

# ANALOG FRONT-ENDS FOR SOFTWARE-DEFINED RADIO RECEIVERS

Vincent Arkesteijn

Samenstelling promotiecommissie:

voorzitter:	prof.dr.ir. A.J. Mouthaan	Universiteit Twente
secretaris:	prof.dr.ir. A.J. Mouthaan	Universiteit Twente
promotor:	prof.dr.ir. B. Nauta	Universiteit Twente
assistent-promotor:	dr.ing. E.A.M. Klumperink	Universiteit Twente
leden:	prof.dr.ir. P.G.M. Baltus	TU Eindhoven
	prof.dr.ir. J.C. Haartsen	Universiteit Twente
	prof.dr.ir. C.H. Slump	Universiteit Twente
	prof.ir. A.J.M. van Tuijl	Universiteit Twente

© 2007, Vincent Arkesteijn  
All rights reserved.

ISBN 978-90-365-2562-6  
ISSN 1381-3617 (CTIT Ph.D.-thesis series No. 07-101)

This research is supported by the Program for Research on Embedded Systems & Software (PROGRESS) of the Dutch organization for Scientific Research NWO, the Dutch Ministry of Economic Affairs and the technology foundation STW, under grant number 05177.

# ANALOG FRONT-ENDS FOR SOFTWARE-DEFINED RADIO RECEIVERS

PROEFSCHRIFT

ter verkrijging van  
de graad van doctor aan de Universiteit Twente,  
op gezag van de rector magnificus,  
prof.dr. W.H.M. Zijm,  
volgens besluit van het College voor Promoties  
in het openbaar te verdedigen  
op vrijdag 14 september 2007 om 16.45 uur

door

Vincent Jan Arkesteijn  
geboren op 2 juli 1975  
te Rotterdam

Dit proefschrift is goedgekeurd door

de promotor prof.dr.ir. B. Nauta

de assistent-promotor dr.ing. E.A.M. Klumperink

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Software (Defined) Radio . . . . .	2
1.3	Project description . . . . .	3
1.4	Previous work . . . . .	4
1.5	Thesis Outline . . . . .	5
<b>2</b>	<b>Software Radio</b>	<b>7</b>
2.1	Sampler-based architectures . . . . .	7
2.1.1	Software Radio . . . . .	8
2.1.2	Direct RF Sampling . . . . .	8
2.2	Requirements . . . . .	9
2.2.1	Sample rate . . . . .	9
2.2.2	SNR . . . . .	10
2.2.3	Jitter . . . . .	12
2.2.4	SFDR . . . . .	22
2.3	Feasibility . . . . .	23
2.4	Summary and Conclusions . . . . .	24
<b>3</b>	<b>Software Defined Radio</b>	<b>25</b>
3.1	Downconverter without RF pre-filtering . . . . .	26
3.1.1	Noise requirement . . . . .	26
3.1.2	Linearity requirement . . . . .	27
3.1.3	Conclusion . . . . .	32
3.2	Downconverter with RF pre-filtering . . . . .	33
3.2.1	Linearity requirement . . . . .	33
3.2.2	Selection of intermediate frequency . . . . .	35
3.3	RF Filtering . . . . .	44
3.4	Local Oscillator . . . . .	45

3.4.1	Jitter . . . . .	45
3.4.2	Sampling and mixing compared . . . . .	46
3.5	Channel filtering and analogue-to-digital conversion . . .	48
3.5.1	Sample rate . . . . .	50
3.5.2	SFDR . . . . .	50
3.5.3	SNR . . . . .	52
3.5.4	Resolution . . . . .	52
3.5.5	Variable gain . . . . .	55
3.5.6	Conclusion . . . . .	57
3.6	Summary and conclusions . . . . .	57
<b>4</b>	<b>A wideband downconverter</b>	<b>59</b>
4.1	Design considerations . . . . .	60
4.2	Circuit overview . . . . .	61
4.3	Low-Noise Amplifier . . . . .	61
4.4	Mixer . . . . .	67
4.5	IF filter and amplifier . . . . .	69
4.6	Simulations . . . . .	69
4.7	Measurements . . . . .	71
4.8	Combination with postprocessing . . . . .	79
4.8.1	Software . . . . .	79
4.8.2	Hardware . . . . .	80
4.8.3	Experimental results . . . . .	84
4.9	Conclusions . . . . .	86
4.10	Discussion . . . . .	87
<b>5</b>	<b>Summary and conclusions</b>	<b>91</b>
5.1	Summary . . . . .	91
5.2	Conclusions . . . . .	92
5.3	Recommendations for further research . . . . .	92
	<b>Samenvatting</b>	<b>93</b>
	<b>Dankwoord</b>	<b>95</b>
<b>A</b>	<b>Intermodulation distortion</b>	<b>97</b>
A.1	Frequencies . . . . .	97
A.1.1	Second order . . . . .	97
A.1.2	Third order . . . . .	98
A.2	Levels . . . . .	98

---

<b>Bibliography</b>	<b>101</b>
<b>List of publications</b>	<b>107</b>





# Chapter 1

## Introduction

### 1.1 Motivation

After its introduction in the late 19<sup>th</sup> century, radio communication has seen increasing use. Not only its use has increased, but also the number of modulation formats and the frequency range that is being used. In recent years, the proliferation of radio standards has progressed even faster. This has led to a plethora of standards, such as GSM and WCDMA for cellular communication, Hiperlan/2 and IEEE 802.11a/b/g for wireless networking, and many more. All these standards have different modulation types, bandwidths, and carrier frequencies. For illustration, figure 1.1 shows the frequency bands that some of these standards use.

The increasing number of standards leads to an increasing number of different radios, which is unwieldy for both manufacturers and con-

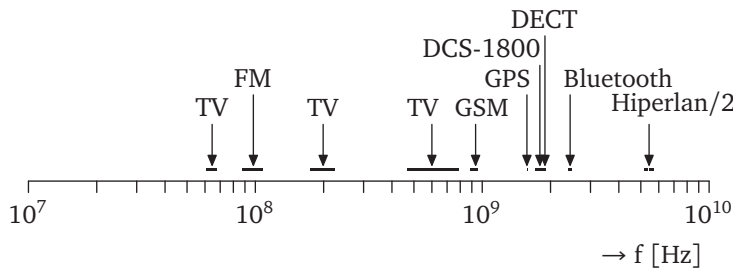


Figure 1.1: part of the available radio spectrum and some commercially relevant uses thereof

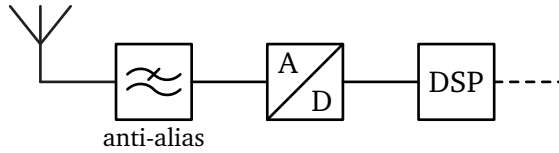


Figure 1.2: an example of a software radio receiver front-end

sumers. Manufacturers need to spread their R&D efforts on a large set of hardware platforms. Furthermore, they need to manufacture, stock and distribute many different radios. Consumers need to buy and carry a different radio for every application they expect to use, and often some more if they intend to visit other regions.

A solution would be a more flexible, reconfigurable radio, which can support many different standards. Such a radio would also have advantages in dynamic environments, where it can adapt to for instance varying channel parameters, or varying levels of interferers.

## 1.2 Software (Defined) Radio

The holy grail of flexible radio receivers is the ‘Software Radio’, sometimes also referred to as ‘Ideal’ or ‘True’ Software Radio. As its name suggests, all of the radio functionality –such as channel selection and demodulation– is implemented in software. This form was described by Mitola in [1]. Assuming that software is flexible or at least replaceable, this approach yields a flexible receiver.

Software runs on digital hardware, and because signals at the antenna interface are analogue, an analogue-to-digital converter (ADC) has to be included as well. A general block schematic of a software radio receiver front-end is shown in figure 1.2.

As will be shown in chapter 2, a true software radio is hard to implement and will likely remain utopian for some time to come. A solution that is more feasible in the near future is a radio receiver where part of the flexibility is achieved by flexible analogue hardware instead of by software. This is called a ‘Software Defined Radio’ (SDR). Another way to view this is that an analogue front-end conditions the antenna signal before it is converted to digital.

### 1.3 Project description

The research described in this dissertation was performed within the project 'Development of a software-radio based embedded mobile terminal'. The goal was to assess feasibility of and develop a mobile software-defined radio terminal. This should lead to a radio which can support multiple standards, and which can easily be updated when the need arises.

The scope was further limited to a receiver, and specifically to the parts of the receiver starting directly after the antenna and up to the demodulator.

This dissertation describes the research into the part of the receiver up to and including the ADC. The research on the digital part of the receiver, which mostly covers channel selection and demodulation, is described in Roel Schiphorst's dissertation [2].

The term 'mobile terminal' covers many types of devices, from a cellular phone to a vehicle-based transceiver. Because power constraints in a cellular phone or a PDA are likely too limiting for a software-defined radio approach, a laptop was chosen as a platform. Not only is more power available, but laptops already have a fast processor, which can be used for software radio signal processing.

As stated earlier, the aim is to design and analyze receivers capable of receiving signals of *any* standard. This is a very broad goal, and therefore the scope has been limited further by selecting two specific but very different standards. These are Bluetooth [3] and Hiperlan/2 [4]. As can be seen in table 1.1, these standards have very different parameters. Hopefully, by choosing two dissimilar standards, not too much generality will be lost and conclusions drawn from experiments with these two standards can be generalised to a larger class of problems. Most importantly, the used frequency bands are quite far apart. This is also shown in figure 1.1.

Yet another choice to be made is the implementation technology. Although technologically advanced processes such as BiCMOS, GaAs, SiGe have clear advantages, especially at higher frequencies, this dissertation focuses on CMOS. First, this is because of advantages such as low cost, portability between fabs, et cetera that are common to most applications. More important in the context of a software defined radio receiver however, is the possibility of monolithic integration with the digital parts

	Bluetooth	Hiperlan/2
frequency band [GHz]	2.4–2.48	5.15–5.725 <sup>1</sup>
channel bandwidth <sup>2</sup> [MHz]	~0.6	16
channel spacing [MHz]	1	20
nominal bit rate [Mb/s]	1	6–54
modulation type	GFSK	QAM + OFDM
multiple access scheme	FHSS	TDMA
duplex scheme	TDD	TDD

Table 1.1: key characteristics of Bluetooth [3] and Hiperlan/2[4]

of the receiver.

## 1.4 Previous work

Naturally, design of radio receivers is as old as radio itself. Although early radio receivers didn't have much selectivity and therefore could be considered wideband, they are not very useful on the present crowded radio spectrum.

Software (defined) radio receivers are a more recent phenomenon. However, at the start of our project already much research has been done in the area. Many projects investigated the software needed for software defined radio, while some had researched the flexible digital hardware needed, e.g. [5].

Few however, included RF hardware in their research. One software radio is described in [6]. In this HF transceiver, the entire spectrum from 0 to 31 MHz is digitised. Another project is the military Speakeasy project [7, 8], where the objective is to cover the entire band from 2 MHz to 2 GHz.

A software defined radio at higher frequencies is described in [9]. Here, signals in both the 1.5 and 1.9 GHz bands are downconverted and digitised. Some other projects had started, like SUNBEAM, SODERA, PROMURA and FIRST. However, no tangible results were available

To the best of our knowledge, no CMOS integrated software-defined radio receivers had been published at the start of this research in September 2000.

---

<sup>1</sup>with a gap between between 5.35 and 5.47 GHz

<sup>2</sup>Many definitions of bandwidth exist and these numbers are not meant to define the bandwidth exactly, but merely to illustrate the difference between the two standards.

## 1.5 Thesis Outline

After the introduction in this chapter, chapter 2 discusses software radio. The ADC requirements will be derived, with most attention to sampling clock jitter requirements.

In chapter 3, software defined radio is discussed. Front-end requirements such as noise figure and intermodulation intercept points will be derived in relation to the amount of RF pre-filtering. This way, a trade-off can be made. Furthermore, requirements on the IF ADC will be derived in relation to the amount of IF filtering.

In chapter 4, the design of a wideband integrated downconverter is discussed. The design has been realised in 0.18  $\mu\text{m}$  CMOS. In addition to measurement results of the downconverter, test results of the downconverter in combination with digital channel selection and demodulation are presented.

And finally, chapter 5 summarizes the conclusions, discusses the results and gives recommendations for further research.



## Chapter 2

# Software Radio

As software is flexible or at least replaceable, a receiver architecture where all signal processing is done in software, will yield much flexibility. A highly digital CMOS receiver implementation also has all the advantages of digital design (easier/faster development, benefits from Moore's law, easy porting to newer technology). Therefore, such an architecture, called 'software radio' seems very attractive. It will be discussed in this chapter.

In addition to this software radio, another architecture, also employing direct RF sampling but without amplitude quantisation will be discussed. This is mainly done because many of its properties and requirements correspond to those of the software radio architecture.

The question that this chapter tries to answer is whether these architectures are feasible given today's available technology. Therefore, after introducing the architectures, requirements on their implementations will be discussed. These requirements are calculated for a receiver capable of receiving both Bluetooth and Hiperlan/2. Finally these requirements are compared to the performance of currently available components and from this conclusions are drawn.

