a(t)$ with a complex rotating phasor which results in

$$\underline{x}_b(t) = \underline{x}_a(t)e^{j2\pi f_c t} \tag{1}$$

where $f_c$ stands for the frequency shift. Often $f_c$ is called the carrier frequency to which a baseband signal is up-converted, or from which a band-pass signal is down-converted. However, in this case $f_c$ would have to be positive. Regarding it as a frequency shift enables us to use positive and negative values for $f_c$.

The real and imaginary parts of a complex signal are also called the in-phase and the quadrature-phase components, respectively.

Digital up- and down-conversion is the digital equivalent of Equation (1). This means that both the signals and the complex phasor are represented by quantized samples (quantization issues are not covered in this chapter). Introducing a sampling period $T$, that fulfills the sampling theorem, digital up- and down-conversion can be written as

$$\underline{x}_b(kT) = \underline{x}_a(kT)e^{j2\pi f_c kT} \tag{2}$$

Assuming perfect analog-to-digital or digital-to-analog conversion, respectively, Equations (1) and (2) are equivalent.

Depending on the sign of $f_c$, up- or down-conversion results. Thus, it is sufficient to deal with one of the two. Only digital down-conversion is discussed in the sequel.

It should be noted that real up- and down-conversion is also possible and indeed very common, i.e. multiplying the signal with a sine or cosine function instead of the complex exponential of Equations (1) and (2). However, real up- and down-conversion is a special case of complex up- and down-conversion and is therefore not discussed separately in this chapter.

### *6.3.2 Theoretical Aspects*

In order to understand the task of digital down-conversion, it is useful to consider the complete signal processing chain of up-conversion in the transmitter, transmission, and final down-conversion in the receiver. It is assumed that the received signal is down-converted twice. First the complete receive band is down-converted in the AFE. This is followed by filtering. The processed signal is again down-converted in the DFE. This is sketched in Figure 6.5.

For the discussion it is assumed that there are no distortions due to the channel, however, it introduces adjacent channel interferers. Thus, the received signal $x_{Rx}(t)$ is equal to the transmitted signal $x_{Tx}(t)$ plus adjacent channel interferers $a(t)$:

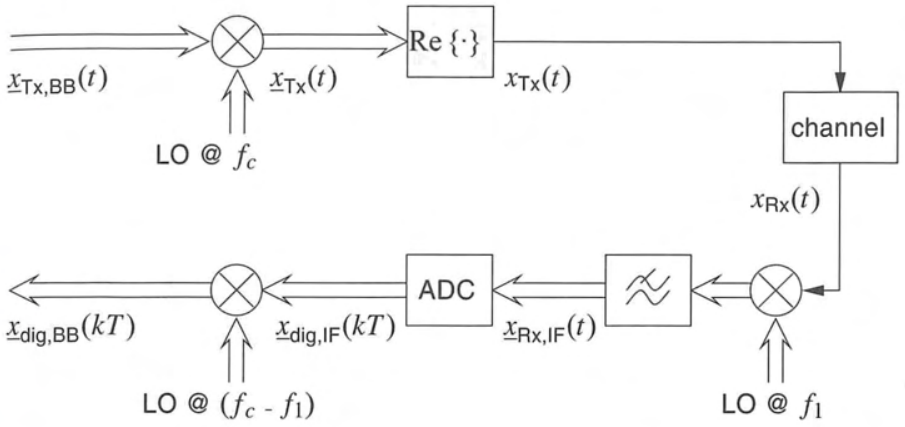


**Figure 6.5** The signal processing chain of up-conversion, transmission, and final down-conversion of a signal (LO stands for local oscillator)

$$x_{\mathrm{Rx}}(t) = x_{\mathrm{Tx}}(t) + a(t)$$

$$= \Re\mathrm{e}\left\{\underline{x}_{\mathrm{Tx,BB}}(t)e^{j2\pi f_c t}\right\} + a(t) \tag{3}$$

$$= \frac{1}{2}\left(\underline{x}_{\mathrm{Tx,BB}}(t)e^{j2\pi f_c t} + \underline{x}^*_{\mathrm{Tx,BB}}(t)e^{-j2\pi f_c t}\right) + a(t) \tag{4}$$

where $\underline{x}_{\mathrm{Tx,BB}}(t)$ is the complex baseband signal to be transmitted. $f_c$ denotes the carrier frequency and $\underline{x}^*$ the conjugate complex of $\underline{x}$. From Equation (4) it can be concluded that the received signal comprises two components besides the adjacent channel interferers: one centered at $f_c$ and another centered at $-f_c$. The first comprises the signal-of-interest $\underline{x}_{\mathrm{Tx,BB}}(t)$. It lies anywhere in the frequency band of bandwidth $B$ which comprises several frequency divided channels, i.e. the channel-of-interest plus adjacent channel interferers. This band is selected by a receive band-pass filter. The arrangement of the channel-of-interest (i.e. the signal $x_{\mathrm{Rx}}(t)$) in the receive frequency band is sketched in Figure 6.6.

As mentioned above the analog front end performs down-conversion of the complete receive frequency band of bandwidth $B$. Inside this frequency band lies the signal-of-interest $\underline{x}_{\mathrm{Tx,BB}}(t)$ which should finally be down-converted to baseband. The following signal is produced at the output of the analog down-converter when down-converting by $f_1$. For reasons of simplicity of the derivation we shall limit $f_1$ to $f_1 < f_c$.

$$\underline{x}_{\mathrm{Rx,IF}}(t) = x_{\mathrm{Rx}}(t)e^{-j2\pi f_1 t} \tag{5}$$

$$= \left(\frac{1}{2}\underline{x}_{\mathrm{Tx,BB}}(t)e^{j2\pi(f_c - f_1)t} + \underline{x}^*_{\mathrm{Tx,BB}}(t)e^{-j2\pi(f_c + f_1)t}\right) + a_{\mathrm{filt}}(t)e^{-j2\pi f_1 t} \tag{6}$$

where $a_{\mathrm{filt}}(t)$ denotes all adjacent channel interferers inside the receive bandwidth $B$. The interesting signal component is centered at the intermediate frequency (IF)

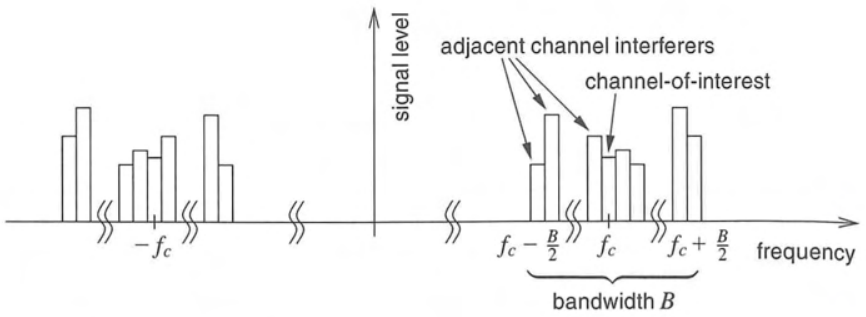


**Figure 6.6** Position of the channel-of-interest in the receive frequency band of bandwidth $B$

$$f_{IF} = f_c - f_1 \tag{7}$$

It is enclosed by several adjacent channel interferers. A second signal component lies $2f_c$ apart from the first (sketched in Figure 6.7).

The latter is of no interest; moreover, it can cause aliasing in the analog-to-digital conversion process. Therefore it is removed by low-pass (or band-pass) filtering. Thus, the digitized signal is:

$$\underline{x}_{\text{dig,IF}}(kT) = \frac{1}{2}\underline{x}_{\text{Tx,BB}}(kT)e^{j2\pi f_{IF}kT} + \underline{a}_{\text{dig}}(kT) \tag{8}$$

where $\underline{a}_{\text{dig}}(kT)$ stands for the remaining adjacent channels after down-conversion, anti-aliasing filtering, and digitization. $T$ is the sampling period that must be small enough to fulfill the sampling theorem. In general the digital IF signal is a complex signal; the interesting signal component is centered at $f_{IF}$.

The objective of digital down-conversion is to shift this interesting component from the carrier frequency $f_{IF}$ down to baseband. By inspection of Equation (8) it can be found that down-conversion can be achieved by multiplying the received signal with a respective exponential function:

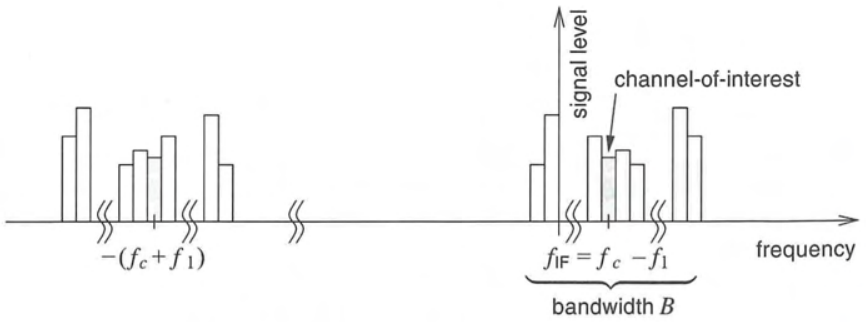


**Figure 6.7** Position of the channel-of-interest at IF

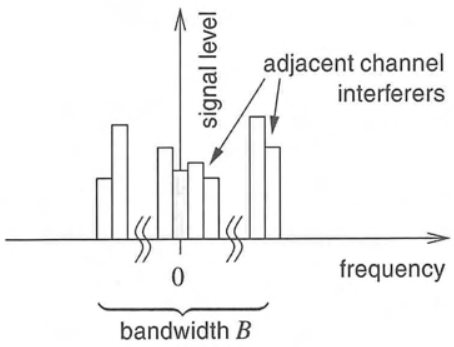


**Figure 6.8** Channel-of-interest at baseband (result of low-pass filtering of the signal of Figure 6.7 followed by digital down-conversion)

$$\underline{x}_{\mathrm{dig,BB}}(kT) = \underline{x}_{\mathrm{dig,IF}}(kT)e^{-j2\pi f_{\mathrm{IF}}kT} \tag{9}$$

$$= \frac{1}{2}\underline{x}_{\mathrm{Tx,BB}}(kT) + \underline{a}_{\mathrm{dig}}(kT)e^{-j2\pi f_{\mathrm{IF}}kT} \tag{10}$$

This yields a sampled version of the transmitted signal $\underline{x}_{\mathrm{Tx,BB}}(t)$ scaled with a factor 1/2. It is sketched in Figure 6.8. The adjacent channel interferers can be removed with a channelization filter (see Section 6.4).

It should be noted that in reality the oscillators of transmitter and receiver are not synchronized. Therefore, down-conversion in the receiver yields a signal with phase offset and frequency offset that must be corrected. The aim of the derivation in this section was to show what happens with the signal in principle in the individual processing stages and not to discuss all possible imperfections.

### *6.3.3 Implementation Aspects*

In practical applications it is necessary to treat the real- and imaginary part of a complex signal separately as two individual real signals. Thus, the signal after analog down-conversion comprises the following two components:

$$\Re\mathrm{e}\{\underline{x}_{\mathrm{Rx,IF}}(t)\} = \Re\mathrm{e}\left\{x_{\mathrm{Rx}}(t)e^{-j2\pi f_1 t}\right\}$$

$$= x_{\mathrm{Rx}}(t)\cos(2\pi f_1 t) \tag{11}$$

$$\Im\mathrm{m}\{\underline{x}_{\mathrm{Rx,IF}}(t)\} = \Im\mathrm{m}\left\{x_{\mathrm{Rx}}(t)e^{-j2\pi f_1 t}\right\}$$

$$= -x_{\mathrm{Rx}}(t)\sin(2\pi f_1 t) \tag{12}$$

It can be concluded that analog down-conversion can be implemented by means of multiplying the received real signal by a cosine signal and a sine signal. The real part of the complex IF signal (also called the in-phase component) is obtained by multiplying the

received signal with a cosine signal; the imaginary part of the complex IF signal (also called the quadrature-phase component) is obtained by multiplying the received signal with a sine signal.

From Equation (8) it can be concluded that the input signal to the digital down-converter is in principle a complex signal. Hence, the digital down-conversion described by Equation (9) requires a complex multiplication. Since the complex signals are only available in the form of their real and imaginary parts, the complex multiplication of the digital down-conversion requires four real multiplications. By separating the real and imaginary parts of Equation (9), we have

$$\Re\mathrm{e}\left\{\underline{x}_{\mathrm{dig,BB}}(kT)\right\} = \Re\mathrm{e}\left\{\underline{x}_{\mathrm{dig,IF}}(kT)\right\}\cos(2\pi f_{\mathrm{IF}}kT) + \Im\mathrm{m}\left\{\underline{x}_{\mathrm{dig,IF}}(kT)\right\}\sin(2\pi f_{\mathrm{IF}}kT) \tag{13}$$

$$\Im\mathrm{m}\left\{\underline{x}_{\mathrm{dig,BB}}(kT)\right\} = \Im\mathrm{m}\left\{\underline{x}_{\mathrm{dig,IF}}(kT)\right\}\cos(2\pi f_{\mathrm{IF}}kT) - \Re\mathrm{e}\left\{\underline{x}_{\mathrm{dig,IF}}(kT)\right\}\sin(2\pi f_{\mathrm{IF}}kT) \tag{14}$$

This can be regarded as a direct implementation of digital down-conversion. It is sketched in Figure 6.9.

There are two special cases:

1. When the signal $\underline{x}_{\mathrm{dig,IF}}(kT)$ is real, it is $\Im\mathrm{m}\left\{\underline{x}_{\mathrm{dig,IF}}(kT)\right\} = 0$. Hence, digital down-conversion can be realized by means of two real multiplications in this case.
2. When applying the above results to up-conversion, it is often sufficient to keep the real part

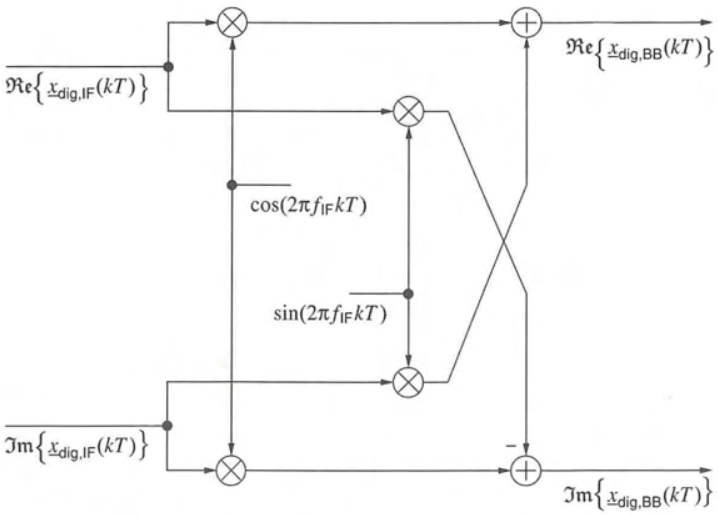


**Figure 6.9** Direct realization of digital down-conversion

of the up-converted signal. Thus, only Equation (13) must be solved resulting in an effort of two real multiplications and one addition per signal sample.