### 2.1 Sampler-based architectures

This section introduces the two architectures discussed in this chapter.

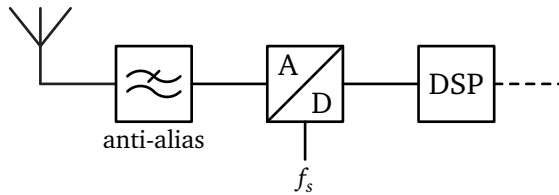


Figure 2.1: architecture of a software radio receiver front-end

### 2.1.1 Software Radio

A true software radio front-end is shown in figure 2.1. The incoming signal is first filtered by a lowpass filter. The filter's only function in this architecture is anti-aliasing, so all signals up to the filter's cut-off frequency will be present at the ADC input.

After filtering, the signal is sampled and quantised by the ADC. To avoid aliasing, the sample rate  $f_s$  is higher than twice the filter's cut-off frequency. All further processing is done on some kind of programmable digital hardware, here indicated with 'DSP'. The programmability of this digital hardware yields the desired flexibility.

The ADC in this diagram is the block whose requirements and feasibility will be discussed in this chapter.

The anti-alias filter in this receiver will be assumed to be a brickwall filter. This cannot be realized, so in practice the requirements on the ADC will be even higher than calculated here.

The requirements on and feasibility of the digital hardware are outside the scope of this thesis, but are discussed in [2] instead.

### 2.1.2 Direct RF Sampling

Another architecture, which was recently proposed [10] also employs direct RF sampling, but without amplitude quantisation. Another difference with the architecture in the previous section, is that here the sample rate  $f_s$  is usually equal to the frequency of the wanted signal  $f_{RF}$ . This way, the signal is downconverted by the sampling process. In that respect it is comparable to a zero-IF receiver, but here the RF signal is sampled, and subsequent processing is done in discrete time.

This architecture scales well to smaller processes. One reason to mention this architecture is its potential flexibility [11]. However, present



implementations of this architecture don't give very flexible radio receivers. Some reasons for this are given in [12].

From that perspective, this architecture does not fit in very well with this chapter. However, part of the requirements that will be derived in the next section are equally valid for this architecture. The fact that RF sampling receivers are feasible therefore implies that at least some of the requirements for software radio receivers can be met. Note however, that these receivers are preceded by a bandpass filter, which greatly relaxes some of the requirements.

## 2.2 Requirements

When implementing the above architectures, limited performance of the building blocks limits receiver performance. Minimum performance requirements therefore impose (combined) requirements on these building blocks. This section discusses the most important requirements for the software radio architecture of figure 2.1.

To assess the feasibility of a software radio receiver, all the requirements will be calculated. This will be done for a receiver capable of receiving both Bluetooth and Hiperlan/2, as described in section 1.3.

Of the requirements, most attention is given to jitter (section 2.2.3). Most of that section was previously published in [13] and [14].

### 2.2.1 Sample rate

According to the Nyquist-Shannon sampling theorem, a signal of finite bandwidth can be reconstructed exactly from its sampled version, if the sampling rate is higher than twice the bandwidth[15]. Our objective is not perfect reconstruction, but merely demodulation and/or detection of one desired signal, so conceivably some other criterion could be found. However, this criterion would depend on the demodulator, which depends on the signal of interest, and in order not to limit the receiver to specific types of signals, the sampling theorem is still assumed to give a necessary condition.

Therefore, assuming a low-pass AD converter, the sampling rate of the receiver should at least equal twice the highest frequency present at the input. Since the cut-off frequency of the anti-alias filter cannot be lower than the maximum signal frequency, this means the sample rate should be at least twice the maximum signal frequency.

### Example

The maximum frequency of interest for a combined Bluetooth and Hiperlan/2 receiver is 5.725 GHz. Therefore, this will be the cut-off frequency of the anti-alias filter, and the sampling rate should be higher than  $2 \times 5.725 = 11.45$  GHz.

### 2.2.2 SNR

Signal-to-noise ratio (SNR) can both be a property of a signal and of a system. When used to describe a system, it usually refers to the maximum SNR of the output signal of the system. This is also the case with analogue-to-digital converters:

Signal-to-noise ratio (SNR) is the ratio of the signal power to the total noise power at the output (usually measured for a sinusoidal input). [16]

As the output signal is digital, power in this context is not physical power, but can be interpreted as signal variance. For deriving this requirement, both the signal power and the maximum allowed total noise power have to be known.

When one strong interferer is present, this dominates the input signal power. Therefore, the maximum input power equals the maximum blocker power within the bandwidth of the filters preceding the ADC.

The noise power at the output of the converter should be low enough not to interfere with demodulation of the signal. Most standards do not specify the maximum noise level, but instead specify the bit (or frame) error rate at some input sensitivity level. Therefore, derivation of the noise floor requires selecting and often simulating a demodulation algorithm. This derivation is outside the scope of this thesis, but results from literature can be used.

### Example

As can be seen in the blocker specification in figure 2.5 on page 18, the maximum input signal is 0 dBm for Hiperlan/2 and -10 dBm for Bluetooth.

Calculating the maximum noise level is somewhat more involved as this is not specified in the standards. For this, both the minimum sensitivity level for which some bit (or frame) error rate is to be achieved, and the SNR at which the demodulator achieves this error rate have to

standard	sensitivity (dBm)	required in-band SNR (dB)	max in-band noise (dBm)	noise floor (dBm/Hz)
HL/2 (6 Mb/s)	-85	6.2	-91.2	-163.3
HL/2 (9 Mb/s)	-83	7.4	-90.4	-162.5
HL/2 (12 Mb/s)	-81	8.9	-89.9	-162.0
HL/2 (18 Mb/s)	-79	10.9	-89.9	-162.0
HL/2 (27 Mb/s)	-75	14.5	-89.5	-161.6
HL/2 (36 Mb/s)	-73	17.0	-90.0	-162.1
HL/2 (54 Mb/s)	-68	22.3	-90.3	-162.4
Bluetooth	-70	18.0	-88.0	-148.0

Table 2.1: Calculation of the maximum noise floor of the combined receiver, for all (sub-)standards. (Hiperlan/2 is abbreviated as HL/2)

be known. The standards specify the sensitivity level. It is shown in the second column in table 2.1.

The required SNR depends on the demodulator. The demodulator used in conjunction with the front-end described in this thesis, has been described in [2]. The numbers in the third column, ‘required in-band SNR’ were taken from there. This is the required SNR for the demodulator’s input signal, and should not be confused with the ADC’s SNR.

The maximum in-band noise (fourth column) is now calculated by subtracting the required SNR from the sensitivity level. The noise floor (fifth column) is the corresponding power spectral density. For this a bandwidth of 1 MHz was used for Bluetooth, and 16.5625 MHz for Hiperlan/2.

The lowest noise floor is for a 6 Mb/s Hiperlan/2 signal: -163.3 dBm/Hz. In the 5.7 GHz bandwidth of the receiver, this corresponds to a total noise of -65.7 dBm.

Therefore, the required SNR for the converter is 0 dBm – -65.7 dBm = 65.7 dB.

To convert this to a resolution, the waveform of the strong out-of-band interferers that dominate the input signal needs to be known. The Bluetooth standard specifies this to be a sine wave. The Hiperlan/2 standard does not specify this, but we will assume the same. Then, we can convert this SNR (in dB) to a number of bits  $n$  using  $SNR = 6.02 \cdot n + 1.76$ . Therefore, if we allow all the noise to be caused by quantisation, the required resolution of the converter is 11 bits.

### 2.2.3 Jitter

The ADC in the software radio architecture from figure 2.1 requires a clock source, as shown in figure 2.2. Jitter in this sampling clock leads to uncertainty in the sampling instant, and thus to an error in the sampled signal. See figure 2.3.

As will be shown in the section ‘white-noise model’, according to a commonly used model for white ADC clock jitter [17], the resolution directly affects the clock jitter requirements, resulting in very stringent numbers. Better jitter models are available in the literature, e.g. [18, 19, 20]. Here, the jitter spectrum is still considered white, resulting in a white error spectrum at the output of the ADC. In [21], spectra of input signal and jitter are taken into account, but only the error signal over the full Nyquist bandwidth is considered. For a radio receiver however, only the error signal in the channel bandwidth is relevant. Therefore, [18, 19, 20, 21] yield too pessimistic jitter requirements.

The section ‘coloured noise model’ will show a model comparable to that in [18, 19, 20, 21]. We show that using this model, taking into account the spectra of both the input signals and the sampling clock jitter, and looking at the jitter-induced error signal only in the frequency band of interest, sampling clock jitter requirements can be relaxed greatly.

#### White noise model

Consider an incoming signal  $s(t)$ . Ideally, the sampled version of this signal with sample rate  $1/\tau$ ,  $s_\tau(k)$ , is constructed as follows.

$$s_{\tau,\text{ideal}}(k) = s(k\tau) \quad (2.1)$$

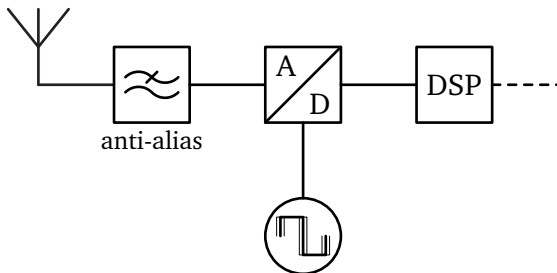


Figure 2.2: architecture of a software radio receiver front-end, with sampling clock source

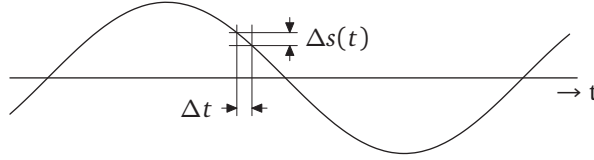


Figure 2.3: An input signal as a function of time and the effect of sampling jitter

Due to sampling jitter however, an error will be introduced, as can be seen in figure 2.3. The sampled signal can now be calculated as follows (for small absolute jitter  $\Delta t$ ).

$$\begin{aligned}
 s_{\tau}(k) &= s(k\tau + \Delta t(k\tau)) \\
 &\approx s(k\tau) + \underbrace{\Delta t(k\tau) \cdot \left. \frac{\partial}{\partial t} s(t) \right|_{k\tau}}_{\Delta s_{\tau}(k)}
 \end{aligned} \tag{2.2}$$

This signal consists of a sampled version of the input signal  $s(t)$  plus an error signal  $\Delta s_{\tau}(k)$  due to jitter. When requiring the RMS value of  $\Delta s_{\tau}(k)$  to be lower than the RMS error due to quantisation, the following relation between required RMS jitter (the RMS value of the absolute jitter  $\Delta t(k\tau)$ ) and resolution can be derived [22].

$$\Delta t_{rms} = \frac{2^{-n}}{\pi f_{max} \sqrt{6}} \tag{2.3}$$

Here, a full swing harmonic input signal at the maximum input frequency  $f_{max}$  is assumed.  $n$  is the resolution of the ADC.

Using this equation, the required RMS jitter can be calculated for a given resolution. For software radio applications this yields clock jitter requirements that are not achievable with currently available (integrated) clock sources, as will be shown in the comparison later on.

### Coloured noise model

The preceding analysis only derives the RMS value of the jitter induced error. This is often sufficient, especially when the signals of interest are wideband. In the case of a software radio receiver however, only a narrow portion of the converted bandwidth is of interest, and the spectral distribution of the error signal  $\Delta s_{\tau}(k)$  becomes relevant.

To derive the spectrum of the error signal, consider the following. As seen in equation 2.2, the error signal  $\Delta s_\tau(k)$  is the time derivative of the input signal  $\frac{\partial}{\partial t}s(t)$  multiplied with the sampling time error  $\Delta t(k\tau)$ :

$$\begin{aligned}\Delta s_\tau(k) &= s(k\tau + \Delta t(k\tau)) - s(k\tau) \\ &\approx \Delta t(k\tau) \cdot \left. \frac{\partial}{\partial t}s(t) \right|_{k\tau}\end{aligned}\quad (2.4)$$

Taking the discrete-time Fourier transform (DTFT) of both sides:

$$\mathcal{F}(\Delta s_\tau(k)) \approx \mathcal{F}(\Delta t(k\tau)) * \mathcal{F}\left(\left. \frac{\partial}{\partial t}s(t) \right|_{k\tau}\right)\quad (2.5)$$

where  $\mathcal{F}$  denotes the DTFT and  $*$  denotes convolution. This result is also obtained in [18, 23]. Apparently, for calculating the spectrum of the error signal, both the input signal and the spectrum of the jitter have to be known.

The input spectrum of a radio receiver is not known in general, but wireless communication standards normally limit the power levels of interfering signals that have to be tolerated at different frequencies. Figure 2.5 shows the blocking levels for Bluetooth [3] and Hiperlan/2 [4]. These blocking levels form an upper limit to the input signal  $s(t)$ . From this power spectrum, an upper limit to  $\frac{\partial}{\partial t}s(t)$  can be calculated by multiplying the corresponding amplitudes by  $j\omega$ . Note that interfering signals that are close in frequency to the wanted signal have far lower maximum power levels than those further away. We will see later that this greatly relaxes phase noise requirements.

Furthermore, the spectrum of  $\Delta t(t)$  has to be known. When the sampling clock is derived from a synthesizer containing a VCO (e.g. an LC or ring oscillator),  $\Delta t(t)$  can be assumed to have a  $f^{-2}$  power spectrum outside the synthesizer loop bandwidth [24]. An example of such a spectrum can be seen in figure 2.7. Above the synthesizer loop bandwidth (which is 100 kHz in this example), the phase noise of the synthesizer is dominated by that of the VCO.

Also, because variance of the absolute jitter at the output of a first or second-order PLL is constant when the observation time exceeds the loop time constant [25], the jitter process is assumed to be stationary.

The effect of applying equation 2.5 to the input and jitter spectra described above can best be illustrated with the example of an interfering

harmonic input signal  $s(t) = A_i \sin(\omega_i t)$ :

$$\begin{aligned} \mathcal{F}(\Delta s_\tau(k)) &\approx \mathcal{F}(\Delta t(k\tau)) * \mathcal{F}\left(\left.\frac{\partial}{\partial t} A_i \sin(\omega_i t)\right|_{k\tau}\right) \\ &= \mathcal{F}(\Delta t(k\tau)) * \omega_i A_i \cdot \mathcal{F}(\cos(\omega_i k\tau)) \end{aligned} \quad (2.6)$$

Due to its  $f^{-2}$  nature, most energy in  $\Delta t(k\tau)$  is at low frequencies. Knowing that in the frequency domain this is convolved with the derivative of the input signal leads to the following.

1. The convolution operation in equation 2.5 shifts the jitter spectrum  $\mathcal{F}(\Delta t(k\tau))$  to the frequencies of input signals. Therefore, the jitter-induced error in the output is concentrated around these frequencies.
2. Input signals with higher power are surrounded by more jitter-induced error in the output than input signals with lower power, due to the linearity of the convolution operation.
3. Input signals of higher frequencies are surrounded by more jitter-induced error in the output than signals at lower frequencies, because of the frequency dependent effect of  $\frac{\delta}{\delta t}s(t)$ . This is in accordance with the results in [20].

The above is illustrated in figure 2.4. There, the effects described under 1 and 2 are clearly visible: the jitter-induced error in the output is concentrated around the input frequencies, and it has more power around input signals with higher power. To compare the jitter-induced error spectrum with the one predicted by a white-noise model, a flat line has been drawn, indicating a white spectrum with the same RMS value as the actual (coloured) noise spectrum.

Because the jitter-induced output error is concentrated around the frequencies with the strongest input signals, it is less of a problem in the frequency band of interest. This is further illustrated in the example of the next section.

## Comparison

To illustrate the significance of the difference between the two ADC models, a numeric example will be given.

The combined receiver for Bluetooth and Hiperlan/2 requires a bandwidth of 6 GHz and a resolution of 11 bits (sections 2.2.1 and 2.2.2).

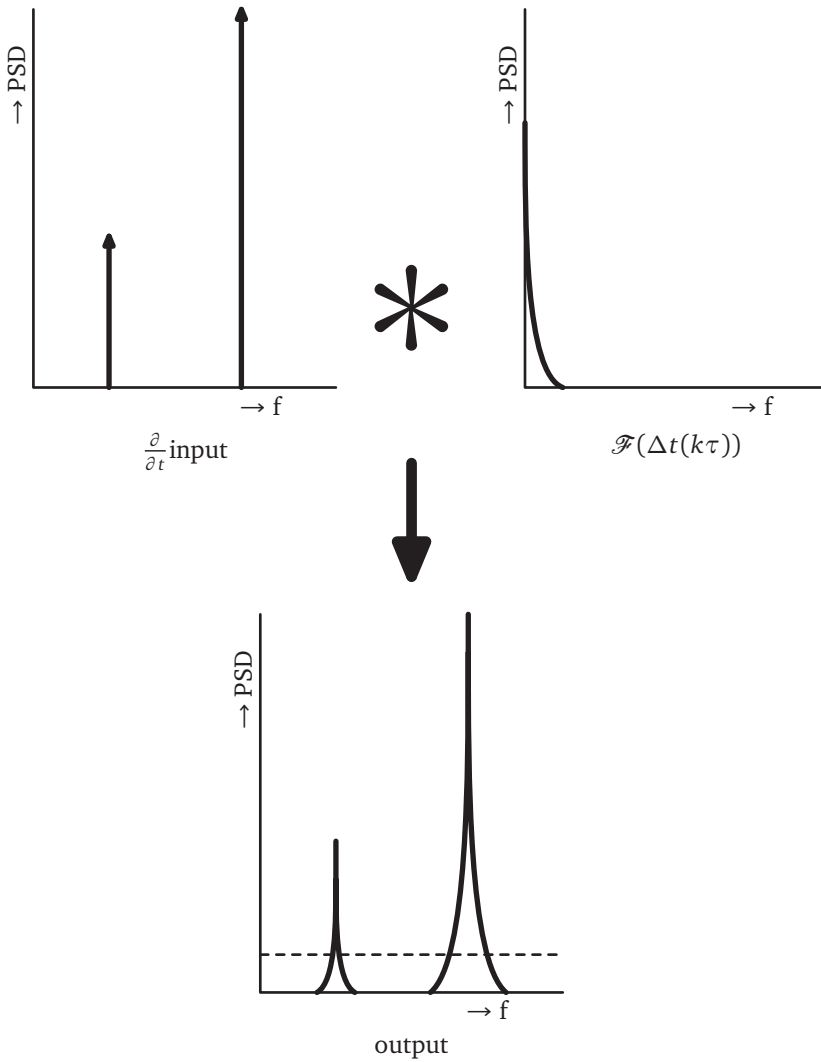


Figure 2.4: Illustration of equation 2.6. The convolution of the input spectrum (upper left) with the spectrum of  $\Delta t(k\tau)$  (upper right) gives the output spectrum of the ADC (bottom). For comparison purposes, the dashed line represents a white error spectrum with the same RMS value as the actual spectrum.



Using these numbers in equation 2.3 results in

$$\Delta t_{rms} = \frac{2^{-11}}{\pi \cdot 6G \cdot \sqrt{6}} \approx 11 \text{ [fs]}$$

This is one or two orders of magnitude smaller than what is achieved by currently published integrated synthesizers. For instance, a state-of-the-art system as described in [26] reports 0.22 ps RMS jitter.

If we use the more realistic ADC model however, results are different. Figure 2.6 shows the output spectrum of an ADC, with four different interfering input signals. The levels of these signals were taken to be the blocking levels as shown in figure 2.5, at the frequencies where their impact is most severe (2.4, 5.06, 7.1 and 12.98 GHz). None of these interferers cause the jitter-induced output error to exceed the maximum allowed in-band noise level (indicated by the shaded area).

To improve readability of this figure, in-band interferers have been left out. Further analysis showed that the jitter-induced error due to in-band interferers is just below the maximum in-band noise level.

The RMS jitter of the 12 GHz sampling clock used for figure 2.6 was 1.3 ps, with a flat power spectrum up to 100 kHz from the carrier and a  $f^{-2}$  roll-off above that. This corresponds to  $-96$  dBc/Hz phase noise at 1 MHz offset. These values have been chosen to just comply with in-band noise requirements. This spectrum is shown in figure 2.7.

To assess the feasibility of an oscillator with this phase noise performance, the ‘oscillator number’ figure of merit (FoM) as defined in [27] as

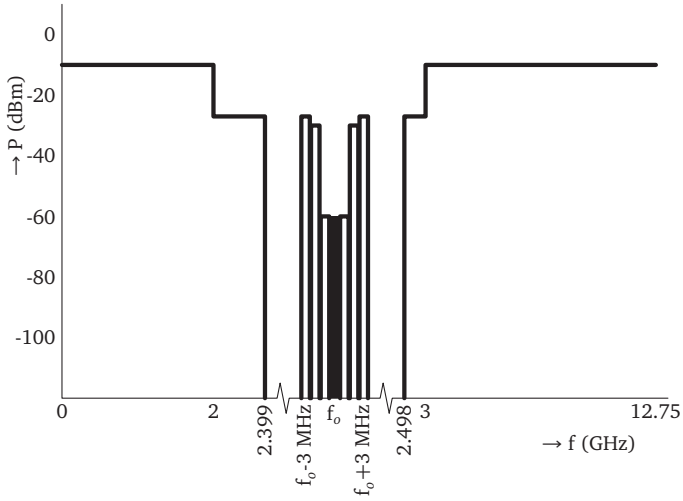
$$\text{FOM}_{\text{Osc-No}} = 10 \log(\mathcal{L}(f_m)) + 20 \log\left(\frac{f_m}{f_{osc}}\right)$$

For the numbers given above:

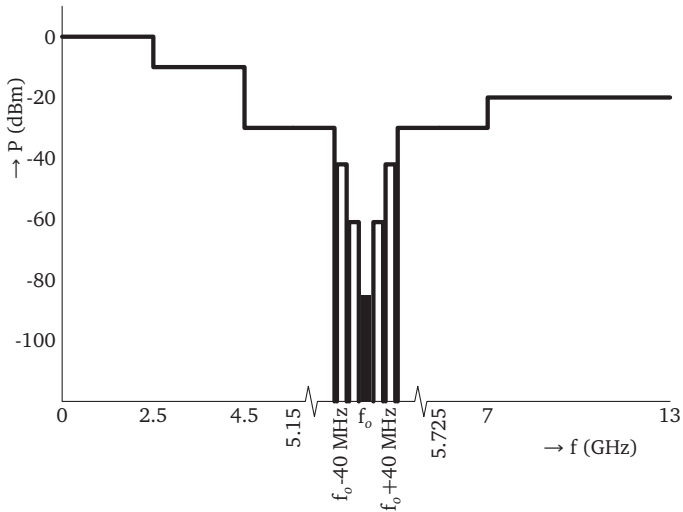
$$\text{FOM}_{\text{Osc-No}} = -96 + 20 \log\left(\frac{1\text{MHz}}{12\text{GHz}}\right) = -176$$

In [27], numbers well under -190 are reported for CMOS integrated oscillators, so the required jitter appears feasible.

It is clear from figure 2.6 that the strongest signal (0 dBm at 2.4 GHz), which was the limiting factor in the first model, does not form a key factor in the second model. This shows that the more realistic model yields far more feasible requirements than the first model (1.3 ps absolute RMS jitter instead of 11 fs).



(a) Bluetooth [3]



(b) HiPerLAN/2 [4]

Figure 2.5: In-band and out-of-band blocking levels for two standards, together with level of the wanted signal (solid bar at  $f_0$ ) during blocking tests. Note: frequency axes are not to scale.

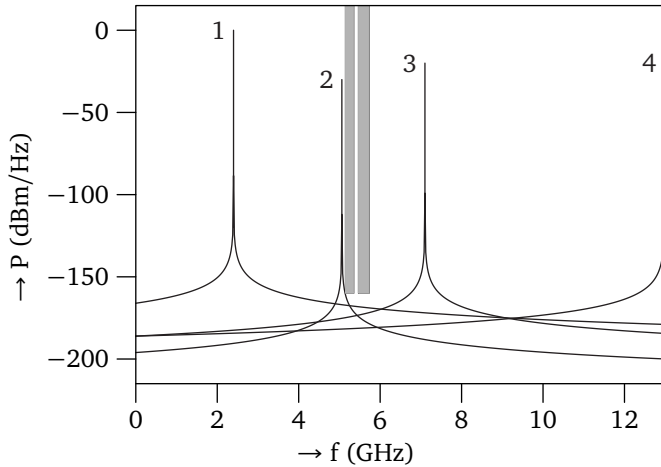


Figure 2.6: ADC output signal (input referred) for four demanding interfering input signals (0 dBm @2.5 GHz, -30 dBm @5.15 GHz, -20 dBm @7 GHz and -20 dBm @13 GHz). Acceptable reception requires the in-band jitter-induced output noise to be below the shaded area. This area is delimited by the band limits (5150-5350 and 5470-5725 MHz) and the maximum acceptable in-band noise level for both Hiperlan/2 bands.

### Maximum jitter spectrum

Until here, the effects of jitter on sampling were analysed, in the context of a software radio receiver. In this section, the converse of this analysis will be discussed. Given the input blockers and a maximum allowed in-band noise level, an upper limit to the jitter spectrum will be derived.