The samples of the discrete-time cosine and sine functions in Figure 6.9 are usually stored in a look-up table. The ROM table can simply be addressed by the output signal of an overflowing phase accumulator representing the linearly rising argument $(2\pi f_{IF}kT)$ of the cosine and sine functions. Requiring a resolution of $n$ bits, the look-up table has a size of approximately $2^n \times n$ bits which together with the four general purpose multipliers results in large chip area, high power consumption, and considerable costs [18].

The large look-up table can be avoided by generating the samples of the digital sine and cosine functions with an infinite length impulse response (IIR) oscillator. It is an IIR filter with a transfer function that has a complex or conjugate complex pole on the unit circle [5]. Another way to generate the sine and cosine samples without the need for a large look-up table is the CORDIC algorithm (CORDIC stands for COordinate Rotation Digital Computer). The great advantage of the CORDIC algorithm is that it not only substitutes the large look-up table but also the required four multipliers. This is possible since the CORDIC algorithm can be used to perform a rotation of the complex phase of a complex number. Interpreting the samples of the complex signal $\underline{x}_{dig,IF}(kT)$ as these complex numbers, and rotating the phase of these samples according to $(2\pi f_{IF}kT)$, the CORDIC algorithm directly performs the digital up- or down-conversion without the need for explicit multipliers.

### *6.3.4 The CORDIC Algorithm*

The CORDIC algorithm was developed by Volder [25] in 1959 for converting between cartesian and polar coordinates. It is an iterative algorithm that solely requires shift, add, and subtract operations. In the circular rotation mode, the CORDIC calculates the cartesian coordinates of a vector which is rotated by an arbitrary angle.

To rotate the vector

$$\underline{v}_0 = e^{j\phi} \tag{15}$$

by an angle $\Delta\phi$, $\underline{v}_0$ is multiplied by the corresponding complex rotating phasor

$$\underline{v} = \underline{v}_0 \cdot e^{j\Delta\phi} \tag{16}$$

The real and imaginary parts of $\underline{v}$ are calculated individually:

$$\Re e\{\underline{v}\} = \Re e\{\underline{v}_0\}\cos(\Delta\phi) - \Im m\{\underline{v}_0\}\sin(\Delta\phi) \tag{17}$$

$$\Im m\{\underline{v}\} = \Im m\{\underline{v}_0\}\cos(\Delta\phi) + \Re e\{\underline{v}_0\}\sin(\Delta\phi) \tag{18}$$

Rearranging yields

$$\frac{\Re e\{\underline{v}\}}{\cos(\Delta\phi)} = \Re e\{\underline{v}_0\} - \Im m\{\underline{v}_0\}\tan(\Delta\phi); \quad |\Delta\phi| \notin \left\{\frac{1}{2}\pi, \frac{3}{2}\pi, \ldots\right\} \tag{19}$$

$$\frac{\Im m\{\underline{v}\}}{\cos(\Delta\phi)} = \Im m\{\underline{v}_0\} + \Re e\{\underline{v}_0\}\tan(\Delta\phi); \quad |\Delta\phi| \notin \left\{\frac{1}{2}\pi, \frac{3}{2}\pi, \ldots\right\} \tag{20}$$

Note that only the tangent of the angle $\Delta\phi$ must be known to achieve the desired rotation. The rotated vector is scaled by the factor $1/\cos(\Delta\phi)$.

For many applications it is too costly to realize the two multiplications of Equations (19) and (20). The idea of the CORDIC algorithm is to perform the desired rotation by means of elementary rotations of decreasing size, thus iteratively approaching the exact rotation by $\Delta\phi$. By choosing the elementary rotation angles as $\tan(\Delta\phi_i) = \pm 1/2^i$, the multiplications of Equations (19) and (20) can be replaced by simple shift operations.

$$\Delta\phi_i = \pm \arctan\left(2^{-i}\right); \quad i = 0, 1, 2, \ldots \tag{21}$$

Consequently, in order to rotate a vector $\underline{v}_0$ by an angle $\Delta\phi = z_0$ with $|\Delta\phi| < \pi/2$, the CORDIC algorithm performs a sequence of successively decreasing elementary rotations with the basic rotation angles $\Delta\phi_i = \pm \arctan(2^{-i})$ for $i = 0, 1, \ldots, n-1$. The limitation of $\Delta\phi$ is necessary to ensure uniqueness of the elementary rotation angles. Finally, the iterative process yields the cartesian coordinates of the rotated vector $\underline{v}_n \approx \underline{v}$. The resulting iterative process can be described by the following equations for $i = 0, 1, \ldots, n-1$:

$$x_{i+1} = x_i - d_i y_i 2^{-i} \tag{22}$$

$$y_{i+1} = y_i + d_i x_i 2^{-i} \tag{23}$$

$$z_{i+1} = z_i - d_i \arctan(2^{-i}) \tag{24}$$

where

$$x_0 = \Re\mathrm{e}\{\underline{v}_0\} \tag{25}$$

$$y_0 = \Im\mathrm{m}\{\underline{v}_0\} \tag{26}$$

$$x_n = \Re\mathrm{e}\{\underline{v}_n\} \tag{27}$$

$$y_n = \Im\mathrm{m}\{\underline{v}_n\} \tag{28}$$

The figure

$$d_i = \begin{cases} -1 & \text{if } z_i < 0 \\ +1 & \text{otherwise} \end{cases} \tag{29}$$

defines the direction of each elementary rotation. After $n$ iterations the CORDIC iteration results in

$$x_n \approx A_n[x_0 \cos(z_0) - y_0 \sin(z_0)] = \Re\mathrm{e}\left\{A_n \underline{v}_0 e^{jz_0}\right\} \tag{30}$$

$$y_n \approx A_n[y_0 \cos(z_0) + x_0 \sin(z_0)] = \Im\mathrm{m}\left\{A_n \underline{v}_0 e^{jz_0}\right\} \tag{31}$$

$$z_n \approx 0 \tag{32}$$

where

$$A_n = \prod_{i=0}^{n-1} \sqrt{1 + 2^{-2i}} \tag{33}$$

is the CORDIC scaling factor which depends on the total number of iterations. Hence, the result of the CORDIC iteration is a *scaled* version of the rotated vector.

In order to overcome the restriction regarding $|\Delta\phi|$ an initial rotation by $\pm\pi/2$ can be performed if necessary before starting the CORDIC iterations. For details see [15,25].

### 6.3.5 Digital Down-Conversion with the CORDIC Algorithm

Interpreting each complex sample of the signal $\underline{x}_{\text{dig,IF}}(kT)$ of Equation (8) as a complex number $\underline{v}_0$, and the angle $\Delta\phi(k) = -2\pi f_{\text{IF}} kT$ as $z_0$, the CORDIC can be used to continuously rotate the complex phase of the signal $\underline{x}_{\text{dig,IF}}(kT)$, thus performing digital down-conversion. Since the CORDIC is an iterative algorithm, it is necessary to implement each of the iterations by its own hardware stage if high-speed applications are the objective. In such pipelined architectures the invariant elementary rotation angles $\arctan(2^{-i})$ of Equation (24) can be hard-wired. The overall hardware effort of such an implementation of the CORDIC algorithm is approximately that of three multipliers with the respective word length. Hence one multiplier and the ROM look-up table of the conventional approach for down-conversion of Figure 6.9 can be saved with a CORDIC realization. The principle of digital down-conversion using the CORDIC algorithm is sketched in Figure 6.10.

For further details on digital down-conversion with the CORDIC the reader is referred to [18] where quantization error bounds and simulation results are given.

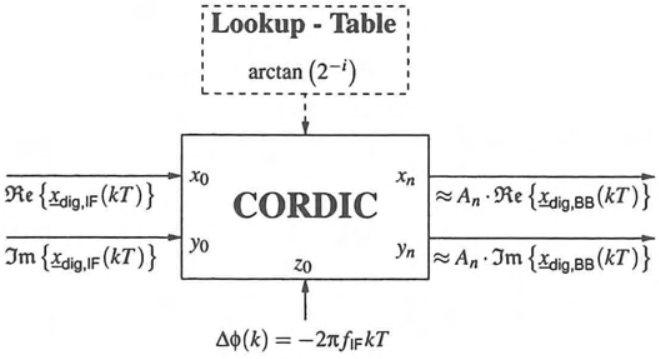


**Figure 6.10** Principle of digital down-conversion using the CORDIC algorithm

### 6.3.6 Digital Down-Conversion by Subsampling

The starting point is Equation (8):

$$\underline{x}_{\text{dig,IF}}(kT) = \frac{1}{2}\underline{x}_{\text{Tx,BB}}(kT)e^{j2\pi f_{\text{IF}} kT} + \underline{a}_{\text{dig}}(kT)$$

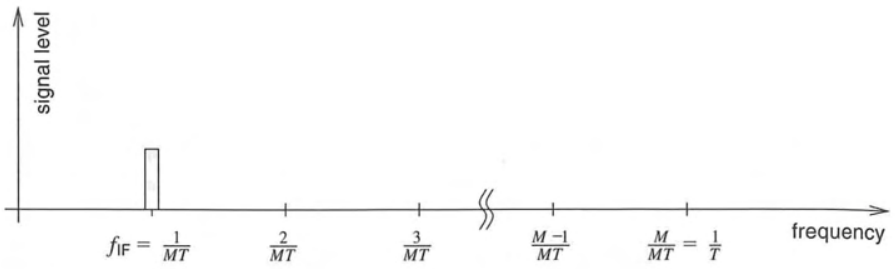


**Figure 6.11** Digitally filtered IF signal (filter bandwidth equals channel bandwidth)

It is assumed that $f_l$ has been chosen so that the channel-of-interest is located at a *fixed* intermediate frequency $f_{IF}$. The channel can be separated from all adjacent channels by means of complex band-pass filtering (see Section 6.4.2) at this frequency. Since the bandwidth of this band-pass filter must be variable in software radio applications, it can be a digital filter that processes the signal directly after digitization. Hence, it delivers the signal

$$\underline{x}_{\text{dig-filt,IF}}(kT) = \frac{1}{2}\underline{x}_{\text{Tx,BB}}(kT)e^{j2\pi f_{IF}kT} \tag{34}$$

that is sketched in Figure 6.11. At this stage it is assumed that the following relation holds:

$$f_{IF} = \frac{n}{M}\frac{1}{T}, \quad n = 1, 2, \ldots, M-1 \tag{35}$$

i.e. the intermediate frequency is an integer multiple of a certain fraction of the sample rate. This can easily be achieved since the IF is fixed in most practically relevant systems. As to the sample rate, the advantage of having a fixed rate has been discussed in Section 6.2.1. Thus, the ratio of Equation (35) is a parameter that can be specified once in the system design phase. Substituting Equation (35) in Equation (34) yields

$$\underline{x}_{\text{dig-filt,IF}}(kT) = \frac{1}{2}\underline{x}_{\text{Tx,BB}}(kT)e^{j2\pi(n/M)k} \tag{36}$$

Decimating (i.e. subsampling) the signal $\underline{x}_{\text{dig-filt,IF}}(kT)$ by $M$ eventually leads to

$$\underline{x}_{\text{dig-filt,IF}}(kMT) = \frac{1}{2}\underline{x}_{\text{Tx,BB}}(kMT)e^{j2\pi(nM/M)k} \tag{37}$$

$$= \frac{1}{2}\underline{x}_{\text{Tx,BB}}(kMT) \tag{38}$$

which is equivalent to the transmitted baseband signal scaled by 1/2 and with sampling period $MT$, supposing that the sampling period $MT$ is short enough to represent the signal, i.e. to fulfill the sampling theorem (see Figure 6.12). A structure for down-conversion by subsampling is sketched in Figure 6.13.

This process of digital down-conversion is called *harmonic subsampling* or *integer-band decimation* [1]. The equivalent for up-conversion is called *integer-band interpolation*. It is based on up-sampling (see Section 6.5) followed by band-pass filtering [1].

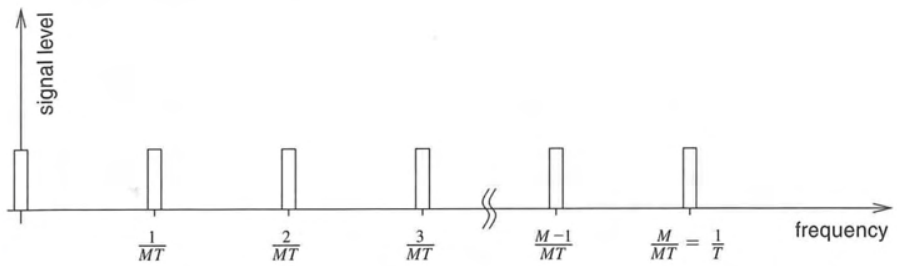


**Figure 6.12** Result of subsampling the signal of Figure 6.11

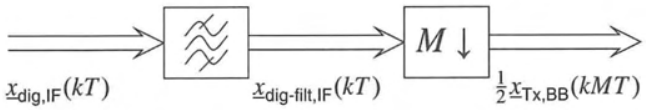


**Figure 6.13** Principal structure for integer-band decimation (digital down-conversion by subsampling)

Both methods, integer-band decimation and interpolation are pure sampling processes and thus, do not require any operation. Still, they do require band-pass filtering, before down-sampling in the case of down-conversion, and after up-sampling in the case of up-conversion, respectively. It is the functionality of channel filtering that must be properly combined with up- or down-sampling in order to have the up- or down-conversion effect. This is discussed in detail in Section 6.5.

## 6.4 Channel Filtering

### *6.4.1 Low-Pass Filtering after Digital Down-Conversion*

#### 6.4.1.1 Direct Approach

Figure 6.8 shows the principal channel arrangement in the frequency domain after digital down-conversion of the channel-of-interest to baseband. This is simply the result of shifting the right-hand side of Figure 6.7.

Besides the channel-of-interest there are many adjacent channels inside the receive frequency band of bandwidth $B$ that have been down-converted. In order to select the channel-of-interest these adjacent channels must be removed with a filter. Since the channel-of-interest has been down-converted to baseband, a low-pass filter is an appropriate choice.

Infinite length impulse response (IIR) filters are generally avoided due to the nonlinear phase characteristics which distort the signal. Of course there are cases, especially if the pass-band is very narrow, where the phase characteristics in the pass-band of the filter can be well controlled. Still, IIR filters with very narrow pass-band tend to suffer more from stability

problems than those with a wider pass-band. On the other hand IIR filters have very short group delay. For that reason they might be advantageous in certain applications.

The problems of IIR filters can be avoided when using linear phase filters with finite length impulse response (FIR). Their great drawback is the generally high order that is necessary to implement certain filter characteristics compared to the order of an IIR filter with equivalent performance. For details on digital filter design the reader is referred to the great amount of literature available in this field.

In order to get some idea of the effort for direct implementation of channel filtering, it is instructive to learn that for many types of FIR filters (including equiripple FIR filters, FIR filters based on window designs, and Chebychev FIR filters) the number of coefficients $K$ can be related to the transition bandwidth $\Delta f$ of the filter and the sample rate $f_S$ at which it operates. This proportionality is [1]

$$K \sim \frac{f_S}{\Delta f}; \quad \Delta f < f_S \tag{39}$$

The transition bandwidth $\Delta f$ is the difference between the cut-off frequency and the lower edge of the stop band. It can be expressed as a certain fraction of the channel bandwidth. Thus, it is obvious that the transition bandwidth gets very small compared to the sample rate $f_S$ if there is a large number of adjacent channels, i.e. the channel bandwidth itself is very small compared to $f_S$.