The convolution operation in equation 2.6 shifts the jitter spectrum  $\mathcal{F}(\Delta t(k\tau))$  by the frequency of the interferer. Evaluating the jitter-induced output error at the frequency of the wanted signal  $\omega_w$  therefore yields

$$\mathcal{F}(\Delta s_\tau(k))\Big|_{\omega_w} \approx \omega_i A_i \cdot \mathcal{F}(\Delta t(k\tau))\Big|_{\omega_w - \omega_i} \quad (2.7)$$

Since both an interferer at  $\omega_i - \omega_w$  above the wanted signal and an interferer at  $\omega_w - \omega_i$  below the wanted signal can be shifted to the frequency of the wanted signal, the absolute value of this frequency difference is taken. Also, because  $\Delta t(k\tau)$  is real,  $\mathcal{F}(\Delta t(k\tau))\Big|_{\omega_m}$  is a symmetrical function.

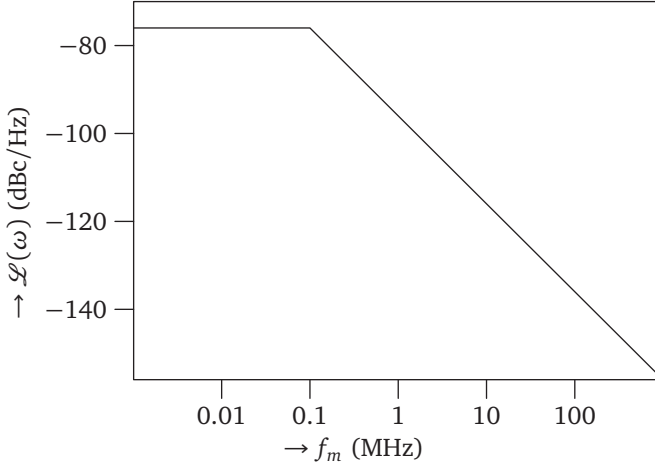


Figure 2.7: Phase noise spectrum of the 12 GHz clock used for the example in figure 2.6.

If we take the PSD  $S_{\Delta s_\tau}(\omega_w)$  of the error signal,

$$\begin{aligned} S_{\Delta s_\tau}(\omega_w) &\approx \omega_i^2 P_i \cdot S_{\Delta t}(|\omega_w - \omega_i|) \\ &= \omega_i^2 P_i \cdot \frac{1}{\omega_{\text{CLK}}^2} S_\Phi(|\omega_w - \omega_i|) \end{aligned} \quad (2.8)$$

Here,  $\omega_{\text{CLK}}$  is the clock frequency,  $S_\Phi$  is the PSD of the phase error and  $S_{\Delta t} = \frac{1}{\omega_{\text{CLK}}^2} S_\Phi$  is the PSD of the time error.

This error signal should be lower than the maximum allowed in-band noise density,  $N_{\text{max}}$ . With  $P_{i,\text{max}}(\omega_i)$  the maximum input power at a certain frequency (the blocker profile),

$$\omega_i^2 P_{i,\text{max}}(\omega_i) \cdot \frac{S_\Phi(|\omega_w - \omega_i|)}{\omega_{\text{CLK}}^2} < N_{\text{max}} \quad (2.9)$$

$$S_\Phi(|\omega_w - \omega_i|) < \frac{N_{\text{max}} \omega_{\text{CLK}}^2}{\omega_i^2 P_{i,\text{max}}(\omega_i)}. \quad (2.10)$$

This can be rewritten as the following set of upper bounds on the phase noise spectrum, with  $\omega_m = |\omega_w - \omega_i|$  the modulation frequency of the

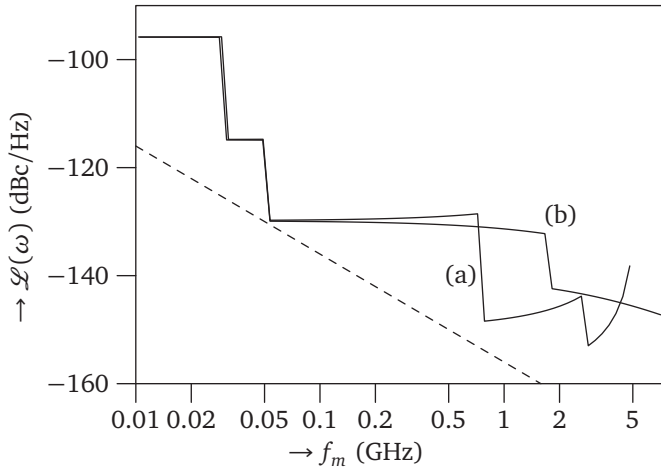


Figure 2.8: An upper bound on the phase noise spectrum for the 12 GHz sampling clock of a software radio receiver capable of receiving Hiperlan/2, calculated using equation 2.11. The labels (a) and (b) correspond to those in equation 2.11. The phase noise spectrum should be below *both* lines. The dashed line represents  $1/f^2$  phase noise of -96 dBc/Hz at 1 MHz offset, just complying with the requirements.

jitter component.

$$\begin{cases} S_{\Phi}(\omega_m) < \frac{N_{\max} \omega_{\text{CLK}}^2}{(\omega_w - \omega_m)^2 P_{i,\max}(\omega_w - \omega_m)} & \text{(a)} \\ S_{\Phi}(\omega_m) < \frac{N_{\max} \omega_{\text{CLK}}^2}{(\omega_w + \omega_m)^2 P_{i,\max}(\omega_w + \omega_m)} & \text{(b)} \end{cases} \quad (2.11)$$

There are two bounds, labeled (a) and (b). These correspond to the two sides of the wanted signal where an interferer might be found.

### Example

This upper bound has been calculated for the same receiver as used in the previous section. The result can be seen in figure 2.8, where  $\mathcal{L}(\omega_m) = \frac{1}{2} S_{\Phi}(\omega_m)$ . Because both the inequalities of equation 2.11 should be satisfied,  $\mathcal{L}(\omega_m)$  should be below both lines.

The steps in the curve correspond to the steps in the blocker profile of figure 2.5. The deviation from horizontal lines is due to the  $(\omega_w \pm \omega_m)$  factor in equation 2.11.

One phase noise spectrum that conforms to these requirements, is a  $1/f^2$  spectrum with  $-96$  dBc/Hz at 1 MHz offset. This is indicated by the dashed line in figure 2.8. One thing to note is the sharp rise towards 5.5 GHz, corresponding to the frequency of the wanted input signal. Jitter at this frequency will convolve with a DC input to the wanted output frequency. As jitter has no effect on sampling a DC signal (its derivative is zero), the jitter spectrum may be arbitrarily high at that frequency.

### Conclusions

When judging the effect of clock jitter on the output of samplers over the entire frequency range, knowledge of only the RMS value of the time jitter, combined with knowledge of amplitude and maximum frequency of the input signal is sufficient. When one is only interested in a narrow portion of the bandwidth of the sampled signal, as in a software radio receiver, the same approach yields overly stringent requirements on the clock jitter.

Combining knowledge of the jitter spectrum with knowledge of the spectrum of the input signal, can lead to more accurate and far more relaxed estimates for clock jitter requirements, in the example shown by more than two orders of magnitude. Actually, jitter requirements for the clock of a sampler-based receiver are quite close to the requirements for the LO in a mixing receiver, as will be clear when they are compared in section 3.4.

#### 2.2.4 SFDR

Due to various non-linear effects, the output of ADCs contain more signals than those present at the input. These are called spurious signals, or spuri.

These spurious signals should not exceed a certain level in order not to interfere with demodulation of the wanted signal. Derivation of this maximum allowed level requires knowledge of the minimum sensitivity level, of the demodulation algorithm and of the requirements on the output data of the demodulator. Unfortunately, for most demodulation algorithms only analyses or simulations with additive white gaussian

sample rate	11.45 GHz
SNR	66 dB (11 bits)
SFDR	91 dB
$f^{-2}$ sampling clock jitter	-96 dBc/Hz @ 1 MHz

Table 2.2: summary of ADC requirements in a combined Bluetooth and Hiperlan/2 software radio receiver

noise are known from the literature. Therefore, ideally some simulations of the demodulation algorithm with harmonic interferers are performed.

The ratio of the single-tone amplitude to the largest spurious signal is called the spurious free dynamic range (SFDR). As with the SNR requirement, the input signal is equal to the maximum blocker signal within the filter bandwidth.

### Example

As with the SNR requirement, the maximum input signal during blocker tests is 0 dBm for Hiperlan/2 en -10 dBm for Bluetooth.

Unfortunately, no data is available for the maximum level of harmonic interfering signals. Therefore, the same levels are taken as for the maximum in-band noise. These are shown in table 2.1 (fourth column) on page 11.

For Bluetooth, during blocker tests the wanted signal is 3 dB over the reference sensitivity level. For Hiperlan/2, the level of the wanted signal during blocking tests is unspecified, so a level of 3 dB over the sensitivity level is assumed as well. Therefore, the combined power of in-band noise+interferer can be 3 dB higher than that of the noise alone, and the power of the in-band interferer can be exactly the same as that of the in-band noise as shown in table 2.1.

As the maximum input signal is 0 dBm and the maximum in-band interferer level is -91.2 dBm, the required SFDR is 91.2 dB.

## 2.3 Feasibility

The requirements, derived in the previous section, are summarised in table 2.2.

To assess the feasibility of a Software Radio, it is not only necessary to know the requirements on the receiver (mainly the ADC), but also the performance that is actually achieved by currently available converters.

Several overviews of available ADCs exist, notably [28], (later summarised in [29]), but they are all more than five years old, and much progress has been made in the mean time. However, it is clear from looking at more recent publications that just the combination of sample rate and SNR is not presently feasible. For instance, a state-of-the-art ADC such as the one presented in [30] achieves an SNR of 48 dB and an SFDR of 65 dB at a sample rate of 1.35 GHz.

Even though such an ADC is not yet feasible, we can estimate the power that such an ADC would consume by extrapolating from existing designs. This is done using a commonly used figure of merit (FoM) for ADCs:

$$\text{FoM} = \frac{P_{diss}}{2^{\text{SNR}_{bits}} f_s} \quad (2.12)$$

Extrapolating from the same state-of-the-art ADC as mentioned above [30], which achieves a FoM of 0.6 pJ/conversion step, the required ADC for our software radio would consume  $\text{FoM} \cdot 2^{\text{SNR}_{bits}} \cdot f_s = 0.6 \cdot 10^{-12} \cdot 2^{11} \cdot 11.45 \cdot 10^9 = 14$  W. Therefore, even if this ADC would exist, its power consumption would be prohibitive for mobile applications.

## 2.4 Summary and Conclusions

The requirements for software radio receiver front-ends were derived, and these requirements were calculated for a receiver for two different standards. From this, it can be concluded that at present, software radio is not feasible.

Software radio however, is only one possible architecture to obtain a flexible radio receiver. The next chapter will discuss a different one.



## Chapter 3

# Software Defined Radio

The previous chapter presented the software radio concept, and concluded that due to constraints in the presently available technology, it is infeasible at the moment.

This was mainly caused by requirements on the ADC. Therefore, a different architecture has to be found. An ADC is still needed because the information has to be available digitally, but the requirements on it can be lowered by filtering. This lowers the dynamic range of the input signal by attenuating strong interferers, and lowers the bandwidth, leading to a lower sample rate requirement. Downconversion is employed as well, because narrowband filters are more easily accomplished at lower frequencies, and because lowpass ADCs avoid many of the problems associated with bandpass ADCs.

Because downconversion and filtering are in part performed by analogue hardware, less functionality will be implemented in software. Despite this, software can still define the functionality of the receiver, hence the name 'software defined radio'.

In this chapter, several downconversion architectures are discussed. First, the requirements for a downconverter without RF pre-filtering are derived. Then, the same requirements are derived for a downconverter with a pre-filter. This is done for filters of different bandwidths and filter orders, to show the trade-off between the requirements of the filter and of the rest of the receiver. After deriving the requirements, some other aspects of downconversion receivers are discussed.

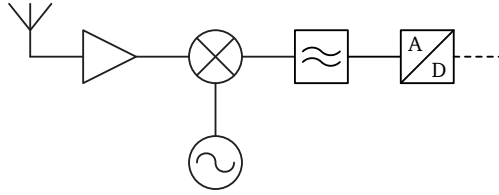


Figure 3.1: A superheterodyne receiver

### 3.1 Downconverter without RF pre-filtering

A more traditional approach to downconversion is the superheterodyne receiver of figure 3.1. Incoming radio signals are first amplified by the low noise amplifier (LNA). The mixer then multiplies them with the LO signal. This corresponds to a convolution with a dirac delta in the frequency domain, i.e. a frequency shift. The downconverted signal is then filtered, and converted to a digital signal. Further filtering and demodulation is performed digitally.

This section presents the requirements for this type of receiver.

#### 3.1.1 Noise requirement

For the software radio receiver in the previous chapter, noise was considered (together with the maximum signal level) as part of the SNR requirement (section 2.2.2). For a more traditional radio receiver however, these parameters are treated separately, and thus noise figure (NF) is a more commonly used parameter.

Noise figure is the difference between the SNR of the output signal and the SNR of the input signal. The input SNR can be calculated from the minimum power level of the input signal, often called the input sensitivity, and the thermal noise within the input bandwidth.

The required output SNR depends on the required bit error rate and on the demodulator. As in section 2.2.2, numbers for this are taken from the literature.

#### Example

Again, the combined Bluetooth / Hiperlan/2 receiver is used as an example. The input SNR (fourth column) is the difference between the minimum input power (second column in table 3.1) and the thermal

standard	required sensitivity (dBm)	in-band th. noise (dBm)	input SNR (dB)	required SNR (dB)	max. NF (dB)
HL/2 (6 Mb/s)	-85	-101.8	16.8	6.2	10.6
HL/2 (9 Mb/s)	-83	-101.8	18.8	7.4	11.4
HL/2 (12 Mb/s)	-81	-101.8	20.8	8.9	11.9
HL/2 (18 Mb/s)	-79	-101.8	22.8	10.9	11.9
HL/2 (27 Mb/s)	-75	-101.8	26.8	14.5	12.3
HL/2 (36 Mb/s)	-73	-101.8	28.8	17.0	11.8
HL/2 (54 Mb/s)	-68	-101.8	33.8	22.3	11.7
Bluetooth	-70	-113.9	43.9	18.0	25.9

Table 3.1: Calculation of the maximum noise figure of the combined receiver, for all (sub-)standards. (Hiperlan/2 is abbreviated here as HL/2)

noise in the input bandwidth (third column). The output SNR required for demodulation (again taken from [2]) is shown in the fifth column. This is subtracted from the input SNR, resulting in the required noise figure (sixth column).

This shows that the required noise figure for the combined Hiperlan/2 and Bluetooth receiver is 10.6 dB.

### 3.1.2 Linearity requirement

When only the wanted signal is present at the input, non-linearity usually has no ill effects. However, when other signals are present as well, through non-linearity these signals can interfere with the wanted signal.

As with the software radio of the previous chapter, the linearity requirements depend on the signal level of the interferers, and on the maximum allowed level of in-band interference. Because numbers for the maximum allowed level of in-band interference are hard to obtain, instead the same levels are taken as for the maximum in-band noise, as was done in section 2.2.4.

This section discusses various non-linear effects and their corresponding requirements.

### Harmonic distortion

Harmonic distortion of an interferer at frequency  $f_i$  leads to products at  $n \cdot f_i, \forall n \geq 2$ . Conversely, a wanted signal at frequency  $f_w$  is susceptible to harmonic distortion products of interferers at frequencies  $\frac{1}{n}f_w$ .

In single-band receivers, interferers at half the frequency of the wanted signal or lower, have been attenuated significantly by the band filter between the antenna and the LNA. Therefore, in these receivers harmonic distortion is usually not a problem.

In wide-band receivers however, it can be a problem. To calculate the maximum harmonic distortion, first the power levels of the interfering signals have to be known. These are taken from the blocker spectra, at the frequencies  $\frac{1}{n}f_w$ .

Also, the maximum level of distortion products that is allowed by the demodulator has to be known. As discussed in section 2.2.4, these are hard to obtain, and again maximum noise levels are used instead.

From these numbers, maximum levels for harmonic distortion could be calculated. This is not done here, because intermodulation already limits the feasibility of the architectures of this section, as will be shown later.

### Second order intermodulation

Intermodulation requires two interfering signals. Taking two harmonic input signals at frequencies  $f_{i,1}$  and  $f_{i,2}$ , then (with  $\phi_{i,1} = 2\pi f_{i,1}t$  and  $\phi_{i,2} = 2\pi f_{i,2}t$ ):

$$\begin{aligned} (\sin \phi_{i,1} + \sin \phi_{i,2})^2 &= \sin^2 \phi_{i,1} + 2 \sin \phi_{i,1} \sin \phi_{i,2} + \sin^2 \phi_{i,2} \\ &= 1 - \frac{1}{2} \cos 2\phi_{i,1} + \mathbf{\cos(\phi_{i,1} - \phi_{i,2})} - \\ &\quad \mathbf{\cos(\phi_{i,1} + \phi_{i,2})} - \frac{1}{2} \cos 2\phi_{i,2} \end{aligned} \quad (3.1)$$

The terms printed in bold are due to intermodulation. Hence, the second order intermodulation products are at frequencies  $f_{i,1} + f_{i,2}$  and  $|f_{i,1} - f_{i,2}|$ .

This means that if one of the interferers is close in frequency to the wanted signal, the other has to be at double that frequency, or at a very low frequency. Therefore, at least one of the two signals is separated far from the wanted signal, In a single-band receiver, that signal has been attenuated by the band filter and second order intermodulation

is usually not a problem. In wide-band receivers however, it can be a problem.

The measure used for second order distortion is the second order input-referred intercept point ( $IIP_2$ ). This is the input power at which (extrapolated) 2<sup>nd</sup> order intermodulation products reach the same output power as does the wanted signal.

To derive the  $IIP_2$  requirement, the level of the interferers has to be known, as well as the maximum in-band interference allowed by the demodulator. The required  $IIP_2$  can be calculated with

$$P_{IIP_2} = \frac{P_i^2}{P_{max,in-band}}$$

Here,  $P_i$  is the input power of the two unwanted signals (usually assumed to be equal) and  $P_{max,in-band}$  is the maximum allowed input-referred in-band interference. As the powers of the two unwanted signals are not necessarily equal, the equivalence formula A.4 derived in appendix A.2 is applied:

$$P_{IIP_2} = \frac{P_{i,1}P_{i,2}}{P_{max,in-band}}$$

There is an infinite number of combinations of  $f_{i,1}$  and  $f_{i,2}$  that leads to second order intermodulation products at  $f_w$ . This is solved by sweeping  $f_{i,1}$  over the whole frequency range over which the blocker profile is defined, and for each  $f_{i,1}$  calculating the two values for  $f_{i,2}$ , that result in an intermodulation product at  $f_w$ .

From all these combinations, the one that leads to the highest  $IIP_2$  requirement is selected.

### Example

Previous linearity requirements were based on only one interferer, but for intermodulation, two are necessary. However, the only requirements from the Bluetooth [3] and Hiperlan/2 [4] standards with out-of-band interferers is the blocker spec, which includes only one interferer. For the design of a single-band receiver this is not of great consequence, since out-of-band interferers are attenuated by the RF filter and this attenuation has a more than linear effect on intermodulation products (quadratic for IM2, cubic for IM3).

In a wideband receiver on the other hand, these interferers are not attenuated. Therefore, assuming two interferers can be present as well, intermodulation distortion can have more effect.

For the combined receiver for Bluetooth and Hiperlan/2, the IIP<sub>2</sub> is calculated from the blocker profile, using the maximisation procedure outlined above.

For Bluetooth, the minimum IIP<sub>2</sub> is determined by two input signals that are both at -10 dBm. Many combinations of frequencies exist that give an intermodulation product in the Bluetooth band, and where the blocker level is -10 dBm. For example, two interferers at 500 and 1950 MHz intermodulate to 2450 MHz. With two interferers at -10 dBm and a maximum in-band noise level of -88 dBm, the minimum IIP<sub>2</sub> is 68 dBm.

For Hiperlan/2, the minimum IIP<sub>2</sub> is determined by two input signals at -10 and 0 dBm. These are for instance present at 750 MHz and 4500 MHz, intermodulating to 5250 MHz. With a maximum in-band noise level of -91 dBm, this gives a minimum IIP<sub>2</sub> of 81 dBm.

Therefore, the combined receiver has an IIP<sub>2</sub> requirement of 81 dBm.

### Third order intermodulation

With two input signals at frequencies  $f_{i,1}$  and  $f_{i,2}$ , third order distortion leads to intermodulation products at  $2f_{i,1} + f_{i,2}$ ,  $f_{i,1} + 2f_{i,2}$ ,  $|2f_{i,1} - f_{i,2}|$  and  $|f_{i,1} - 2f_{i,2}|$ , as can be seen from the following equation.

$$\begin{aligned}
 & (\sin \phi_{i,1} + \sin \phi_{i,2})^3 \\
 = & \sin^3 \phi_{i,1} + 3 \sin^2 \phi_{i,1} \sin \phi_{i,2} + \\
 & 3 \sin \phi_{i,1} \sin^2 \phi_{i,2} + \sin^3 \phi_{i,2} \\
 = & \frac{3}{4} \sin \phi_{i,1} - \frac{1}{4} \sin 3\phi_{i,1} + \\
 & \frac{3}{2} \sin \phi_{i,2} - \frac{3}{4} \sin(\phi_{i,2} - 2\phi_{i,1}) - \frac{3}{4} \sin(\phi_{i,2} + 2\phi_{i,1}) + \\
 & \frac{3}{2} \sin \phi_{i,1} - \frac{3}{4} \sin(\phi_{i,1} - 2\phi_{i,2}) - \frac{3}{4} \sin(\phi_{i,1} + 2\phi_{i,2}) + \\
 & \frac{3}{4} \sin \phi_{i,2} - \frac{1}{4} \sin 3\phi_{i,2} \tag{3.2}
 \end{aligned}$$

The four terms due to intermodulation are printed in bold. By reasoning analogous to that in the previous two sections, the second and last of those four terms are usually irrelevant in single-band receivers. The other two however, are relevant.

In wide-band receivers, all four terms can be a problem. Selecting the combination of two interfering signals that intermodulate to the wanted frequency is done analogous to the procedure in the previous

section, except that now for every  $f_{i,1}$  there are four possible frequencies  $f_{i,2}$  that lead to an intermodulation product at  $f_w$ .

The required IIP<sub>3</sub> can be calculated with:

$$P_{IIP_3} = \frac{P_i \sqrt{P_i}}{\sqrt{P_{max,in-band}}}$$

As before,  $P_i$  is the input power of the unwanted signals and  $P_{max,in-band}$  is the maximum allowed input-referred in-band interference. As the powers of the two unwanted signals are not necessarily equal, the equivalence formula A.5 is applied:

$$P_{IIP_3} = \frac{\sqrt{P_{i,1}^2 P_{i,2}}}{\sqrt{P_{max,in-band}}}$$

or, for intermodulation products at  $|2f_{i,2} - f_{i,1}|$  or  $2f_{i,2} + f_{i,1}$ :

$$P_{IIP_3} = \frac{\sqrt{P_{i,1} P_{i,2}^2}}{\sqrt{P_{max,in-band}}}$$

### Example

The IIP<sub>3</sub> is calculated using the same maximisation procedure as was used for IIP<sub>2</sub>. For Bluetooth this resulted in an IIP<sub>3</sub> requirement of 29 dBm. This number was determined by two -10 dBm signals at for instance 750 and 950 MHz intermodulating to 2450 MHz. For Hiperlan/2, two 0 dBm signals at for instance 1500 and 2250 MHz intermodulate to 5250 MHz. These numbers lead to a minimum IIP<sub>3</sub> of 45.5 dBm. Therefore, the IIP<sub>3</sub> requirement for the combined receiver is 45.5 dbm.

### Compression

Because the output power of an amplifier is limited, gain will drop for increasing input signals. Therefore, in the presence of strong input signals, the wanted signal will also have lower amplification, and sensitivity will drop. This phenomenon is called compression or desensitisation.

In single-band receivers, strong out-of-band interferers are usually attenuated by a filter. However, in a multi-band receiver this is not the case and those interferers will cause desensitisation.

noise figure	10.6 dB
IIP <sub>2</sub>	81 dBm
IIP <sub>3</sub>	45.5 dBm
3-dB compression	0 dBm

Table 3.2: summary of downconverter requirements in a combined Bluetooth and Hiperlan/2 software define radio receiver without an RF pre-filter

#### Example

For Bluetooth, absent a blocker, a wanted signal at the sensitivity level has to be decoded with a certain bit error rate. In the presence of a blocker, a wanted signal at 3 dB over this sensitivity level has to be decoded at the same bit error rate. The maximum blocker level is  $-10$  dBm [3, part A, sec. 4.3].

Therefore, with the blocker present, the wanted signal can be attenuated by 3 dB, so the  $-3$ -dB compression point should be  $-10$  dBm for Bluetooth reception.

For Hiperlan/2, a similar analysis holds, except the maximum blocker level is 0 dBm.

Therefore, the combined reception needs a 3-dB compression point of at level 0 dBm.

#### 3.1.3 Conclusion

This section presented various requirements on a wideband downconverter without RF pre-filtering. Table 3.2 shows a summary of these requirements.

Many downconverters exist with a noise figure of less than 10.6 dB. However, the author is not aware of any published downconverters that come close to meeting the presented linearity requirements. Another problem with this type of receiver would be harmonic mixing: signals around harmonics of the LO will also be downconverted.

Therefore, a different architecture will be discussed in the next section.



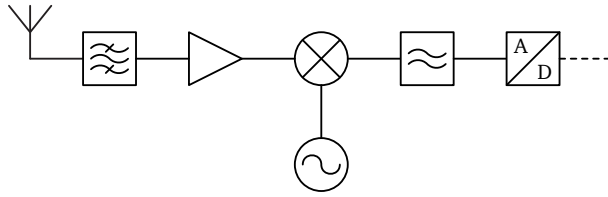


Figure 3.2: A superheterodyne receiver with RF bandpass filter

## 3.2 Downconverter with RF pre-filtering

The major shortcoming of the architecture in the previous section is the high linearity requirement on the receiver. These requirements can be relaxed by adding a bandpass filter in front of the receiver, as shown in figure 3.2.