Besides the number of coefficients another figure increases with a large number of adjacent channel interferers: the dynamic range of the signal (see Section 6.1.2). In the case of wide-band reception of a GSM signal the dynamic range of the signal can easily reach 80 dBand more. In order to sufficiently attenuate all adjacent channels of such a signal, the processing word length of the digital filter must be relatively high. A large number of coefficients, a high coefficient and processing word length, and a high clock rate are indicators for high effort and costs that are required if the channel filtering functionality is directly implemented by means of a conventional FIR filter.

As the bandwidth of the digital signal is reduced by filtering there is no reason to keep the high sample rate that was necessary before filtering. As long as the sampling theorem is obeyed, the sample rate can be reduced. This results in lower processing rates and thus, lower effort. Therefore, the high sample rate is usually reduced down to the bit, chip or symbol rate of the signal after filtering (or a small integer multiple of it). Knowing about the sample rate reduction after the filtering, it is possible to reduce the filtering effort considerably by combining filtering and sample rate reduction. This approach is called multirate filtering.

#### 6.4.1.2 Multirate Filtering

The direct approach of implementing the channel filter is a low-pass filter (followed by a down-sampler). The down-sampler reduces the sample rate according to the bandwidth of the filtered signal. This is described in the previous section.

For the following discussion it is useful to regard the combination of the filter and the down-sampler as a *system for sample rate reduction* (see also Section 6.5). Down-sampling is a process of sampling. Therefore, it causes aliasing that can be avoided if the signal is sufficiently band-limited before down-sampling. This band limitation is achieved with

anti-aliasing filtering. The low-pass filter preceding the down-sampling process, i.e. the channel filter, acts as an anti-aliasing filter.

Thus, the task of the anti-aliasing filter is to suppress potential aliasing components, i.e. signal components which would cause distortion when down-sampling the signal. At this point in the discussion, it is important to note that only the channel-of-interest must not be distorted. But there is no reason why the adjacent channels should not be distorted. They are of no interest. Hence, anti-aliasing is only necessary in a possibly small frequency band. In order to understand the effect of this anti-aliasing property it is useful to introduce the over-sampling ratio (OSR) of a signal, i.e. the ratio between the sample rate $f_S$ of the signal, and the bandwidth $b$ of the signal-of-interest (i.e. the region to be kept free from aliasing).

$$\text{OSR} = \frac{f_S}{b} \tag{40}$$

From Figure 6.14 it becomes clear that there are no restrictions as to how the frequencies are occupied outside the spectrum of the signal-of-interest (e.g. by adjacent channels). This reflects a general view on oversampling.

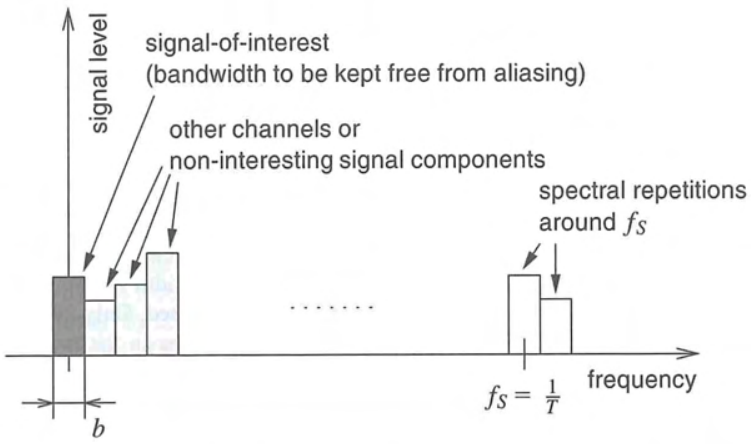


**Figure 6.14** Illustrating the oversampling ratio (OSR) of a signal

The relative bandwidth (compared to the sample rate) of potential aliasing components (that must be attenuated by the anti-aliasing filter) depends on the OSR *after* sample rate reduction. The higher the OSR is, the smaller the pass-band and the stop-bands can be for this filter. Hence, it can be concluded that a high OSR (after sample rate reduction) allows a wide transition band $\Delta f$ of the filter and therefore leads to a smaller number of coefficients (see Equation (39).

Further details on sample rate reduction as a special type of sample rate conversion are discussed in Section 6.5. The possible savings of multirate filtering are illustrated with the following example.

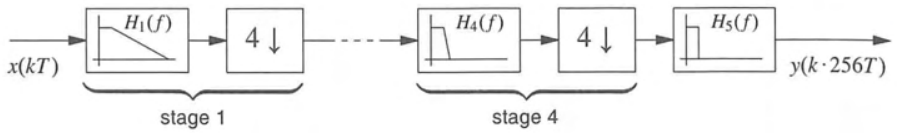


**Figure 6.15** Structure of a multirate filter

**Example 6.4.1**

Assuming a sample rate of $f_S = 100$ MSps, a channel bandwidth of $b = 200$ kHz, a transition bandwidth of $\Delta f = 40$ kHz, and a filter-type specific proportionality factor $C$, the number of coefficients of a direct implementation is, with Equation (39),

$$K_{\text{direct}} = C\frac{f_S}{\Delta f} = C\frac{100 \text{ MHz}}{40 \text{ kHz}}$$

$$\approx C \times 2500$$

Further, assuming decimation by 256, only every 256th sample at the output of the filter needs to be calculated. This results in a multiplication rate (in millions of multiplications per second, Mmps) of

$$\Psi(K_{\text{direct}}) = K_{\text{direct}}\frac{f_S}{256} \approx C \times 980 \text{ Mmps}$$

Now a multirate filter with four stages should be applied instead, each stage decimating the signal by a factor of 4. After these four filters and down-samplers, a fifth filter does the final filtering (see Figure 6.15). In this case the transition band of the first four filters is equal to the difference of the sample rate after decimation minus the bandwidth of the channel. This ensures that potential aliasing components are sufficiently attenuated. Only in the fifth filter is the transition bandwidth set to 40 kHz. The same filter type as in the previous case is assumed, hence the same factor $C$.

$$K_{\text{multirate}} = \sum_{i=1}^{5} K_i$$

$$= C\left[\sum_{i=1}^{4} \frac{\frac{100 \text{ MHz}}{4^{i-1}}}{\frac{100 \text{ MHz}}{4^i} - 200 \text{ kHz}} + \frac{\frac{100 \text{ MHz}}{4^4}}{40 \text{ kHz}}\right]$$

$$\approx C \times 30.7$$

Each of the filter stages runs at the lower sampling rate. Thus, the resulting multiplication rate is

$$\sum_{i=1}^{5} \Psi(K_i) = \sum_{i=1}^{4} K_i\frac{f_S}{4^i} + K_5\frac{f_S}{4^4}$$

$$\approx C\left(4\frac{f_S}{4}+4.1\frac{f_S}{16}+4.6\frac{f_S}{64}+8.2\frac{f_S}{256}+9.8\frac{f_S}{256}\right)$$

$$\approx C\times 141 \text{ Mmps}$$

There is a saving in terms of multiplications per second of a factor 7, while the hardware effort can be reduced by a factor of 81 in the case of multirate filtering. It should be stressed that this is an example. The figures can vary considerably in different applications. However, despite being very special, this example shows the potential of savings that multirate filtering offers.

Even more savings are possible by employing different filter types for the separate stages in a multirate filter. The above mentioned factor $C$ is a proportionality factor that was selected for the direct implementation, e.g. a conventional FIR filter. In the case of multirate filtering it has been seen that in the first few stages the OSR is very high. This results in relatively large transition bands. In other words, the stop bands are very narrow. Hence, comb filters sufficiently attenuate these narrow stop-bands. A well-known class of comb filters are cascaded integrator comb filters (CIC filters) [14]. These filters implement the transfer function

$$H(z)=\left(\sum_{i=0}^{M-1}z^{-i}\right)^R=\left(\frac{1-z^{-M}}{1-z^{-1}}\right)^R$$

without the need for multipliers. $M$ is the sample rate reduction factor and $R$ is called the order of the CIC filter. Ony adders, subtractors, and registers are needed. Hogenauer [14] states that these filters generally perform sufficiently for decimating down to four times the Nyquist rate. Employing these filters in the first three stages of the above example yields $K_1 = K_2 = K_3 = 0$ (i.e. no multiplications required) which would result in a multiplication rate of as low as $C\times 7$ Mmps. This is a considerable saving compared to the direct implementation of a low-pass filter followed by 256 times down-sampling.

A great advantage of CIC filters is that they can be adapted to different rate change factors by simply choosing $M$. There is no need to calculate new coefficients or to change the underlying hardware. Thus, they are a very flexible solution for software defined radio transceivers. However, as mentioned the OSR after decimation should be at least 4. Thus the necessary remaining channel-filtering (and possibly matched filtering) can be achieved with a cascade of two half-band filters, each followed by decimation by 2. Half-band filters are optimized filters (often conventional FIR filters) for decimation by 2. The half-band filters do not need to be tunable. Their output sample rate and thus, the signal bandwidth is always half of that at the input. Hence, by changing the rate-change factor in the CIC filter preceding the half-band filters, the bandwidth of the overall channel filter is tuned. A final 'cosmetic' filtering can be applied to the signal at the lowest sample rate. The respective filter must be tunable in certain limits, e.g. it must be able to implement root-raised-cosine filters with different roll-off factors for matched filtering purposes.

For further reading on multirate filtering, the reader is referred to the literature, e.g. [1].

### *6.4.2 Band-Pass Filtering before Digital Down-Conversion*

#### 6.4.2.1 Complex Band-Pass Filtering

Assuming that the channel-of-interest is perfectly selected by the low-pass channel filter with the discrete-time impulse response $h_{LP}(kT)$ (no down-sampling after filtering), it can be written:

$$\hat{\underline{x}}_{dig,BB}(kT) = \sum_{i=-\infty}^{+\infty} h_{LP}((k-i)T)\underline{x}_{dig,BB}(iT) \tag{41}$$

where $\hat{\underline{x}}_{dig,BB}(kT)$ represents the channel-of-interest according to Equation (10):

$$\hat{\underline{x}}_{dig,BB}(kT) = \frac{1}{2}\underline{x}_{Tx,BB}(kT) \tag{42}$$

Substituting Equation (9) into Equation (41) yields

$$\hat{\underline{x}}_{dig,BB}(kT) = \sum_{i=-\infty}^{+\infty} h_{LP}((k-i)T)\underline{x}_{dig,IF}(iT)e^{-j2\pi f_{IF}iT} \tag{43}$$

Extracting the factor $e^{-j2\pi f_{IF}kT}$, we have

$$\hat{\underline{x}}_{dig,BB}(kT) = e^{-j2\pi f_{IF}kT}\sum_{i=-\infty}^{+\infty} h_{LP}((k-i)T)\,\underline{x}_{dig,IF}(iT)e^{j2\pi fIF(k-i)T} \tag{44}$$

$$= e^{-j2\pi f_{IF}kT}\sum_{i=-\infty}^{+\infty} \underline{h}_{BP}((k-i)T)\,\underline{x}_{dig,IF}(iT) \tag{45}$$

with

$$\underline{h}_{BP}(kT) = h_{LP}(kT)\,e^{j2\pi f_{IF}kT} \tag{46}$$

The latter is the impulse response of the low-pass filter frequency shifted by $f_{IF}$. It is a complex band-pass filter. The digitized IF signal $\underline{x}_{dig,IF}(kT)$ is filtered with this complex band-pass filter before it is down-converted to baseband. Hence, the down-conversion followed by low-pass filtering can equivalently be performed by means of complex band-pass filtering followed by down-conversion.

Both solutions are equivalent in terms of their input–output behavior. However, there are differences with respect to implementation and realization. Since down-conversion is explicitly necessary in both cases, only the filtering operations should be compared.

The length of both impulse responses, the band-pass filter's and the low-pass filter's, are the same. However, the impulse response of the low-pass filter $h_{LP}(kT)$ is real. Hence, each addend of the sum of Equation (41) is a result of multiplying a complex number (i.e. a sample of the complex signal $\underline{x}_{dig,BB}$) with a real number (i.e. a sample of the real impulse response $h_{LP}$). Consequently, each addend requires two real multiplications, resulting in $2K$ multiplications per output sample if $K$ is the length of the impulse response.

In the case of complex band-pass filtering, Equations (45) and (46) suggest that each addend is a result of a complex multiplication (i.e. a multiplication of a sample of the complex signal $\underline{x}_{dig,IF}$ and the complex impulse response $\underline{h}_{BP}$) that is equivalent to four real multi-

plications. Hence, the resulting multiplication rate is $4K$ multiplications per output sample which is twice the rate required for low-pass filtering after down-conversion.

Since there are no advantages of complex band-pass filtering over real low-pass filtering, the higher effort disqualifies complex band-pass filtering as an efficient solution to channelization, at least if it is implemented as described in this section. However, complex band-pass filtering plays an important role in filter bank channelizers (Section 6.4.3).

However, there are certain cases where the multiplication rate of a complex band-pass filter can be halved. This is the case for instance if the IF in Equation (46) is $f_{IF} = 1/(4T) = f_S/4$. In this case the exponential function becomes the simple sequence $\{e^{j(\pi/2)k}\} = \{\mathbf{1}, j, \mathbf{-1}, \mathbf{-}j, 1, j, -1, -j, \ldots\}$ whose samples are either real or imaginary. Thus, two of the four real multiplications required for each addend in Equation (45) are dropped. Even the following digital down-conversion can be simplified when applying harmonic subsampling by a multiple of 4 (see Section 6.3.6), provided that the sampling theorem is obeyed. This is sketched in Figure 6.16.

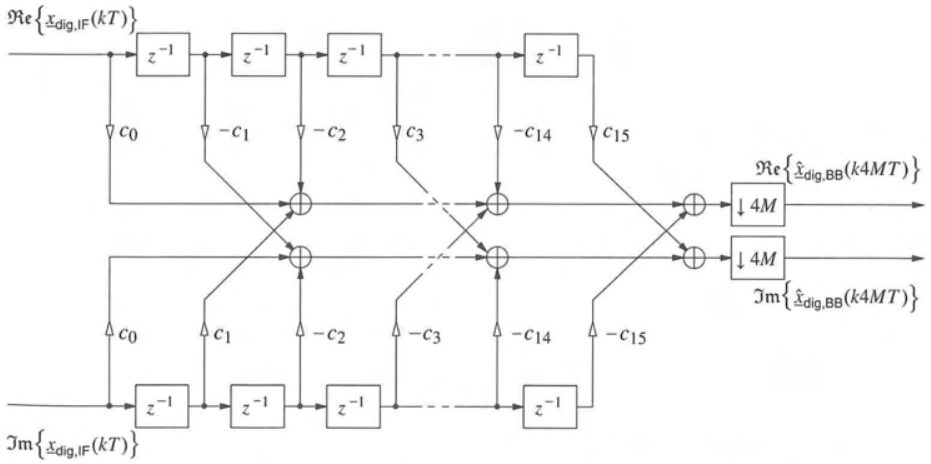


**Figure 6.16** Channelization by simplified complex band-pass filtering at $f_{IF} = f_S/4$ followed by harmonic subsampling by $4M, M \in \{1, 2, \ldots\}$ (the coefficients $c_i$ are identical to those of the equivalent 16-tap FIR low-pass filter that follows the digital down-converter in a conventional system; see Section 6.4.1)

With the assumption that $f_{IF} = f_S/4$ the multiplication rate of low-pass filtering after digital down-conversion can also be halved. In this case digital down-conversion can be realized by multiplying the signal with the sequence $\{e^{-j(\pi/2)k}\} = \{\mathbf{1}, \mathbf{-}j, \mathbf{-1}, j, 1, -j, -1, j, \}$. The result is a complex signal whose samples are mutually pure imaginary or real enabling the multiplication rate to be halved.

It should be noted that due to the fixed ratio between IF and sample rate, the channel-of-interest must be shifted to IF by proper analog down-conversion in the AFE prior to digital down-conversion and channel filtering.

#### 6.4.2.2 Real Band-Pass Filtering

The question is, can the number of necessary multiplications be reduced when employing real instead of complex band-pass filtering? The impulse response of a real band-pass filter can be obtained by taking the real part of Equation (46):

$$\tilde{h}_{\mathrm{BP}}(kT) = \Re\mathrm{e}\{\underline{h}_{\mathrm{BP}}(kT)\} \tag{47}$$

$$= \Re\mathrm{e}\left\{h_{\mathrm{LP}}(kT)\, e^{j2\pi f_{\mathrm{IF}}kT}\right\} \tag{48}$$

$$= h_{\mathrm{LP}}(kT)\, \cos\left(2\pi f_{\mathrm{IF}}kT\right) \tag{49}$$

$$= h_{\mathrm{LP}}(kT)\frac{1}{2}\left(e^{j2\pi f_{\mathrm{IF}}kT} + e^{-j2\pi f_{\mathrm{IF}}kT}\right) \tag{50}$$

Filtering the signal $\underline{x}_{\mathrm{dig,IF}}$ with this real band-pass filter yields

$$\underline{\tilde{x}}_{\mathrm{dig,IF}}(kT) = \sum_{i=-\infty}^{+\infty} \tilde{h}_{\mathrm{BP}}((k-i)T)\, \underline{x}_{\mathrm{dig,IF}}(iT) \tag{51}$$

$$= \sum_{i=-\infty}^{+\infty} h_{\mathrm{LP}}((k-i)T)\frac{1}{2}\left(e^{j2\pi f_{\mathrm{IF}}(k-i)T} + e^{-j2\pi f_{\mathrm{IF}}(k-i)T}\right)\underline{x}_{\mathrm{dig,IF}}(iT) \tag{52}$$

Equivalent to Equation (45), the filtered signal is eventually down-converted to baseband:

$$\underline{\tilde{x}}_{\mathrm{dig,BB}}(kT) = e^{-j2\pi f_{\mathrm{IF}}kT}\underline{\tilde{x}}_{\mathrm{dig,IF}}(kT) \tag{53}$$

The complex exponential function of Equation (53) can be combined with the complex exponentials which $\underline{\tilde{x}}_{\mathrm{dig,IF}}$ comprises.

$$\underline{\tilde{x}}_{\mathrm{dig,BB}}(kT) = \frac{1}{2}\sum_{i=-\infty}^{+\infty} h_{\mathrm{LP}}((k-i)T)\left(\underline{x}_{\mathrm{dig,IF}}(iT)\, e^{-j2\pi f_{\mathrm{IF}}iT} + \underline{x}_{\mathrm{dig,IF}}(iT)\, e^{-j2\pi f_{\mathrm{IF}}(2k-i)T}\right) \tag{54}$$

Applying Equation (9) to the first term and using Equation (41), we can write

$$\underline{\tilde{x}}_{\mathrm{dig,BB}}(kT) = \frac{1}{2}\underline{\hat{x}}_{\mathrm{dig,BB}}(kT) + \frac{1}{2}e^{-j2\pi f_{\mathrm{IF}}2kT}\sum_{i=-\infty}^{+\infty} h_{\mathrm{LP}}((k-i)T)\, \underline{x}_{\mathrm{dig,IF}}(iT)\, e^{j2\pi f_{\mathrm{IF}}iT} \tag{55}$$

and moreover with Equation (42),

$$\underline{\tilde{x}}_{\mathrm{dig,BB}}(kT) = \frac{1}{4}\underline{x}_{\mathrm{Tx,BB}}(kT) + \frac{1}{2}e^{-j2\pi f_{\mathrm{IF}}2kT}\sum_{i=-\infty}^{+\infty} h_{\mathrm{LP}}((k-i)T)\, \underline{x}_{\mathrm{dig,IF}}(iT)e^{j2\pi f_{\mathrm{IF}}iT} \tag{56}$$

Thus, the signal obtained from real band-pass filtering and down-conversion is a sum of two components. The first is a scaled version of the transmitted signal, i.e. the signal-of-interest, while the second term is obviously the result of the following operations: the wide-band IF-signal $\underline{x}_{\mathrm{dig,IF}}$ is shifted in frequency and low-pass filtered; the resulting low-pass signal is finally shifted in frequency by $-2f_{\mathrm{IF}}$. It is a narrow-band signal centered at $-2f_{\mathrm{IF}}$ with a bandwidth equal to the pass-band width of the employed filter. Thus, it does not distort the signal-of-interest at baseband. However, for further signal processing it might be necessary to

remove this 'high-frequency' component of $\underline{\tilde{x}}_{\mathrm{dig,BB}}$. This can be achieved with another low-pass filter.

It can be concluded that the multiplication rate of complex band-pass filtering can be halved by means of using real band-pass filtering. In order to achieve similar results as with complex band-pass filtering, real band-pass filtering and down-conversion must be followed by a low-pass filtering step. This increases the multiplication rate again.

#### 6.4.2.3 Multirate Bandpass Filtering

Multirate filtering yields savings in hardware effort and multiplication rate compared to the direct implementation of low-pass filters. These savings are also possible with band-pass filters. In the case of complex band-pass filtering the same restrictions as in the low-pass filtering case apply, namely the sampling theorem: the sample rate must be at least as high as the (two-sided) bandwidth of the signal.

In the case of real band-pass filtering the band-pass filtered signal comprises another signal component besides the channel-of-interest. Therefore the band-pass sampling theorem applies, which states that the sample rate must be at least twice as high as the (one-sided) bandwidth $B$ of the signal. Moreover, the following relation must hold:

$$\frac{Mf_S + B}{2} < f_c < \frac{(M+1)f_S - B}{2} \tag{57}$$

where $M$ is an integer number, $f_S$ is the sample rate, $f_c$ is the center frequency of the signal band of bandwidth $B$. Obeying the above relation ensures that the sample rate reduction does not cause any overlap (aliasing) between the two signal components of $\underline{\tilde{x}}_{\mathrm{dig,IF}}$.

Multirate band-pass filtering results in similar savings as illustrated in Example 6.4.1 (see also [1]). However, complex band-filtering requires twice the multiplication rate of the equivalent low-pass filter. A reduction of this rate as suggested above depends on the relation between $f_{IF}$, $f_S$, and the decimation factors of the multirate filter. Similar dependencies must be obeyed when using *real* multirate band-pass filters as Equation (57) suggests. Due to these restrictions, multirate bandpass filtering is generally not the first choice for channelization in software defined radio transceivers.

### *6.4.3 Filterbank Channelizers*

#### 6.4.3.1 Channelization in Base Stations

In base stations it is necessary to process more than one channel simultaneously. Basically, there are two solutions to achieve this. The first is simply to have $N$ one-channel channelizers in parallel (if $N$ is the number of channels to be dealt with in parallel). This approach is sometimes referred to as the 'per-channel approach'. In contrast to this there is the filter bank approach that is based on the idea of having one common channelizer for all channels.

The per-channel approach looks simplistic and rather brute force. Still it offers some very important advantages:

- the channels are completely independent and can therefore support signals with different bandwidth
- a practical system is easily scalable by simply adding or removing a channel

- a failure of one channel does not affect the others.

The first bullet might not appear to be an obvious advantage over the filter bank approach. However, this will become clear in the sequel to this section when feasible filter bank structures are discussed. Filter banks are comprehensively covered in the literature. Three important books are [1,4,23]. The most typical applications are signal analysis and sub-band coding. A special type of filter bank, namely the transmultiplexer, has been widely applied in communications systems. A most recent application of the transmultiplexer is orthogonal frequency division multiplex (OFDM) which is used in digital broadcasting systems. Several (possibly independent) data signals are mapped onto parallel frequency divided carriers. In an OFDM receiver these frequency divided channels are separated and converted to baseband which eventually yields the originally sent data channels. This task is very similar to the functionality of a channelizer. Therefore an attempt has been made to apply the filter bank approach to channelization in base stations [26]. In the following, the basic ideas of filter bank channelizers are presented.

### 6.4.3.2 Filter banks

A filter bank is simply a parallel arrangement of filters that usually provide at their outputs different parts of the spectrum of the signal. This is sketched in Figure 6.17. It looks very similar to the per-channel-approach. In the following only digital filter banks are considered, i.e. filter banks with digital filters.

A filter bank is called a *modulated filter bank* if the transfer functions of all involved filters

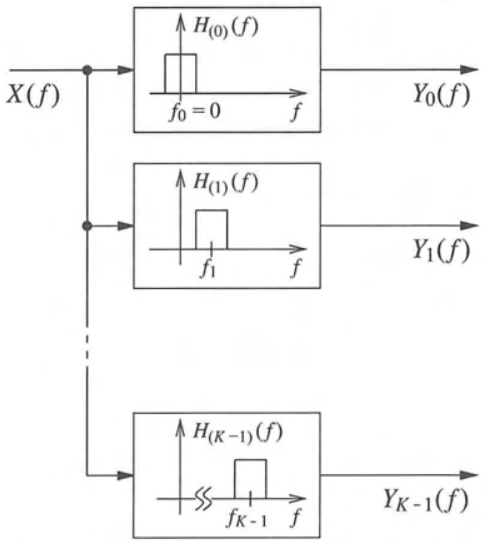


**Figure 6.17** The basic principle of a filter bank ($H_{(i)}(f)$ are frequency characteristics of independent individual filters)

can be derived by modulating one prototype transfer function. The modulation can be, e.g. the discrete Fourier transform (DFT), or the cosine transform. If, moreover, the center frequencies $f_k$ of the $K$ filters are evenly distributed over the available frequency band $[0, f_S)$ where $f_S$ is the sample rate, e.g.

$$f_k = \frac{k}{K} f_S, \quad k = 0, 1, \ldots, K-1 \tag{58}$$

the filter bank is called a *uniform filter bank*. The channel stacking arrangement of Equation (58) is called an *even* channel stacking ($f_0 = 0$).

What is the reason to build a uniform filter bank? If the overall cost (hardware, multiplication rate) for implementing the prototype filter and the modulation is less compared to the parallel arrangement of $K$ independent filters, there is a good reason to employ filter banks.

In order to derive the uniform DFT filter bank it is useful to remember the polyphase decomposition of a transfer function of a filter [4]. It is

$$H(z) = \sum_{\lambda=0}^{N-1} H_\lambda^{[\mathrm{P}]}(z) \tag{59}$$

with the $N$ polyphase components

$$H_\lambda^{[\mathrm{P}]}(z) = z^{-\lambda} H_\lambda\left(z^N\right) = z^{-\lambda} \sum_{n=-\infty}^{\infty} h((nN+\lambda)T)\, z^{-nN} \tag{60}$$

The $H_\lambda(z)$ are the time-shifted and down-sampled polyphase components. Different types of the polyphase decomposition are named in the literature [4,21]. The one given above is commonly called the *Type-1* polyphase decomposition.

Defining a vector comprising all $N$ time-shifted and down-sampled polyphase components,

$$\underline{\mathbf{H}}(z) := [\, H_0(z) \quad H_1(z) \quad \cdots \quad H_{N-1}(z)]^T \tag{61}$$

and another vector comprising all of the $N$ modulated transfer functions of the filter $H(z)$

$$\underline{H}^{[\mathrm{M}]}(z) := \left[\, H(z) \quad H\left(z \cdot w_N^{+1}\right) \quad \cdots \quad H\left(z \cdot w_N^{+(N-1)}\right) \right]^T \tag{62}$$

with

$$w_N^k = e^{-j2\pi(k/N)} \tag{63}$$

we can write

$$\underline{H}^{[\mathrm{M}]}(z) = W_N^* \cdot \Lambda^{-1}(z) \cdot \underline{\mathbf{H}}\left(z^N\right) \tag{64}$$

where $W_N^*$ is the transjugate of the DFT matrix

$$W_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_N^1 & w_N^2 & \cdots & w_N^{(N-1)} \\ 1 & w_N^2 & w_N^4 & \cdots & w_N^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & w_N^{(N-1)} & w_N^{2(N-1)} & \vdots & w_N^{(N-1)^2} \end{bmatrix} \tag{65}$$

and

$$\Lambda(z) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & z & 0 & \cdots & 0 \\ 0 & 0 & z^2 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & z^{N-1} \end{bmatrix} \tag{66}$$

Thus, all subfilters of a uniform DFT filter bank can be built from the time-shifted and down-sampled polyphase components of one prototype filter. Usually it is a low-pass filter, e.g. the filter with frequency characteristics $H_{(0)}(f)$ of Figure 6.17. In order to derive a structure from the above equations, the factors of Equation (64) are rearranged:

$$\underline{H}^{[\mathrm{M}]}(z) = W_N^* \cdot \Lambda^{-1}(z) \begin{bmatrix} H_0(z^N) & 0 & \cdots & 0 \\ 0 & H_1(z^N) & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & H_N - 1(z^N) \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{67}$$

$$= W_N^* \begin{bmatrix} H_0(z^N) & 0 & \cdots & 0 \\ 0 & H_1(z^N) & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & H_N - 1(z^N) \end{bmatrix} \begin{bmatrix} 1^{-1} \\ \vdots \\ z^{-(N-1)} \end{bmatrix} \tag{68}$$

Instead of implementing the individual subfilters directly, i.e. the elements of $\underline{H}^{[\mathrm{M}]}(z)$, the filter bank can be realized by implementing Equation (68). Since the signal bandwidth at the output of each of the $N$ subfilters is reduced by the factor $N$, the sample rate can also be reduced by the factor $N$ (if the filters provide enough aliasing attenuation). This decimation also converts the individual bandpass signals to baseband (see Section 6.3.6). The resulting structure is called a *maximally decimated filter bank*. It is sketched in Figure 6.18 where the decimation has been moved to the input of the polyphase filters thereby removing the power of $N$ that is present in Equation (68). The structure of Figure 6.18 is what is often understood by the notion *filter bank*.