The blocker spectrum seen in figure 2.5 on page 18 poses an upper limit to the unwanted signals. When pre-filtering is used, the power transfer of the filter can easily be applied to the blocker spectrum to arrive at the equivalent blocker spectrum. This is illustrated in figure 3.3.

As can be seen in this figure, filtering lowers the maximum signal present at the input of the downconverter. A 6<sup>th</sup> order filter with a relatively high bandwidth (1-dB points at the band edges) already attenuates out-of-band blockers to the same level as in-band signals. One could therefore expect this to lower linearity requirements to levels comparable to single-band receivers. To verify this, the rest of this section presents the linearity requirements depending on filter parameters.

### 3.2.1 Linearity requirement

Strong out-of-band interferers imposed very high linearity requirements on the filterless downconverter of the previous section. Using a filter, these out-of-band signals are attenuated, so the linearity requirements on the receiver front-end should be lower.

To see how much lower, the procedure for assessing the various linearity requirements outlined in the previous section can be used again. The difference is that now the filtered blocker spectra are used. This is done for both standards and with several filter orders and filter bandwidths.

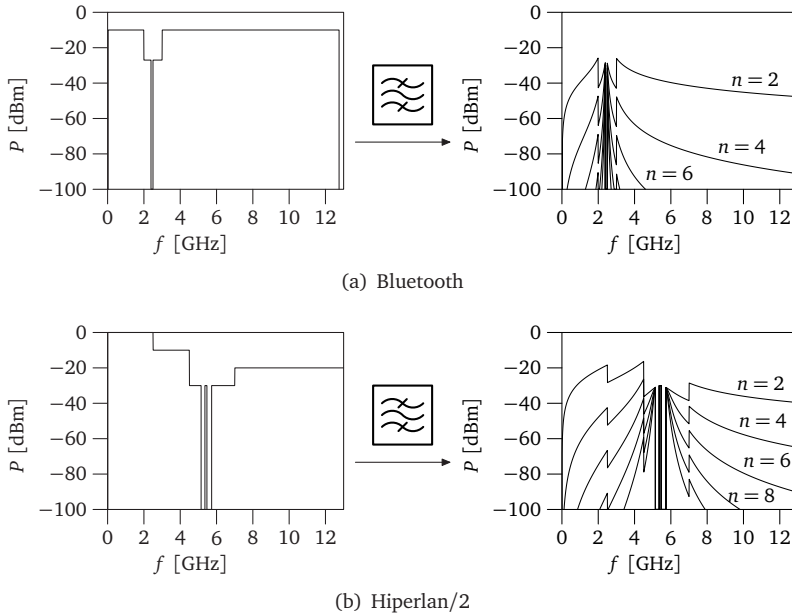


Figure 3.3: Effect of filtering on the blocker spectrum of Bluetooth and Hiperlan/2, using a Butterworth filter of order  $n=2, 4, 6, 8$  or  $10$ , with its  $-1$  dB points at  $2400$  and  $2483.5$  MHz (Bluetooth band edges), and  $5150$  and  $5725$  MHz (Hiperlan/2 band edges).

All calculations are done using a Butterworth filter. Other filter types exist that achieve better steepness, but Butterworth filters are less dependent on component variations and therefore more reproducible. To facilitate a trade-off between filter attenuation and linearity requirements, the attenuation of the filter at the band edges has been varied, as well as the filter order.

In addition to the out-of-band interferers discussed above, the Bluetooth standard also specifies levels of in-band interferers for the purpose of setting odd-order intermodulation distortion requirements [3, section 4.4]. For this test, the wanted signal is at  $-64$  dBm. With a minimum required SNR of  $18$  dB for demodulation (as before), this results in a maximum level for noise+interference of  $-82$  dBm. With noise at  $-88$  dBm, this allows a maximum interference level of  $-83.3$  dBm. The power of the interferers is specified at  $-39$  dBm. These numbers lead to

a minimum  $IIP_3$  of -13.8 dBm. As these interferers are in-band, they are assumed to be independent of the RF pre-filter. Therefore, this level is taken as the minimum  $IIP_3$  for Bluetooth. A comparable specification for Hiperlan/2 does not exist.

The results of these calculation are shown in figures 3.4 ( $IIP_2$ ) and 3.5 ( $IIP_3$ ).

In figure 3.6, the required 3-dB compression point for both Bluetooth and Hiperlan/2 are shown. Both are shown as a function of band-edge attenuation and filter order.

From figures 3.4, 3.5 and 3.6, it can be concluded that indeed the linearity requirements are lowered by filtering. Also, it can be seen that a 4<sup>th</sup> order filter with moderate band-edge attenuation (between 0.5 and 1 dB) yields feasible linearity requirements.

Of interest is the flat requirements for  $IIP_3$  and compression point for Hiperlan/2 reception, as indicated by the horizontal lines in figures 3.5b and 3.6b. These are due to the filter passing both Hiperlan/2 sub-bands, plus the frequency range between them. Therefore, interferers in-between those sub-bands are not attenuated, independent of filter width.

To gain more insight into the blockers that determine the requirements shown in this section, tables 3.3, 3.4 and 3.5 are provided. These tables show the frequency and the level of the ‘worst-case’ blockers, i.e. those that determine the requirements. This is only done for filters with a band edge attenuation of 0.5 dB.

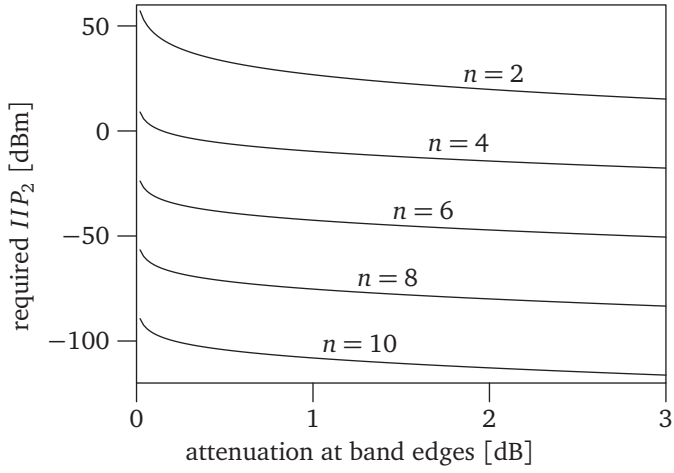
The figures in this section all use the attenuation at the band edges as a parameter. As this attenuation directly adds to the required noise figure, this is an important parameter that can be used when making a trade-off between the required noise figure and linearity.

Another factor to consider is the feasibility of the filter. To assess this, the filter Q is more often used as a parameter. To relate this to the band-edge attenuation, table 3.6 is provided.

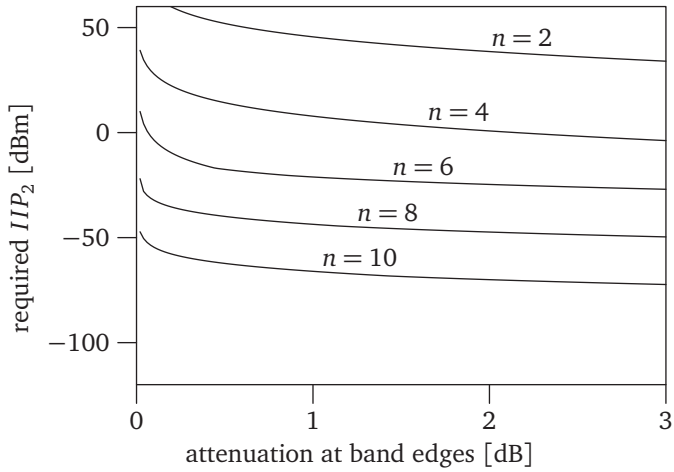
### 3.2.2 Selection of intermediate frequency

Until here, the frequency of the local oscillator and the intermediate frequency (IF) have not been discussed. Several options exist for this.

The main impact this has is on the method to suppress image signals. Image signals have the same frequency difference to the LO signal as the wanted signal, but are ‘on its other side’. They are converted to the same frequency as the wanted signal, and thus lead to interference.

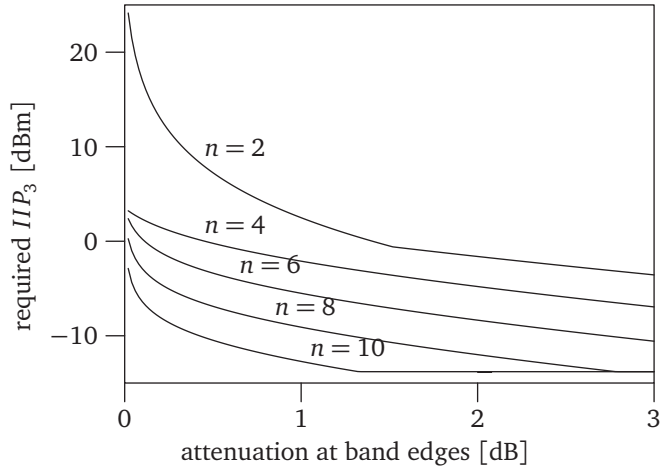


(a) Bluetooth

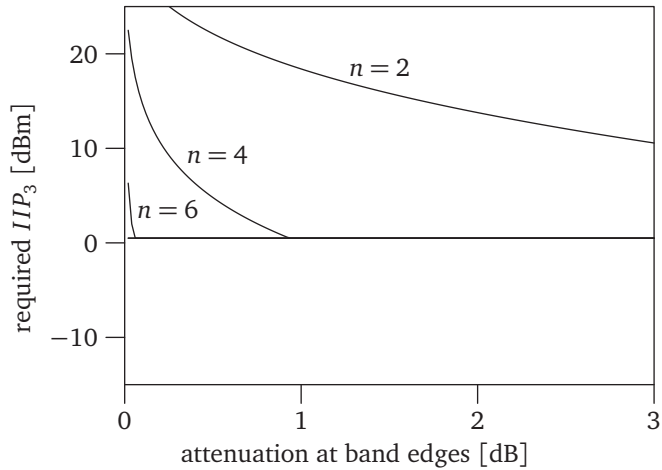


(b) Hiperlan/2

Figure 3.4: required  $IIP_2$  vs. band-edge attenuation for Butterworth filters of order  $n$ .

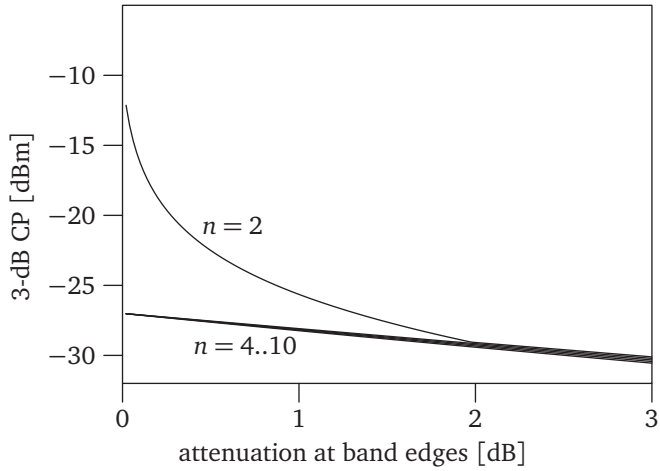


(a) Bluetooth

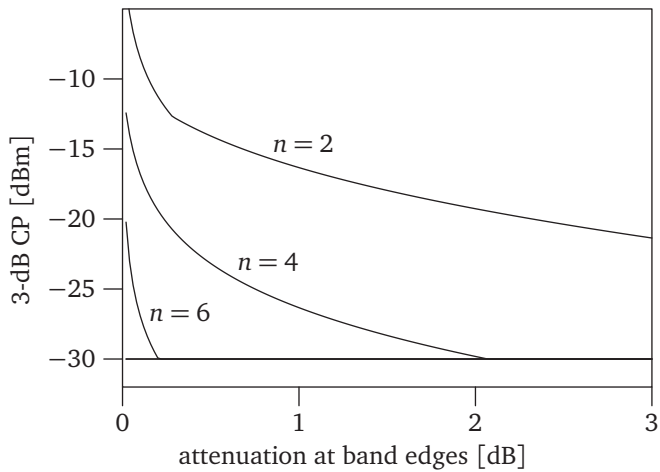


(b) Hiperlan/2

Figure 3.5: required  $IIP_3$  vs. band-edge attenuation for Butterworth filters of order  $n$ .