Implementing the DFT by a fast Fourier transform (FFT) results in comparably low effort. $N$ must be a power of two. In many practical cases, it is sufficient to select only a small number of channels (e.g. $<10$). In case $N$ is large it is not efficient to calculate the output of all $N$ channels if only a few are required. In this case the Goerzel algorithm [20] is more efficient than the FFT. The Goerzel algorithm is an algorithm that computes only one point of an $N$-point DFT. It is basically a one-pole IIR filter with its pole placed on the unit circle at $e^{j2\pi f_i}$ where $f_i$ is the point of the DFT to be calculated. Depending on the implementation of the filter the algorithm requires approximately $N$ multiplications. If $M$ points should be calculated the number of required multiplications is approximately $MN$. Comparing this to the typically assumed $N \log_2 N$ multiplications of an $N$-point FFT, gives the number of points $M$ for which the Goerzel algorithm is computationally more efficient than the FFT. This is $M < \log_2 N$.

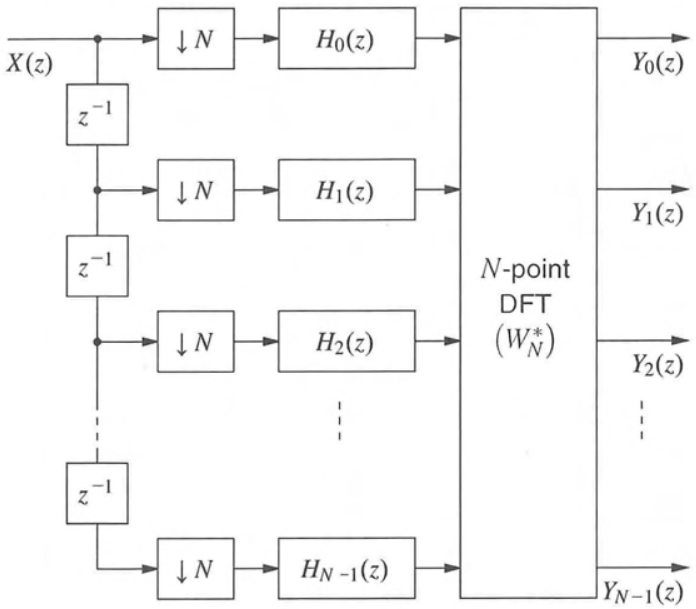


**Figure 6.18** A maximally decimated uniform DFT filter bank ($H_i(z)$ are transfer functions of the polyphase components of $H(z)$)

Since the described filter bank analyses the signal, it is called an analysis filter bank. Its counterpart is the synthesis filter bank that takes $N$ narrowband signals at its input and synthesizes a wide-band signal at its output.

#### 6.4.3.3 The Filter Bank as a Channelizer

By setting $N = K$ the filter bank can be used for channelization. What are the characteristics of a filter bank channelizer? If a maximally decimated filter bank is used, the input sample rate as well as the input signal bandwidth is divided by $K$. Hence, the output sample rate and the input sample rate are related by an integer factor. The same holds for the signal bandwidth at the input and each of the $K$ outputs. In order to break the dependency between the decimation factor and the channel stacking, several methods have been suggested [1,26]. However, some severe drawbacks of (maximally decimated) filter bank channelizers remain in the context of software radio applications:

- all channels have the same bandwidth (which is a drawback in base stations that should support different services simultaneously, e.g. GSM and UMTS)
- the channel stacking depends on the number of channels and the input sample rate
- in the case of a large number of channels, the filter must be a narrow-band filter

These three bullets are discussed in the following. Basically, there are two approaches to overcome the first problem: tree-structured filter banks can be used to split the input signal

bandwidth into few channels of relatively large bandwidth. These channels can further be decomposed by another filter bank, etc. However, the position of wide-band and narrow-band channels in terms of their center frequency is strongly connected to the structure of the filter bank (e.g. the number of channels). Another means to serve different bandwidths is to use an analysis filter bank that decomposes the signal into a very large number of narrow-band signals. Any group of consecutive narrow-band signals can be merged to a signal of wider bandwidth by a following synthesis filter bank. However, such an analysis filter bank would require filters with very narrow bandwidth resulting in immense effort.

The second bullet cannot be overcome. Hence, the center frequencies of the individual filters are always located at integer multiples of a $K$th of the input sample rate (except for a possible frequency offset that is equal for all filters). A first choice to meet this problem could be to adapt the input sample rate. However, the channels-of-interest can be arbitrarily distributed in frequency. Therefore the complete wide-band input signal must be regarded as a signal-of-interest. As will be seen in Section 6.5, sample rate conversion is not feasible for such very wide-band signals. So, how to deal with the fixed channel stacking? Can it be made a 'soft' channel stacking? Not really, since the discrete channel stacking is a property of the DFT. However, a 'soft' channel stacking can virtually be achieved with a very large $K$ (i.e. a fine quantization of the frequency axis). Although this results in an increased effort for the DFT, the cost for the filter does not necessarily rise. The bandwidth of the prototype filter is chosen according to the channel bandwidth; by choosing a large $K$ this bandwidth can be centered at an 'arbitrary' position (see Figure 6.19). Obviously, such a filter bank is not a maximally decimated one.

The last bullet results from the fact that in software radio receivers a maximal bandwidth is digitized that might comprise narrow-band channels. In the case of for example, a sample rate of 40 MHz, and a channel bandwidth of 200 kHz, the OSR is 200. Thus, the relative pass-band width of a respective channel is 0.005 which results in very strong constraints if the adjacent channels are to be attenuated. Since a filter bank implementation requires all polyphase components of the filter, the complete impulse response of the filter must be implemented directly. Cascaded multirate systems as suggested in Section 6.4 are not possible. This means that in the case of FIR filtering several hundreds or even thousands of filter coefficients must be implemented.

Certainly, the effort of a filter bank channelizer can compete with the per-channel approach if the OSR is much lower. However, in software defined base stations narrow-band signals must also be processed.

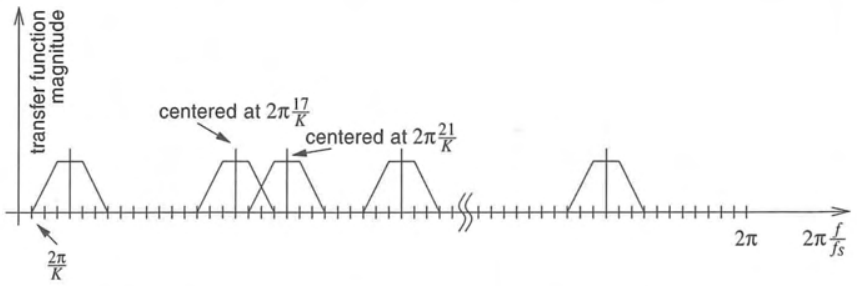


**Figure 6.19** Virtually arbitrary channel stacking with a large number of filter bank channels

Summarizing, it can be said that filter bank channelizers suffer from many structure-dependent restrictions that can only be overcome with great effort. Therefore, it is questionable if filter bank channelizers are a good solution for channelization in software defined base stations.

## 6.5 Sample Rate Conversion[1]

### *6.5.1 Resampling after Reconstruction*

Software defined radio requires as many functionalities as possible to be programmable or at least parameterizable by means of software. This is difficult in the case of analog signal processing. Therefore it is a first target to implement as many functionalities as possible by means of digital signal processing. For that reason a fixed sample rate has been suggested for the analog/digital interface in Section 6.2. In order to provide signals at symbol- or chip-rate (i.e. an arbitrary target rate), sample rate conversion (SRC) has been introduced as one of the functionalities of the DFE of a software defined radio transceiver.

Having a digital signal at a sample rate $f_1 = 1/T_1$, the brute force approach to provide this signal at another sample rate $f_2 = 1/T_2$ is to reconstruct this signal (i.e. perform digital-to-analog conversion), and eventually resample this reconstructed signal. This approach is sketched in Figure 6.20. Although not sensible in practice, *resampling after reconstruction* is a very good model that enables the SRC process to be characterized. Therefore it is used here.

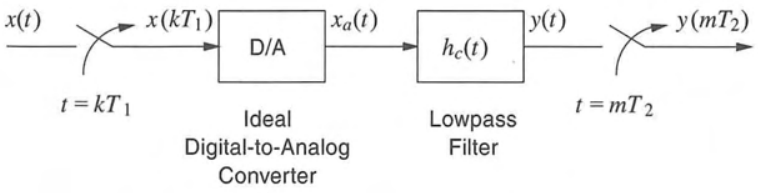


**Figure 6.20** Resampling after reconstruction (adapted from [12], © 2000 IEEE)

The analog-to-digital conversion process is assumed to be ideal, i.e. the samples $x(kT_1)$ are simply taken as weights for a Dirac-impulse comb with period $T_1$:

$$x_a(t) = T_1 \sum_{k=-\infty}^{\infty} x(kT_1)\delta(t - kT_1) \tag{69}$$

Filtering the signal $x_a(t)$ with a low-pass filter having the impulse response $h_c(t)$ yields

$$y(t) = \int_{\tau=-\infty}^{\infty} x_a(\tau)h_c(t-\tau)\,d\tau$$

[1] Material for this section is adapted from [9].

$$= \int_{\tau=-\infty}^{\infty} T_1 \sum_{k=-\infty}^{\infty} x(kT_1)\delta(\tau - kT_1)h_c(t-\tau)\,d\tau$$

$$= T_1 \sum_{k=-\infty}^{\infty} x(kT_1)h_c(t-kT_1) \tag{70}$$

Perfect reconstruction, i.e. $y(t) = x(t)$ can be achieved with an ideal low-pass filter with its cut-off frequency at half the sample rate:

$$h_c(t) = \frac{1}{T_1}\mathrm{sinc}\left(\frac{t}{T_1}\right) \tag{71}$$

However, perfect reconstruction is not necessary in most cases as we shall see in the following. Finally, the signal $y(t)$ is resampled with period $T_2$. With $h(t) = T_1 h_c(t)$, the resampled signal is

$$y(mT_2) = \sum_{k=-\infty}^{\infty} x(kT_1)\,h(mT_2 - kT_1) \tag{72}$$

$$= \sum_{k=-\infty}^{\infty} x(kT_1)\,h\left(T_1\left(\frac{mT_2}{T_1} - k\right)\right)$$

$$= \sum_{k=-\infty}^{\infty} x(kT_1)\,h\left(T_1\left(\left\lfloor\frac{mT_2}{T_1}\right\rfloor + \mu_m - k\right)\right)$$

$$= \sum_{n=-\infty}^{\infty} x\left(T_1\left(\left\lfloor\frac{mT_2}{T_1}\right\rfloor - n\right)\right)h(T_1(n+\mu_m)) \tag{73}$$

with

$$\mu_m = \frac{mT_2}{T_1} - \left\lfloor\frac{mT_2}{T_1}\right\rfloor \tag{74}$$

and

$$n = \left\lfloor\frac{mT_2}{T_1}\right\rfloor - k$$

$\lfloor\bullet\rfloor$ denotes the *floor* operation, i.e. $(\bullet)$ is rounded to the nearest integer towards minus infinity. The quantity $\mu_m \in [0, 1)$ reflects the position of the currently to be calculated sample inside the sample period $T_1$. It is called the *intersample position*. The timing relations and the intersample position are illustrated in Figure 6.26.

Equation (73) can be regarded as a direct digital representation of sample rate conversion. The impulse response $h(t)$ of the reconstruction filter is sampled with the period $T_1$ of the input signal, however, with different time offsets $\mu_m T_1$ that depend on the position of the output sample with respect to the input samples. For each output sample a new set of samples of the impulse response $h(t)$ is used. Hence Equation (73) describes a time-varying filtering operation.

In order to understand the characteristics of SRC it is useful to turn to Figure 6.20 and to identify the signal processing steps that fundamentally influence the signal. The first is the ideal digital-to-analog conversion. As mentioned above the output of the ideal DAC provides a Dirac-impulse train with period $T_1$ whose impulses are weighted with the samples $x(kT_1)$. It is well known from signal theory that a Dirac-impulse comb in the time domain corresponds to a Dirac-impulse train in the frequency domain. This leads to a spectral repetition of the signal $x(t)$ (also known as *imaging*).

The second fundamentally influencing signal processing step is the resampling with period $T_2$ which again causes spectral repetition (as every sampling process). If the signal is not properly band limited, spectral overlap occurs (also known as *aliasing*). This overlap cannot be removed once it has occurred. Therefore it must be avoided in advance by respectively band limiting the signal. This is done with the filter $h_c(t)$. Hence, the filter $h_c(t)$ (and thus, $h(t)$) is the point where the quality of the SRC process can be influenced.

Figure 6.21 shows the different signals that appear in the signal processing chain of Figure 6.20. $|Y_a(f)|$ is the magnitude spectrum of the sampled version of $y(t)$, i.e. the resampled signal. Assuming that the sampling period $T_1$ is short enough with respect to the bandwidth of the band limited signal $x(t)$, there is no aliasing due to the first sampling

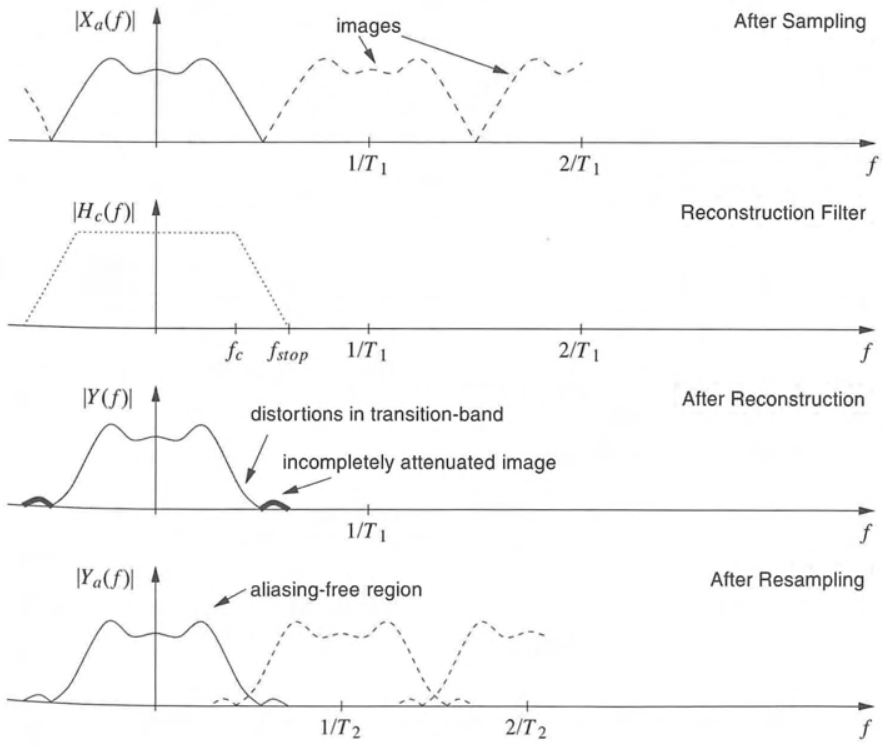


**Figure 6.21** Spectral interpretation of SRC, given for the case $T_1 < T_2$ (adapted from [12], © 2000 IEEE)

process and the digital-to-analog conversion process. The filter $h_c(t)$ reconstructs the continuous-time signal and, moreover, performs anti-aliasing filtering. Since the real filters do not have perfect characteristics, the transition-band causes distortions of the signal, and some residual parts of the first image remain. Resampling with period $T_2$ eventually causes aliasing. However, there is a spectral part of the resampled signal that is free of distortions. When designing SRC for signals comprising several adjacent channel interferers besides the channel-of-interest, care must be taken that only the latter lies in the distortion-free region of the transfer function of the filter. The necessary stop-band attenuation of the filter depends on the power of the potential aliasing components (i.e. the power of the adjacent channels, see Figure 6.3).

Thus, the design of the filter is the first task to be solved when tackling the problem of SRC. The main purpose of this filter is to control aliasing. As stated in [9], *anti-aliasing is the most prominent constraint to be obeyed by any sample rate conversion system*. From Figure 6.21 it can be concluded that the smaller the region which should be kept free from aliasing, the wider the transition band of the filter attenuating the aliasing components can be. This enables the application of comb filters with reduced effort.

### 6.5.2 Rational Factor SRC

In many practical scenarios it is sufficient of consider sampling periods $T_1$ and $T_2$ that can be expressed as a ratio of two (relatively small) positive integers, i.e. $T_1/T_2 = L/M$. Hence, Equation (73) can bewritten as

$$y(mT_2) = \sum_{n=-\infty}^{\infty} x\left(T_1\left(\left\lfloor \frac{mM}{L} \right\rfloor - n\right)\right) h(T_1(n + \mu_m)) \tag{75}$$

In this case, the intersample position $\mu_m = (mM)(\mathrm{mod}\, L)/L$ can only take $L$ distinct values. This reduces the number of relevant sets of samples of $h(t)$ to $L$. The intersample position $\mu_m$ is periodic with period $L$. Therefore, the filter $h$ can be implemented as a periodically time-varying filter with period $LT_2$.

The well-known block diagram of a system for rational factor SRC [1] can be derived by introducing the period $T_0 = T_1/L = T_2/M$, and the two signals $v(nT_0)$ and $w(nT_0)$. This is sketched in Figure 6.22. The signal $w(nT_0)$ is the result of up-sampling the signal $x(kT_1)$ by $L$. It is the output signal of the $L\uparrow$ block that is called an *up-sampler* or a *sample-rate expander*. Up-sampling by $L$ is realized by inserting $L-1$ zeros between two consecutive samples of the original signal. The output signal $y(mT_2)$ is obtained by down-sampling the signal $v(nT_0)$ by the factor $M$. This is done by the $M\downarrow$ block which is called a *down-sampler* or a *sample-rate compressor*. Down-sampling by $M$ is realized by deleting all but every $M$th sample from the signal.

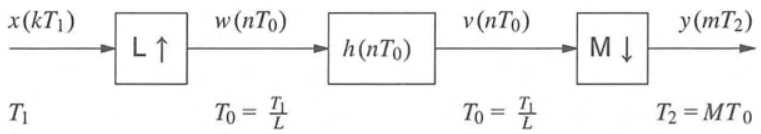


**Figure 6.22** Rational factor SRC (adapted from [12], © 2000 IEEE)

Due to the possibly very high intermediate sample rate $1/T_0$ at which the filter would have to be clocked, the structure of Figure 6.22 is not applicable in many applications. However, it is very useful for investigations.

### *6.5.3 Integer Factor SRC*

Integer factor SRC can be derived from rational factor SRC by simply setting $L$ or $M$ to 1. Since this is a special case of rational factor SRC, all considerations made with respect to rational factor SRC can be applied directly to integer factor SRC.

Increasing the sample rate by an integer factor ($L > 1$, $M = 1$) yields nonoverlapping spectral images. There is no aliasing. The filter is a pure anti-imaging filter. It is also called an *interpolation filter* which – together with the up-sampler – forms an *interpolator*, i.e. a system for integer factor sample rate increase.

Sample rate reduction by $M$ can be achieved by setting $L = 1$. Depending on the bandwidth of the signal before SRC the images resulting from resampling might overlap which means that aliasing occurs. Aliasing can be reduced by proper anti-aliasing filtering. The combination of the anti-aliasing filter and the down-sampler is called a *decimator*.

It should be noted that the necessary filtering that goes with a decimation removes adjacent channel interferers. Thus, SRC and channel filtering can be performed with the same filters.

### *6.5.4 Concepts for SRC*

As mentioned above it is sufficient to perform rational factor SRC in most practical cases (an exception to this is when two asynchronous digital systems are interfaced [1]). Therefore, only rational conversion factors are dealt with in the following. Besides the conversion factor, another design parameter is important: the ratio between the sample rate of the signal and the bandwidth of the channel-of-interest, i.e. the oversampling ratio (OSR) of the channel-of-interest. This has been introduced in Section 6.4.1 in Equation (40).

In order to comprehend the importance of the OSR in the context of SRC it should be recalled that channel filtering is not assumed to have taken place before SRC. The signal whose sample rate should be changed can be a wide-band signal comprising several channels among which there is only one channel-of-interest. The anti-aliasing property stated in Section 6.5.1 means that the sample rate of such a signal is changed so that at least the channel-of-interest is not distorted. All other channels can be completely distorted by aliasing.

The OSR *after* the SRC process determines the relative bandwidth (compared to the sample rate) of potential aliasing components that have to be attenuated by the SRC filter. A high OSR allows small pass-band and stop-bandwidths of this filter. Hence, a high OSR (after SRC) relaxes the design constraints leading to simpler filter structures. A consequence of this is that SRC can be advantageously implemented on a cascaded multirate system as has been proposed for channel filtering. Thus, it is possible to combine SRC and channel filtering. Since the design constraints on the filter for SRC are relaxed for a high OSR, a trade-off between sample rate and hardware effort (number of coefficients) can be made in a multirate system for SRC. It should be noted that this is particularly simple in the case of the per-

channel approach for channelization, which is another advantage of the per-channel approach over the filter bank channelizer.

Several cases of SRC can be distinguished. Assuming $f_2/f_1 = L/M$ is the ratio of the output sample rate and the input sample rate, where $L$ and $M$ are relatively prime (i.e. the greatest common divisor of $L$ and $M$ is one), there is

- an effective reduction ($L < M$), and
- an effective increase ($L > M$) of the sample rate.

Moreover, it can be distinguished between

- an integer factor, and
- fractional SRC.

While integer factor SRC can be combined with channel filtering in a multirate filter, fractional SRC is basically a pure SRC task. Therefore it is sensible not to treat the overall rational conversion factor as it is, but to separate the integer factor from fractional SRC. The rational rate change factor can be factorized to a fractional part $(L/M)_{\text{frac}}$ and an integer part $L_{\text{int}}$ (or $M_{\text{int}}$):

$$\frac{L}{M} = \begin{cases} \left(\frac{L}{M}\right)_{\text{frac}} L_{\text{int}} & \text{effective increase} \\ \left(\frac{L}{M}\right)_{\text{frac}} \frac{1}{M_{\text{int}}} & \text{effective reduction} \end{cases} \tag{76}$$

The fractional factor $(L/M)_{\text{frac}}$ is limited to the interval $(0.5, 2)$. If the sample rate is effectively reduced this factor can be limited further to the interval $(0.5, 1)$, or to the interval $(1, 2)$ if the sample rate is effectively increased.

Having separated fractional SRC from integer-factor SRC the question arises of how to order them. Since the OSR before and after fractional SRC is of the same order of magnitude, the sequential order of fractional and integer-factor SRC has no remarkable influence on the latter. How is this for fractional SRC? Is it better to place fractional SRC first at a high, or last at a low sample rate? When placing fractional SRC at a high sample rate in a multirate

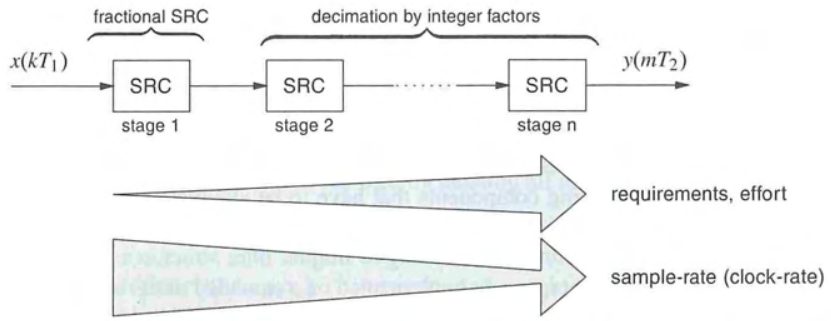


**Figure 6.23** Effort vs. sample rate in a cascaded SRC system realizing an effective sample rate reduction, $T_1 < T_2$, with the suggestion of placing fractional SRC at a high sample rate (adapted from [12], © 2000 IEEE)

system, i.e. where the OSR is high, simple filter structures can be used. However, these filters must run at a high clock rate. At a low rate, more sophisticated filters are required. Thus, the above mentioned trade-off between sample rate and hardware effort can be made for the fractional part of SRC separately.

In Figure 6.23 a cascaded multirate structure for SRC is sketched showing an increasing hardware effort as the sample rate and thus the OSR decreases. A minimization of the hardware effort for fractional SRC can be achieved by placing it first in the cascade where the OSR is highest. However, this does not necessarily yield a minimum multiplication rate.

In general, it is difficult to compare the two extremes of placing fractional SRC at the highest sample rate where the signal-of-interest is highly oversampled, or at the lowest sample rate. This is due to the fact that different filter types would be used for either approach. If the OSR is high, comb filters merely suppressing the aliasing components are a very efficient choice. However, for low OSRs such filters do not perform sufficiently.

Generally, the advantages of placing fractional SRC at a high sample rate (where the OSR is high) are:

- lower hardware complexity due to relaxed constraints
- if noise-shaping ADCs with low output word length (i.e. sigma-delta ADCs) feed the filter, coefficient multipliers can be realized by simple switches or look-up tables of relatively small size.

The disadvantages are:

- the high clock rate at which the filters have to run
- the air interface dependent OSR after fractional SRC that might result in different filter design constraints for different air interfaces
- the required high aliasing attenuation of the filter due to the high dynamic range of multichannel signals.

From these advantages and disadvantages, the pros and cons of placing fractional SRC at a lower sample rate can be derived. Since the requirements of the filters are stronger with lower OSRs the effort for fractional SRC at a lower sample rate is higher. This is mainly reflected in the number of coefficients of the filter and thus, the number of multipliers. If the sample rate is so low that time-division hardware sharing is possible, several coefficient multipliers can be realized by a small number of multiply-accumulate (MAC) units of a DSP. In an FPGA or application specific integrated circuit (ASIC) based implementation, each coefficient is usually realized by its own multiplier. Therefore, in this case it is far more advantageous to place fractional SRC at a high sample rate. This enables the application of simple comb filters whose implementation requires only a small number of multipliers.

### *6.5.5 Systems for SRC*

The *direct approach* of realizing rational factor SRC is a cascade of an up-sampler, a filter, and a down-sampler as shown in Figure 6.22. It can be observed that the filter is placed at a high intermediate sample rate. This is infeasible with input sample rates of some million samples per second.

Alternative structures exploit the fact that all but every $L$th sample are zero at the input to the filter, and all but every $M$th sample are thrown away at the output of the down-sampler.

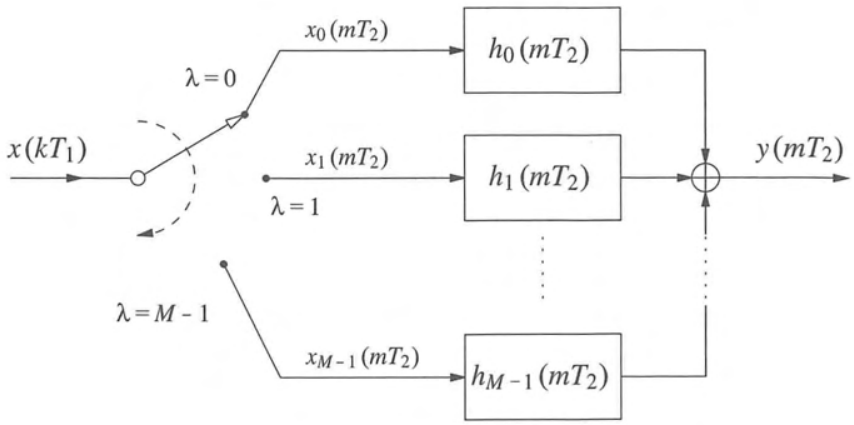


**Figure 6.24** A polyphase decimator

For $M = 1$ or $L = 1$ these structures are known as polyphase interpolators or decimators, respectively [1,4]. A polyphase decimator is shown in Figure 6.24 for illustration purposes. Combining an up-sampler with a polyphase decimator or a polyphase interpolator with a down-sampler yields systems for rational factor SRC that avoid the drawbacks of the direct approach. An up-sampler followed by a polyphase decimator results in exactly the same structure as shown in Figure 6.24 except for the input switch, that steps through the $M$ inputs with a step-size of $L$ rather than 1 (thereby exploiting the fact that all but every $L$th sample are zero of the up-sampled signal).

Polyphase filters are based on the polyphase representation of both the signal and the impulse response of the filter. The most important property of polyphase decimators and interpolators is that the filtering operation is performed on the lower of the two sample rates involved. As to the hardware effort, all coefficients of the filter must also be implemented with a polyphase realization. Thus, having a filter with $K$ coefficients, the polyphase realization must also implement $K$ coefficients. These coefficients can be realized by fixed coefficient multipliers. Since only one polyphase branch is used at a time, it is also possible to realize $K/N$ general purpose multipliers and change the coefficients in a time-varying manner, if $N$ is the number of polyphase branches.

For IIR filters it is not possible to give generally valid estimates of possible savings. This is due to the fact that the polyphase components of IIR filters are also impulse responses of infinite length.

The disadvantage of such a polyphase realization of a sample rate converter is that it is always related to a filter that is specifically designed for a certain rate-change factor. However, in software defined radios the rate change factor must be tunable. How can this be solved?

From Equation (72) it can be seen that the impulse response $h(t)$ is sampled with a period that depends on $T_1$ and $T_2$, i.e. on $L$ or $M$ in the case of integer-factor SRC. Thus, different digital filters (i.e. samples of $h(t)$) result for different $L$ or $M$. Obviously it is not feasible to implement digital filters for all possible factors $L$ or $M$. But it is possible to implement the

continuous-time impulse response $h(t)$ and calculate its samples at the time they are required. In order to keep the effort low for calculating the samples of $h(t)$, simple descriptions of $h(t)$ are sought. Filters whose impulse responses are piecewise polynomials can be used.