(a) Bluetooth



(b) Hiperlan/2

Figure 3.6: required 3-dB compression point vs. band-edge attenuation for Butterworth filters of order  $n$ .

n	$f_1$ [GHz]	$P_{in}@f_1$ [dBm]	$A@f_1$ [dB]	$P_{out}@f_1$ [dBm]	$f_2$ [GHz]	$P_{in}@f_2$ [dBm]	$A@f_2$ [dB]	$P_{out}@f_2$ [dBm]	$IIP_2$ [dBm]
0	0.500	-10.0	0.0	-10.0	1.950	-10.0	0.0	-10.0	68.0
2	2.000	-10.0	12.5	-22.5	4.450	-10.0	22.3	-32.3	33.2
4	2.399	-27.0	0.5	-27.5	4.849	-10.0	56.3	-66.3	-5.9
6	2.399	-27.0	0.6	-27.6	4.849	-10.0	89.1	-99.1	-38.7
8	2.399	-27.0	0.6	-27.6	4.849	-10.0	121.8	-131.8	-71.4
10	2.399	-27.0	0.6	-27.6	4.849	-10.0	154.6	-164.6	-104.2

(a) Bluetooth

n	$f_1$ [GHz]	$P_{in}@f_1$ [dBm]	$A@f_1$ [dB]	$P_{out}@f_1$ [dBm]	$f_2$ [GHz]	$P_{in}@f_2$ [dBm]	$A@f_2$ [dB]	$P_{out}@f_2$ [dBm]	$IIP_2$ [dBm]
0	0.750	0.0	0.0	0.0	4.500	-10.0	0.0	-10.0	81.0
2	2.500	0.0	15.2	-15.2	2.750	-10.0	13.9	-23.9	51.9
4	4.500	-10.0	13.2	-23.2	9.750	-20.0	33.6	-53.6	14.2
6	5.150	-30.0	0.5	-30.5	10.400	-20.0	58.0	-78.0	-17.5
8	5.150	-30.0	0.5	-30.5	10.400	-20.0	80.4	-100.4	-39.9
10	5.150	-30.0	0.5	-30.5	10.400	-20.0	102.8	-122.8	-62.3

(b) Hiperlan/2

Table 3.3: An overview of the interferers that determine  $IIP_2$ , using filters with a band edge attenuation of 0.5 dB and varying order. This corresponds to the results shown in figure 3.4 at 0.5 dB.

### High IF

In a conventional superheterodyne receiver, the problem of image signals is solved using an image reject filter. A filter was already needed for relaxing the front-end linearity requirements, but more filtering may be necessary.

One option for lowering the image filter requirement is choosing a very high IF. This is called upconversion and is generally considered a separate architecture. It is discussed in the next section.

Another problem with receivers that reject image signals using a filter, is that this filter is usually placed in front of the LNA. Therefore, noise generated by the LNA at the image frequency is not suppressed, and is subsequently converted to the IF frequency.

### Upconversion

In an upconverting receiver, the IF is placed (much) higher than the input frequencies. This relaxes requirements on the image reject filter, up to the point that even in a wideband receiver, this filter does not

### 3. SOFTWARE DEFINED RADIO

n	$f_1$ [GHz]	$P_{in}@f_1$ [dBm]	$A@f_1$ [dB]	$P_{out}@f_1$ [dBm]	$f_2$ [GHz]	$P_{in}@f_2$ [dBm]	$A@f_2$ [dB]	$P_{out}@f_2$ [dBm]	$IIP_3$ [dBm]
0	0.950	-10.0	0.0	-10.0	0.750	-10.0	0.0	-10.0	29.0
2	3.550	-10.0	17.9	-27.9	3.000	-10.0	12.8	-22.8	7.2
4	2.348	-27.0	6.3	-33.3	2.399	-27.0	0.5	-27.5	-0.2
6	2.348	-27.0	12.6	-39.6	2.399	-27.0	0.6	-27.6	-3.4
8	2.348	-27.0	19.6	-46.6	2.399	-27.0	0.6	-27.6	-6.9
10	2.348	-27.0	26.7	-53.7	2.399	-27.0	0.6	-27.6	-10.5

(a) Bluetooth

n	$f_1$ [GHz]	$P_{in}@f_1$ [dBm]	$A@f_1$ [dB]	$P_{out}@f_1$ [dBm]	$f_2$ [GHz]	$P_{in}@f_2$ [dBm]	$A@f_2$ [dB]	$P_{out}@f_2$ [dBm]	$IIP_3$ [dBm]
0	2.250	0.0	0.0	0.0	1.500	0.0	0.0	0.0	45.5
2	3.750	-10.0	8.6	-18.6	4.500	-10.0	4.1	-14.1	22.1
4	3.750	-10.0	25.1	-35.1	4.500	-10.0	13.2	-23.2	4.8
6	5.470	-30.0	0.0	-30.0	5.360	-30.0	0.0	-30.0	0.5
8	5.470	-30.0	0.0	-30.0	5.360	-30.0	0.0	-30.0	0.5
10	5.470	-30.0	0.0	-30.0	5.360	-30.0	0.0	-30.0	0.5

(b) Hiperlan/2

Table 3.4: An overview of the interferers that determine  $IIP_3$ , using filters with a band edge attenuation of 0.5 dB and varying order. This corresponds to the results shown in figure 3.5 at 0.5 dB.

need to be tunable or switchable. However, some filtering was required anyway to lower linearity requirements.

One of the disadvantages of a much higher LO frequency is the increased jitter, which scales linearly with LO frequency, as discussed further in section 3.4. Another is that the relative bandwidth of the IF filter is much smaller at a higher IF, requiring a higher filter quality factor.

#### Low-IF & Zero-IF

With a low or zero IF, image signals are suppressed by phasing techniques. See figure 3.7 for an example. This obviates the need for an image-reject filter, but instead requires two separate paths, and wide-band 90° phase-shift generation.

The image rejection depends on the matching between the in-phase (I) and quadrature (Q) paths, and on the accuracy of the 90° phase shift. This is more of a problem for low-IF receivers than for zero-IF receivers, because the image signal in a low-IF receiver is a different and possibly stronger channel, while in a low IF receiver the image is the other sideband of the wanted signal. However, in both receiver types known



n	$f_1$ [GHz]	$P_{in}@f_1$ [dBm]	$A@f_1$ [dB]	$P_{out}@f_1$ [dBm]	3-dB CP [dBm]
0	12.750	-10.0	0.0	-10.0	-10.0
2	2.000	-10.0	12.5	-22.5	-22.5
4	2.399	-27.0	0.5	-27.5	-27.5
6	2.399	-27.0	0.6	-27.6	-27.6
8	2.399	-27.0	0.6	-27.6	-27.6
10	2.399	-27.0	0.6	-27.6	-27.6

(a) Bluetooth

n	$f_1$ [GHz]	$P_{in}@f_1$ [dBm]	$A@f_1$ [dB]	$P_{out}@f_1$ [dBm]	3-dB CP [dBm]
0	2.500	0.0	0.0	0.0	0.0
2	4.500	-10.0	4.1	-14.1	-14.1
4	4.500	-10.0	13.2	-23.2	-23.2
6	5.430	-30.0	0.0	-30.0	-30.0
8	5.434	-30.0	0.0	-30.0	-30.0
10	5.439	-30.0	0.0	-30.0	-30.0

(b) Hiperlan/2

Table 3.5: An overview of the interferers that determine the 3-dB compression point, using filters with a band edge attenuation of 0.5 dB and varying order. This corresponds to the results shown in figure 3.6 at 0.5 dB.

errors can be compensated digitally and algorithms exist for estimating these errors for certain classes of input signals.

Another problem is the mixer's imperfect isolation from the LO port to the RF port. Because an LNA has imperfect reverse isolation, the LO signal is radiated by the antenna. Because the LO is (almost) at the same frequency as the incoming signals, they are not attenuated by the bandpass filter. This effect leads to interference for other receivers.

Then there are various effects that lead to a DC offset at the output of the mixer. This is not a problem for low-IF receivers, but it can be for zero-IF receivers.

First, the LO signal can leak to the RF port, where it is partly reflected to the mixer. It then mixes with the original LO signal, leading to a DC offset at the output. The DC level depends on the reflection coefficient of the antenna, which depends on the environment. Assuming a relatively slowly moving environment, this leads to slow variations in this offset.

### 3. SOFTWARE DEFINED RADIO

order	band edge att. (dB)	3-dB BW (MHz)	$f_c/BW$
2	0.5	239	10.2
2	1.0	164	14.9
2	1.5	130	18.8
6	0.5	119	20.6
6	1.0	105	23.3
6	1.5	97	25.2
10	0.5	103	23.7
10	1.0	96	25.5
10	1.5	91	26.8

(a) Bluetooth,  $f_c=2441.75$  MHz,  $x$ -dB BW=83.5 MHz

order	band edge att. (dB)	3-dB BW (MHz)	$f_c/BW$
2	0.5	1646	3.3
2	1.0	1130	4.8
2	1.5	895	6.1
6	0.5	816	6.7
6	1.0	720	7.5
6	1.5	666	8.1
10	0.5	710	7.7
10	1.0	658	8.2
10	1.5	628	8.6

(b) Hiperlan/2,  $f_c=5437.5$  MHz,  $x$ -dB BW=575 MHz

Table 3.6: 3-dB bandwidth and  $Q=f_c/BW$  corresponding to various orders and band-edge attenuations of the Butterworth filter.

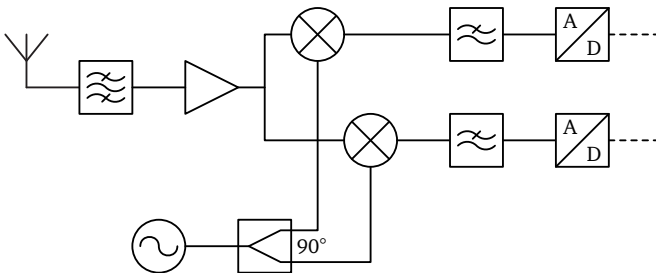


Figure 3.7: A zero-IF receiver

Second, strong signals at the input will, either by 2<sup>nd</sup> order distortion or by the combination of RF-LO leakage, LO port reflection and self-mixing, lead to a DC offset. If these input signals are amplitude-modulated, this results in a quickly varying 'DC' value.

The third effect that leads to DC offset is imperfect balance in the mixer.

A last problem in zero-IF or low-IF receivers can be low-frequency noise caused by mixers. Also, high-frequency operation of mixers generally requires short transistors. Small transistors generate more low-frequency noise than large ones. This will be discussed further in chapter 4.

In summary, low-IF and zero-IF receivers have various imperfections. Some of these are specific to either zero-IF or low-IF. Because their architectures are basically the same, a combination could be made. For signals that have information content close to the carrier frequency, a DC offset cannot be blocked and a low-IF architecture is more attractive. When, on the other hand, neighbouring channels can be strong, a zero-IF approach would be more appropriate.

### **Selection**

All mentioned choices for the intermediate frequency have their advantages and disadvantages. For high-IF receivers, filtering requirements are prohibitively high. This is solved by upconverting receivers, but their jitter demands and lack of integratable IF filters make them unfeasible for full integration. These points also hold for multiple-conversion architectures, where the signals are up- or downconverted more than once.

What rests, are zero-IF and low-IF receivers. Their architectures are almost the same. What separates them, is mainly the IF filter and the low-frequency noise requirements.

For Hiperlan/2, low-frequency noise and DC offset are not a large problem because the central carriers in the OFDM system are not used. A zero-IF receiver is therefore a viable solution.

The above does not hold for Bluetooth.

Therefore, in the spirit of flexibility, a combined zero-IF and low-IF receiver has been chosen. As a zero-IF receiver, it can receive one Hiperlan/2 channel. As a low-IF receiver, in the same bandwidth, it can receive 16 Bluetooth channels. Selection of one of these 16 channels will then be performed digitally.

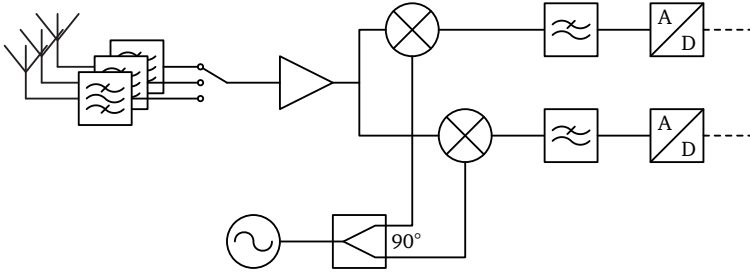


Figure 3.8: A zero-IF or low-IF receiver with multiple bandpass filters

The design of this downconverter will be discussed in the next chapter.

### 3.3 RF Filtering

As concluded above, filters are required to relax linearity requirements on the receiver. In a single-band receiver, this is no problem, as only one fixed-frequency filter is required. For a multi-band receiver however, another solution has to be found.

One would be the use of a tunable filter. Several options exist for this, but most are unpractical. Filters based on electronically variable capacitors probably lack the required linearity. Tunable filters based on yttrium-iron-garnet (YIG) are too bulky and power hungry for use in an integrated receiver. They're also expensive. Filters based on mechanically tunable capacitors or inductors, as seen in older radio equipment are also too bulky and expensive. In the future, RF MEMS filters could be an option [31].