Given polynomial pieces of degree $n$ and length $\Delta$

$$p_j(t) = \begin{cases} \sum_{i=0}^{n} c_i(j)\left(\frac{t}{\Delta}\right)^i, & 0 \le t < \Delta \\ 0, & \text{else} \end{cases} \tag{77}$$

a piecewise impulse response composed from $N$ polynomials $p_j(t)$ is

$$h(t) = \sum_{j=0}^{N-1} p_j(t - j\cdot\Delta) \tag{78}$$

An equivalent description is

$$h(t) = \begin{cases} p_{\lfloor t/\Delta \rfloor}\left(t - \left\lfloor \frac{t}{\Delta} \right\rfloor \Delta\right), & 0 \le t < N\Delta \\ 0, & \text{else} \end{cases} \tag{79}$$

where $\lfloor \cdot \rfloor$ denotes the floor operation, i.e. the largest integer smaller than or equal to $(\cdot)$. Equation (79) might seem to be a somewhat odd description. However, it enables us to give up the usual limitations on $t$ for the polynomial pieces in Equation (77) by shifting them to the description of $h(t)$ itself (Equation (79)). Thus, it becomes possible to directly substitute Equation (77) into Equation (79).

$$h(t) = \begin{cases} \sum_{i=0}^{n} c_i\left(\left\lfloor \frac{t}{\Delta} \right\rfloor\right)\left(\frac{t}{\Delta} - \left\lfloor \frac{t}{\Delta} \right\rfloor\right)^i, & 0 \le t < N\Delta \\ 0, & \text{else} \end{cases} \tag{80}$$

An open question is the choice of $\Delta$. There are two choices for which Equation (80) can be simplified considerably and a hardware structure can be derived, namely $\Delta = T_1$ and $\Delta = T_2$.

Combining the polynomial impulse response of Equation (80) and the polyphase approach leads to structures which can be realized very efficiently. One of them is the *Farrow structure* [2,19] resulting from setting $\Delta = T_1$. As will be explained later, it is a structure that is very well suited to interpolation.

An equivalent structure that is perfectly suited to decimation has been suggested by the author [9,10]. Due to its equivalence to the Farrow structure, it has been named the *transposed Farrow structure*. Since it is not as well known but is as important as the Farrow structure, it is explained below.

Setting $\Delta = T_2$ and substituting Equation (80) into Equation (72) yields

$$y(mT_2) = \sum_{k=-\infty}^{\infty} x(kT_1) \sum_{i=0}^{n} c_i\left(\left\lfloor \frac{mT_2 - kT_1}{T_2} \right\rfloor\right)\left(m - k\frac{T_1}{T_2} - \left\lfloor \frac{mT_2 - kT_1}{T_2} \right\rfloor\right)^i, \tag{81}$$

$$0 \le mT_2 - kT_1 < NT_2$$

which can be simplified to

$$y(mT_2) = \sum_{i=0}^{n} \sum_{k=-\infty}^{\infty} \hat{x}_i(kT_1)\, c_i\left(m - \left\lceil k\frac{T_1}{T_2}\right\rceil\right), \quad 0 \le m - \left\lceil k\frac{T_1}{T_2}\right\rceil < N \tag{82}$$

with

$$\hat{x}_i(kT_1) = x(kT_1)\,(\mu_k)^i \tag{83}$$

and

$$\mu_k = \left\lceil k\frac{T_1}{T_2}\right\rceil - k\frac{T_1}{T_2} \in [0, 1) \tag{84}$$

where $\lceil \bullet \rceil$ stands for the ceil operation, i.e. it rounds $(\bullet)$ to the nearest integer towards plus infinity. It should be noted that there are coefficients $c_i(j)$ for certain values of $j$ only. Hence, the summation over $k$ is restricted as given in Equation (82). For reasons of simplicity and clarity, we have not inserted the real limits.

The quantity $\mu_k$ indicates the distance between the current input sample and the next output sample. It is called the intersample position and it is related to $\mu_m$ of Equation (74) by $\mu_k T_2 = \mu_m T_1$ as can be seen from Figure 6.26. From Equations (82)–(84), a structure can be derived that realizes SRC with piecewise polynomial filters with a piece length of $\Delta = T_2$. This is shown in Figure 6.25. In order to see the equivalence between Equations (82)–(84) and the structure of Figure 6.25 it is important to understand Equation (82). Its second sum describes a convolution-like operation.

For any fixed $m$ certain consecutive samples $\hat{x}_i(kT_1)$ might be weighted with the same $c_i(l)$ before being summed up and contributing to a new output sample. This comes from the fact that $\lceil kT_1/T_2 \rceil$ does not necessarily change when incrementing $k$ (see Figure 6.26 where two

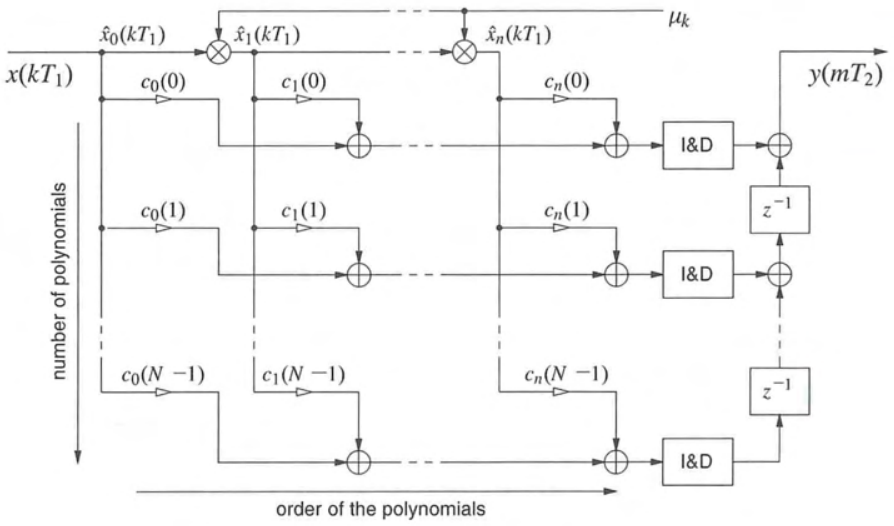


**Figure 6.25** Transposed Farrow structure for decimation, I&D stands for integrate-and-dump (integrate with period $T_1$ and dump with period $T_2$)

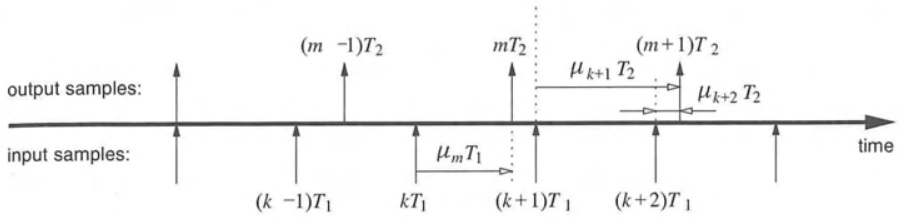


**Figure 6.26** Sample time relations

input samples arrive at $(k+1)T_1$ and $(k+2)T_1$ before a new output sample is generated at $(m+1)T_2$). The respective summation can be realized by means of an *integrate-and-dump* circuit.

It should be noted that the above mentioned (original) Farrow structure can be derived in the same manner as the transposed Farrow structure by simply setting $\Delta = T_1$ (see [10]).

The fundamental difference about the performance of the Farrow structure and the transposed Farrow structure is the length $\Delta$ of the polynomial pieces. In [22] it has been stated that polynomial filters with polynomial length $\Delta$ derived from Lagrange's interpolation formula have transfer zeros clustered at integer multiples of $1/\Delta$. Thus, setting $\Delta = T_1$ results in a filter that attenuates the signal components at integer multiples of $1/T_1$, which are the image components in the case of interpolation. If decimation is the issue, the aliasing components must be attenuated that lie at integer multiples of $1/T_2$. This can be achieved with the transposed Farrow structure, i.e. with $\Delta = T_2$. So it seems as if the Farrow structure is advantageous in the case of interpolation, and the transposed Farrow structure in the case of decimation. This is indeed the case.

The Farrow structure is independent of both $T_1$ and $T_2$. Its signal processing characteristics are determined by the polynomial coefficients $c_i(j)$ and the period $T_1$. These characteristics are completely independent of $T_2$. Changing $T_1$ automatically adapts the underlying continuous-time impulse response respectively; changing $T_2$ (which is a fundamental property of a system for SRC in a software radio) does not cause any adaptation. However, changing $T_2$ influences the characteristics of the resampling process (see Figure 6.20) and thus influences aliasing. These changes could only be met by designing a new filter thus also changing the coeffcients $c_i(j)$. It can be concluded that the Farrow structure is not matched to an anti-aliasing task. It is matched to the sample rate before SRC, i.e. $1/T_1$. Hence it is well suited to maintain the original signal if no aliasing is to be expected from SRC, e.g. in the case of integer factor sample rate increase.

In general anti-aliasing is the most prominent task of a system for SRC. $T_2$ is the sample rate after SRC and thus determines the highest frequency component of the signal that can pass through the SRC process undistorted. Thus, the filter for SRC must depend on $T_2$ rather than on $T_1$. This constraint is served by the transposed Farrow structure. Regardless of the SRC factor, the filter implemented on the transposed Farrow structure is always related to the signal after decimation. Changing $T_2$ automatically adapts the filter characteristics to the new constraints. No coefficients $c_i(j)$ need to be changed.

From this short discussion is is clear that in general the transposed Farrow structure is the better choice. However, there are cases where the (original) Farrow structure is preferably

employed. If there are no signal components besides the signal-of-interest, all potential aliasing components are equal to the spectral repetitions of the signal-of-interest. Thus, anti-imaging might be the issue in this case for which the Farrow structure is perfectly suited. However, it must be ensured that the images are sufficiently attenuated to avoid aliasing.

The actual coefficients $c_i(j)$ of both the Farrow structure and the transposed Farrow structure can be obtained, e.g. by means of approximating a desired impulse response with the polynomial impulse response of Equation (80). Several solutions can be found in [24]. It should be noted that if both structures are realized with exactly the same coefficients, the impulse responses of the underlying filters are similar, i.e. they are identical except for a time stretch (and thus, a frequency compression) of one compared to the other.

Finally, it should be mentioned that the necessary multipliers of the two Farrow structures might be undesirable. For integer factor SRC there are filter classes that do not require multipliers. They can be obtained from sensible factorizations of certain transfer functions. Two widely known multiplier-free filters for integer factor SRC are cascaded comb filters [5] and cascaded integrator comb filters (CIC filters) [14]. However, the great advantage of low effort must be paid for with relatively poor transfer characteristics. Due to the small width of their stop-bands they can only be applied at high OSRs of the channel-of-interest. This is partially offset by the simple structure of these filters which supports an implementation at high sample rates (where the OSR of the signal can be kept high).

In order to use CIC filters also for rational factor SRC, the author has proposed a combination of a CIC interpolator and a CIC decimator that is implemented in a polyphase-like fashion [8,9]. In this case multipliers are required.

## 6.6 Example

### *6.6.1 Design Parameters*

At the end of this chapter a digital front end is designed step by step in order to illustrate the design process. From the previous sections it should have become clear that software radio is possible with respect to the digital front end, however, only within certain boundaries. These boundaries have to be defined first for the example design. Only the receive branch will be dealt with.

- The digital front end should be designed for a base station.
- The per-channel approach is used because of its flexibility (see Section 6.4.3). Hence, only one channel has to be designed (that is instantiated according to the required number of channels).
- The sample rate of the ADC is assumed to be state-of-the art, i.e. 80 MHz for a 14 bit quantizer.
- The analog front end performs I-Q down-conversion of a signal of very wide bandwidth (theoretically 80 MHz). Thus, a pair of matched ADCs sample the signal.

The tasks of the digital front-end are:

- digital down-conversion of the channel(s)-of-interest
- sample-rate conversion, and
- channel filtering.

Eventually, the design should serve the requirements of the GSM and the UMTS. Due to the large bandwidth that is defined by the sample rate of the ADC, several frequency channels of one or the other air interface are received simultaneously. For UMTS as a wide-band signal a necessary adjacent channel selectivity of 60 dB is assumed; for GSM this figure is set to 100 dB. It is further assumed that the signals at the output of the DFE should have twice the target rate $f_{\text{target}}$, i.e. the chip rate or bit rate, respectively (7.68 MHz for UMTS and 541.67 kHz for GSM). Thus, the design parameters have been set.

In order to be able to perform the filtering operations (for SRC and channel filtering) at baseband, the signal is converted to baseband first.

### *6.6.2 Digital Down-Conversion*

The channel-of-interest is digitally down-converted with a CORDIC-based down-converter. According to [18] a CORDIC with 16 iterations and 18 fractional binary digits to represent the intermediate results performs sufficiently. It achieves a spurious free dynamic range of approximately 98 dB.

In order to meet the speed requirements a pipelined CORDIC is favored, i.e. for each of the 16 iterations a separate CORDIC stage is realized. The stages are separated by registers.

### *6.6.3 Sample Rate Conversion*

In this example fractional SRC is placed at the high sample rate directly after digital down-conversion. This enables very simple filters (see Section 6.5.4). A first order polynomial filter (see Figure 6.27) provides enough aliasing attenuation. It should be noted that if the polynomial length was $T_1$ rather than $T_2$ the filter would be a linear interpolator with very poor anti-aliasing properties. The polynomial filter with polynomial length $T_2$ can be implemented on the transposed Farrow structure. It is sketched in Figure 6.28. Only one multiplier is required.

It should be recalled that the sample rate after fractional SRC is not less than half the sample rate before fractional SRC, i.e. it is not less than 40 MHz. Hence, the OSR of the signal-of-interest is still very high. This is exactly the reason why such simple filters can be employed.

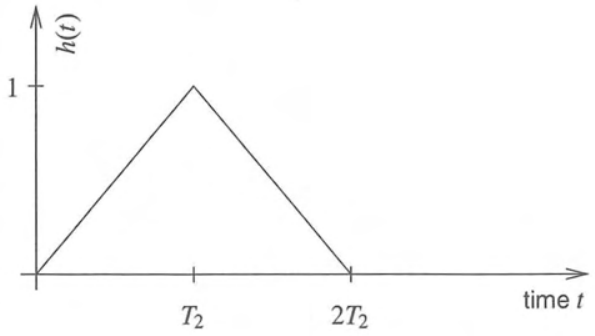


**Figure 6.27** Impulse response of a first-order polynomial filter

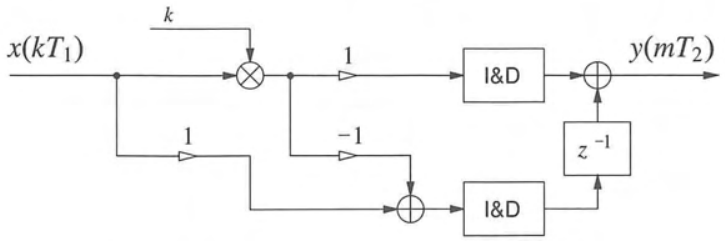


**Figure 6.28** Transposed Farrow structure implementing the impulse response of Figure 6.27, I&D stands for integrate-and-dump (Integrate with period $T_1$ and dump with period $T_2$)

For UMTS a feasible rate change factor is 24/25 and for GSM 13/15. Hence, in order to achieve the target sample rates, decimation by 10 must follow for UMTS, and by 128 for GSM. The respective decimation filters simultaneously perform channel filtering. Therefore they are discussed in the next section.

### *6.6.4 Channel Filtering*

The channel filter must remove all adjacent channels and possibly perform matched filtering. Due to the high oversampling ratio of the channel-of-interest at the input of the channel filter, multirate filtering is an efficient method. Thus channel filtering can be combined with integer-factor decimation.

As long as the oversampling ratio of the channel-of-interest is sufficiently large after decimation, simple filters, e.g. CIC filters can be used. A fourth-order CIC filter is chosen for this example. From the tables given in [14] it can be concluded that these filters can be used for decimating the GSM signal down to 8 times the bit rate, and the UMTS signal down to 4 times the chip rate.

The remaining decimation task is realized by a cascade of half-band filters. If perfect linearity is an issue, only FIR filters can be used. Otherwise IIR filters are also applicable. There are also attempts to realize approximately linear-phase IIR half-band filters [16]. Stability and effort issues on IIR filters can be tackled with the application of wave-digital filters (see [17] for an application, and [3] for fundamentals on wave digital filters).

If linear phase FIR filters are used, the half-band filters can be designed with any filter design tool. However, in order to reduce the effort, optimized half-band filters can be used that exploit the oversampling ratio of the signal, i.e. the half-band filter for decimating from 8 times the target rate to 4 times the target rate has different properties than the filter used for decimating from 4 times to 2 times the target rate (see Figure 6.15). Henker [7] has suggested a cascade of a half-band filter designed with Lagrange's formula [4] and a half-band filter proposed by Goodman and Carey [6].

The coefficients of a fifth-order Lagrange filter are [3, 0, −25, 0, 150, 256, 150, 0, −25, 0, 3]. They can be encoded with 9 bit fixed-point two's complement numbers. The filter achieves approximately 100 dB of aliasing attenuation in the band-of-interest. Since this filter can be employed for decimating from 8 times to 4 times the target rate, independent

from the air interface, the coefficients do not need to be changed and thus can be hard-wired with a few shift and add operations.

The coefficients of the second half-band filter (Goodman and Carey called it *F9* [6]) are [18, 0, −116, 0, 429, 0, −1277, 0, 5042, 8192, 5042, 0, −1277, 0, 429, 0, −116, 0, 18]. They can be coded with 14 bit fixed-point numbers (or with 13 bit if the impulse response is negated). The filter provides aliasing attenuation of approximately 79 dB in the frequency band-of-interest. This is sufficient at this stage of the multirate filter (see Figure 6.3 for the decreasing effort as the bandwidth decreases). As with the first half-band filter, the second can also be employed independent from the air interface and can thus be hard-wired.

Filtering that depends on the air interface in terms of transition-band characteristics (e.g. matched filtering) can be applied at twice the target rate, i.e. at the output of the suggested DFE. It is not included in this example. However, it should be mentioned that this filter can also be used to correct a possible pass-band droop introduced by the comb filters.

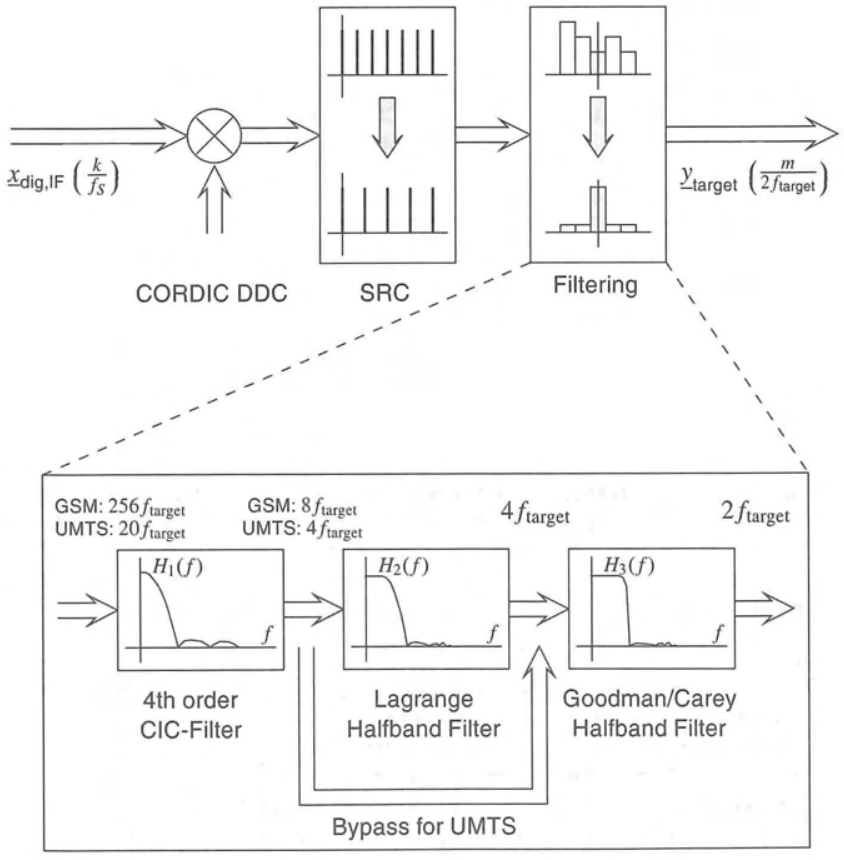


**Figure 6.29** Structure of the suggested DFE

### 6.6.5 Summary

The structure of the suggested DFE is shown in Figure 6.29. Although it looks rather hard-wired and fixed, it can indeed be employed in a software defined radio. The carrier frequency can be freely chosen with the CORDIC down-converter, likewise the sample rate conversion factor. Depending on the decimation factor of the CIC filters, the target rate can be tuned. This goes automatically with the necessary filtering. Possibly a final filter stage must be foreseen that performs, e.g. matched filtering.

The effort is summarized in Table 6.1 where only multiplications are counted. Since the processed signals are complex signals all filters must be doubled to have one in the I- and one in the Q-branch. It should be noted that the half-band filters can be clocked at the lower of the two sample rates involved (polyphase realization). If the half-band filters are realized hard-wired their contribution to the multiplication rate can be neglected, especially for the Lagrange half-band filter which has very simple coefficients. Therefore, the CORDIC and the transposed Farrow structure for SRC are the main contributors to the multiplication rate and thus, to the power consumption.

**Table 6.1** Hardware effort and computational effort of the suggested DFE

| | CORDIC | SRC | CIC (4th order) | Lagrange half-band filter | Second half-band filter |
|---|---|---|---|---|---|
| Number of general purpose multipliers | equiv. 3 | $2 \times 1$ | – | – | – |
| Number of coefficient multipliers | – | – | – | $2 \times 4$ | $2 \times 6$ |
| Running at | $f_S$ | $f_S$ | $f_S$, $8 \times f_{target}$ | $4 \times f_{target}$ | $2 \times f_{target}$ |
| for GSM | 80 MSps | 80 MSps | 80 MSps, ≈2.17 MSps | ≈1.08 MSps | ≈0.54 MSps |
| for UMTS | 80 MSps | 80 MSps | 80 MSps, ≈31 MSps | – | ≈7.7 MSps |
| Multiplication rate (mps) | $3 \times f_S$ | $2 \times f_S$ | – | $8 \times 4 \times f_{target}$ | $12 \times 2 \times f_{target}$ |
| for GSM | 240 Mmps | 160 Mmps | – | ≈8.7 Mmps | ≈6.5 Mmps |
| for UMTS | 240 Mmps | 160 Mmps | – | – | ≈92 Mmps |

## 6.7 Conclusion

This chapter has covered concepts of digital signal processing typically applied between the analog/digital interface and base-band processing. Special attention has been paid to the implementation feasibility in software programmable transceivers. Due to the high sample rates at which parts of the described signal processing steps must be performed, hard-wired implementations would be optimal solutions in terms of hardware effort and power consumption, if they were adaptable to the different air interfaces that a software radio must cope with. This is the motivation why the signal processing concepts (for digital down-conversion, channel-filtering, and sample rate conversion) are based on a fixed hardware platform that can be adapted to different requirements by means of parameterization. Thus, they are a

means to realize the functionalities of the digital front end of a software radio transceiver with low hardware effort and low power consumption.

The concept of the digital front end allows software radio to be implemented today.

## Acknowledgments

The authors would like to thank Matthias Henker and Michael Löhning from Dresden University of Technology for the numerous discussions and their invaluable comments which helped to considerably improve the chapter. This chapter is based upon research activities funded by the German Ministry of Science and Research (BMBF) and Alcatel Research and Innovation, Stuttgart, Germany.

## References

[1] Crochiere, R.E. and Rabiner, L.R., *Multirate Digital Signal Processing*, Prentice-Hall, 1983.

[2] Farrow, C.W., 'A continuously variable digital delay element', in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '88)*, Espoo, Finland, June, 1988, pp. 2641–2645.

[3] Fettweis, A., 'Wave digital filters: theory and practice', *Proceedings of the IEEE*, 1986.

[4] Fliege, N.J., *Multirate Digital Signal Processing: Multirate Systems, Filter Banks, Wavelets*, John Wiley & Sons, 1994.

[5] Frerking, M.E., *Digital Signal Processing in Communication Systems*. Van Nostrand Reinhold, 1994.

[6] Goodman, D.J. and Carey, M.J., 'Nine digital filters for decimation and interpolation', *IEEE Transactions on Acousics, Speech and Signal Processing*, ASSP-25, No. 2, April, 1977, pp. 121–126.

[7] Henker, M., Abtastratenanpassung in Software-programmierbaren Mobilfunkempfängern, Master's thesis, Technische Universität Dresden - Fakultät Elektrotechnik - Institut für Nachrichtentechnik, October, 1998.

[8] Henker, M., Hentschel, T. and Fettweis, G.P., 'Time-variant CIC-filters for sample-rate conversion with arbitrary rational factors', in *IEEE 6th International Conference on Electronics, Circuits and Systems (ICECS '99)*, Paphos, Cyprus, September, 1999, pp. 67–70.

[9] Hentschel, T., *Sample Rate Conversion for Software Radio*, Artech House, 2002, in press.

[10] Hentschel, T. and Fettweis, G., 'Continuous-time digital filters for sample-rate conversion in reconfigurable radio terminals', in *Proceedings of the European Wireless*, Dresden, Germany, September, 2000, VDE, VDE Verlag Berlin Offenbach, pp. 55–59.

[11] Hentschel, T. and Fettweis, G.P., 'AD conversion and channelization for multi-mode terminals', in *MTT-S European Wireless 98*, Amsterdam, Netherlands, October, 1998, pp. 169–174.

[12] Hentschel, T. and Fettweis, G.P., 'Sample rate conversion for software radio', *IEEE Communications Magazine*, Vol. 38, No. 8, August, 2000, pp. 142–150.

[13] Hentschel, T., Henker, M. and Fettweis, G.P., 'The digital front-end of software radio terminals', *IEEE Personal Communications*, Vol. 6, No. 4, August, 1999, pp. 40–46.

[14] Hogenauer, E.B., 'An economical class of digital filters for decimation and interpolation', *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-29, No. 2, April, 1981, pp. 155–162.

[15] Hu, X., Harber, R.G. and Bass, S.C., 'Expanding the range of convergence of the CORDIC algorithm', *IEEE Transactions on Computers*, Vol. 40, No. 1, January, 1991, pp. 13–21.

[16] Johansson, H. and Wanhammar, L., 'Filter structures composed of all-pass and FIR filters for interpolation and decimation by a factor of two', *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, Vol. 46, No. 7, July, 1999, pp. 896–905.

[17] Johansson, H. and Wanhammar, L., 'Wave digital filter structures for high-speed narrow-band and wide-band filtering', *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, Vol. 46, No. 6, June, 1999, pp. 726–741.

[18] Löhning, M., Hentschel, T. and Fettweis, G.P., 'Digital down conversion in software radio terminals', in *Proceedings of the 10th European Signal Processing Conference (EUSIPCO)*, Vol. 3, Tampere, Finland, September, 2000, pp. 1517–1520.

[19] Meyr, H., Moeneclaey, M. and Fechtel, S.A., *Digital Communication Receivers: Synchronization, Channel Estimation and Signal Processing*, Wiley series in telecommunications and signal processing. John Wiley & Sons, New York, 1998.
[20] Proakis, J.G. and Manolakis, D.G., *Digital Signal Processing*, 3rd Edition, Prentice-Hall, 1996.
[21] Reng, R.L., 'Polyphase and modulation descriptions of multirate systems - a systematic approach', in *International Conference on Digital Signal Processing*, Limassol, Cyprus, June, 1995, pp. 212–217.
[22] Schafer, R.W. and Rabiner, L.R., 'A digital signal processing approach to interpolation', *Proceedings of the IEEE*, Vol. 61, No. 6, June, 1973, pp. 692–702.
[23] Vaidyanathan, P.P., *Multirate Systems and Filter Banks*. Prentice Hall PTR, Englewood Cliffs, NJ, 1993.
[24] Vesma, J., Optimization and applications of polynomial-based interpolation filters, PhD thesis, Tampere University of Technology, Tampere, Finland, May, 1999.
[25] Volder, J.E., 'The CORDIC trigonometric computing technique', *IRE Transactions on Electronic Computers*, EC-8, September, 1959, pp. 330–334.
[26] Zangi, K.C. and Koilpillai, R.D., 'Software radio issues in cellular base stations', *IEEE Jounal on Selected Areas in Communications*, JSAC-17, No. 4, April, 1999, pp. 561–573.

# software defined radio

## Enabling Technologies

Edited by Walter Tuttlebee
*Mobile VCE, UK*

Software defined radio (SDR) is one of the most important topics of research, and indeed development, in the area of mobile and personal communications. SDR is viewed as an enabler of global roaming and as a unique platform for the rapid introduction of new services into existing live networks. It therefore promises mobile communication networks a major increase in flexibility and capability. SDR brings together two key technologies of the last decade – digital radio and downloadable software. It encompasses not only reconfiguration of the air interface parameters of handset and basestation products but also the whole mobile network, to facilitate the dynamic introduction of new functionality and mass-customised applications to the user's terminal, post-purchase.

Contributed by internationally respected industry practitioners and researchers this book provides a holistic treatment of SDR addressing the full breadth of relevant technologies – radio frequency design, data conversion, reconfigurable signal processing hardware, and software issues at all levels of the protocol stack and network. As such it provides a solid grounding for a new generation of wireless engineers for whom radio design in the future will assume dynamic flexibility as a given.

In particular it explores

- the unique demands of SDR upon the RF subsystem and their implications for front end design methodologies
- the recent concepts of the 'digital front end' and 'parametrization'
- the role and key influence of data conversion technologies and devices within software radio, essential to robust product design
- the evolution of signal processing technologies, describing new architectural approaches
- management of terminal reconfiguration and its network implications
- the concepts of the waveform description language
- potential breakthrough technologies, such as superconducting RSFQ technology and the possible future role of MEMS in RF circuitry
- competing approaches, eg all-software radios implemented on commodity computing vs advanced processing architectures that dynamically optimise their configuration to match the algorithm requirements at a point in time

Suitable for today's engineers, technical staff and researchers within the wireless industry, the book will also appeal to marketing and commercial managers who need to understand the basics and potential of the technology for future product development. Its balance of industrial and academic contributors also makes it suitable as a text for graduate and postgraduate courses aiming to prepare the next generation of wireless engineers.

ISBN 0-470-84318-7
9 780470 843185