Another option is the use of several fixed filters and to switch between them. See figure 3.8. This will be the option used in the remainder of this thesis.

One downside of this approach is the loss of flexibility: the supported frequency bands have to be chosen before production. Another downside is the need for switches, mainly because of the loss they introduce.

Both of these downsides however, are not as big as they might seem. For efficient transmission multiple antennas are required, so band choices probably have to be made, and switches were required anyway.

Moreover, as filters will often be realised off-chip, the receiver IC itself need not be redesigned or remanufactured. Usually only a new

filter has to be selected to support a new band, often even without the need to redesign the PCB. This does not help with in-field flexibility, but is still an important benefit with regard to non-recurring costs and time to market.

### 3.4 Local Oscillator

Synthesizers used as local oscillators have many requirements, such as tuning range, step size, settling time, spurious signals and phase noise. These requirements all depend on the standard for which a receiver is designed.

In designing an LO for a single-standard receiver, the combination of these requirements determines the design. A multistandard receiver has to fulfil the requirements of all standards. A complicating factor is that the tuning range is much larger than that of any of the individual standards.

Local Oscillator design is outside the scope of this thesis, except for jitter. This is presented here to contrast with section 2.2.3 on jitter requirements for software radio ADC sampling clock jitter.

#### 3.4.1 Jitter

Sampling and mixing are two very similar operations. The first can be modeled by multiplying a signal with a pulse train, while the second can be modeled by multiplying it with a sine (or square) wave.

This could lead to the idea that the effects of clock jitter in a sampler-based receiver and in a mixer-based receiver are equal. To see whether this is true, first the effect of jitter in mixer-based receivers is analysed. Then, these results are compared to those obtained in section 2.2.3 for a sampler-based receiver. Finally, a numerical example is given.

Assume an input signal  $s(t)$ , which is now multiplied with a local oscillator at frequency  $\omega_{LO}$ .

$$\begin{aligned}
 u(t) &= s(t) \cdot \sin(\omega_{LO}(t + \Delta t(t))) \\
 &= s(t) \cdot [\sin(\omega_{LO}t) \cdot \cos(\omega_{LO}\Delta t(t)) \\
 &\quad + \cos(\omega_{LO}t) \cdot \sin(\omega_{LO}\Delta t(t))] \\
 &\approx s(t) \cdot \sin(\omega_{LO}t) + \\
 &\quad s(t) \cdot \cos(\omega_{LO}t) \cdot \omega_{LO}\Delta t(t)
 \end{aligned} \tag{3.3}$$

The first term in this equation is the wanted mixing product, the second term is the unwanted product due to local oscillator phase noise. Taking

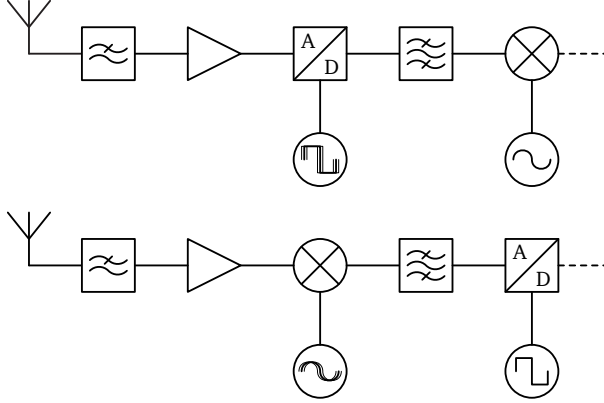


Figure 3.9: Two receiver structures, used for comparing the effects of jitter in mixer-based and sampler-based receivers.

the Fourier transform of the error signal  $\Delta u(t) = u(t) - s(t) \cdot \sin(\omega_{LO}t)$ :

$$\mathcal{F}(\Delta u(t)) = \mathcal{F}(s(t)) * \mathcal{F}(\cos(\omega_{LO}t)) * \omega_{LO} \mathcal{F}(\Delta t(t)) \quad (3.4)$$

With a harmonic input signal  $s(t) = A_{in} \sin(\omega_{in}t)$ , the result is as follows.

$$\mathcal{F}(\Delta u(t)) = \mathcal{F}(\Delta t(t)) * \omega_{LO} A_{in} \cdot \mathcal{F}(\sin(\omega_{in}t)) * \mathcal{F}(\cos(\omega_{LO}t)) \quad (3.5)$$

### 3.4.2 Sampling and mixing compared

A superficial comparison of equations 2.6 (repeated here for convenience):

$$\mathcal{F}(\Delta s_{\tau}(k)) \approx \mathcal{F}(\Delta t(k\tau)) * \omega_i A_i \cdot \mathcal{F}(\cos(\omega_i k\tau)) \quad (2.6)$$

and 3.5 indicates a similarity between the effects of jitter on sampling and mixing. However, as the functional behaviour of samplers and mixers is quite different, a direct comparison is impossible. To compare their jitter effects, we need to have two circuits with identical functionality.

This can be done using the two circuits in figure 3.9. Both of them output a time-discrete signal at a lower frequency than that of the input signal. The difference is that in the upper circuit, the downconversion is performed digitally, while in the lower circuit this is done using an analogue mixer. As discrete-time signal processing is insensitive to jitter, the extra circuit blocks in the upper circuit will have no effect on jitter.

	output noise proportional to	spectral shape
ADC I (sec. 2.2.3)	$A_{in,max}, \omega_{in,max}$	white
ADC II (sec. 2.2.3)	$A_{in}, \omega_{in}$	same as clock, around input freqs
mixer (sec. 3.4.1)	$A_{in}, \omega_{LO}$	same as clock, around input freqs

Table 3.7: Power and spectral shape of jitter induced output noise

Furthermore, because of the lower frequencies involved, we will ignore the sampling jitter effects in the ADC of the lower circuit.

Using derivations analogous to the ones in sections 2.2.3 and 3.4.1, the following expressions for the spectra of the jitter-induced error in the output signals of the two circuits can be derived.

Sampler-based receiver:

$$\mathcal{F}(\Delta s_\tau(k)) \approx \mathcal{F}\left(\left.\frac{\partial}{\partial t}s(t)\right|_{k\tau}\right) * \mathcal{F}(\sin(\omega_{LO}k\tau)) * \mathcal{F}(\Delta t(k\tau)) \quad (3.6)$$

For a harmonic input signal  $A_{in} \sin(\omega_{in})$ , this becomes:

$$\begin{aligned} \mathcal{F}(\Delta s_\tau(k)) &\approx \\ \omega_{in}A_{in} \cdot \mathcal{F}(\cos(\omega_{in}k\tau)) * \mathcal{F}(\sin(\omega_{LO}k\tau)) * \mathcal{F}(\Delta t(k\tau)) \end{aligned} \quad (3.7)$$

Mixer-based receiver:

$$\mathcal{F}(\Delta u_\tau(k)) \approx \mathcal{F}(s(k\tau)) * \mathcal{F}(\cos(\omega_{LO}k\tau)) * \omega_{LO}\mathcal{F}(\Delta t(k\tau)) \quad (3.8)$$

For a harmonic input signal  $A_{in} \sin(\omega_{in})$ , this becomes:

$$\begin{aligned} \mathcal{F}(\Delta u_\tau(k)) &\approx \\ A_{in} \cdot \mathcal{F}(\sin(\omega_{in}k\tau)) * \mathcal{F}(\cos(\omega_{LO}k\tau)) * \omega_{LO}\mathcal{F}(\Delta t(k\tau)) \end{aligned} \quad (3.9)$$

Comparing equations 3.7 and 3.9 shows that although sampling and mixing are very comparable operations, there is one important difference in the jitter-induced output error. For a mixer, this error is not proportional to the frequency of the input signal, but to the frequency of the LO signal.

This comparison is summarized in table 3.7, together with the first ADC model that was discussed. ADC I is that first, commonly used model, as described in section 2.2.3. ADC II is the more realistic model,

as described in section 2.2.3. This shows the difference between the two ADC models, and also shows that the difference between the second ADC model and the mixer is a factor  $\omega_{LO}/\omega_{in}$ .

It is interesting to note that  $\omega_{LO}$  in a zero-IF or low-IF receiver, is very close to  $\omega_{in}$  of the wanted signal, and usually of the same order of magnitude as frequencies of unwanted signals. This leads to very comparable in-band jitter-induced noise levels for both types of receivers.

Note that in this comparison  $\Delta t(t) = \Delta\phi(t)/\omega_{LO}$  is assumed here to be independent of LO frequency (and thus,  $\Delta\phi(t)$  changes with frequency). This is valid with respect to frequency division and multiplication, and typically also holds for oscillators at different frequencies, but with equal tank Q and power consumption.

## 3.5 Channel filtering and analogue-to-digital conversion

The conclusion of the previous chapter was that analogue-to-digital conversion of a wide RF band is currently unfeasible. Filtering and down-converting are considered as a solution and the preceding part of this chapter discussed requirements on this downconverter. This section discusses the remaining question whether this has actually relaxed the requirements on the ADC.

The required sample rate, SFDR and SNR are discussed. These are the exact same requirements as were discussed for the ADC in chapter 2, except for jitter. Jitter is not discussed because the lower signal frequencies after downconversion make it far less of an issue.

These requirements are given separately for both Bluetooth and Hiperlan/2. Because signals for both standards are received using the same baseband filter and ADC, the hardware has to meet both requirements.

The requirements depend on the lowpass filter that precedes the ADC. Comparable to the calculation of the downconverter requirements, the ADC requirements will be calculated for varying order and band edge attenuation of the filter.

The bandwidth of one Hiperlan/2 channel is 16.5625 MHz (53 sub-carriers<sup>1</sup>, with a spacing of 0.3125 MHz). In a zero-IF configuration, a filter passband of  $16.5625/2 \approx 8.3$  MHz is therefore sufficient. Therefore, this will be taken as the band edge at which the attenuation will be varied. Within this passband, 16 Bluetooth channels can be received.

---

<sup>1</sup>A Hiperlan/2 signal consists of 48 data subcarriers, 4 pilot subcarriers, and one unused subcarrier at the RF carrier frequency.



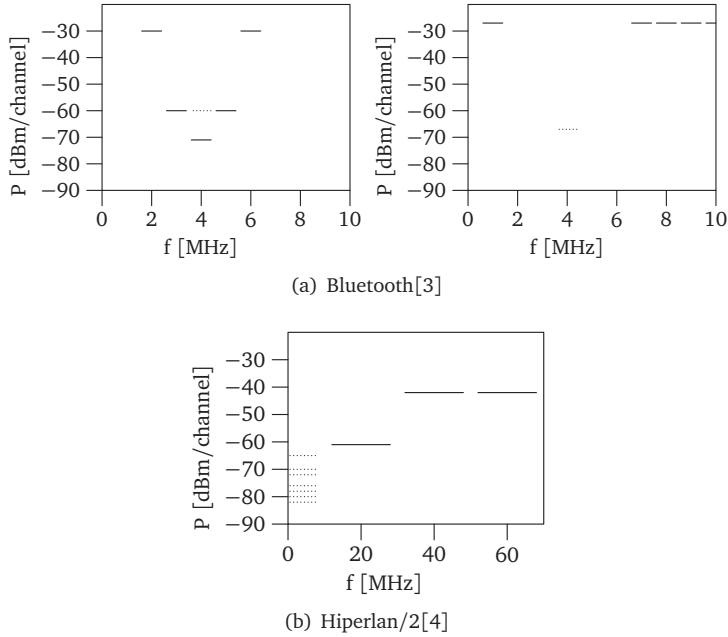


Figure 3.10: In-band blockers, after downconversion. The levels are input-referred. The dotted lines represent the level of the wanted signal during these tests. For Bluetooth two separate graphs are shown, because the level of the wanted signal is not the same for all tests.

Out-of-band blockers will still be taken into account for these calculations, but as they will be strongly attenuated by the filter, the in-band blockers now play a more prominent role. These are shown in figure 3.10. For Bluetooth, two graphs are shown. This is because the level of the wanted signal (indicated by the dotted lines in the graphs) is not the same for all blocking tests. An exception from these blocker levels is made for the image frequency. Here, a blocker only 9 dB stronger than the wanted channel has to be tolerated. A co-channel interferer is also defined, as indicated by the solid line 11 dB below the dotted line of the wanted signal. For Hiperlan/2, seven different levels for the wanted signal are indicated, corresponding to the seven bitrates available. With all tests, only one interfering signal is present at any one time.

For the requirement calculations, the frequency of the wanted signal

was set to 2444 MHz (Bluetooth) or 5250 MHz (Hiperlan/2). The signals are first filtered with a 4<sup>th</sup> order Butterworth RF pre-filter with 0.5 dB attenuation at the band edges (this is one of the filter options used earlier in this chapter to derive front-end requirements, and is the one that will be used for the downconverter in the next chapter). The filtered signals are then downconverted using an LO of 2440 MHz (Bluetooth) or 5250 MHz (Hiperlan/2). The downconverted signal is then lowpass filtered by the filter discussed in this section.

#### 3.5.1 Sample rate

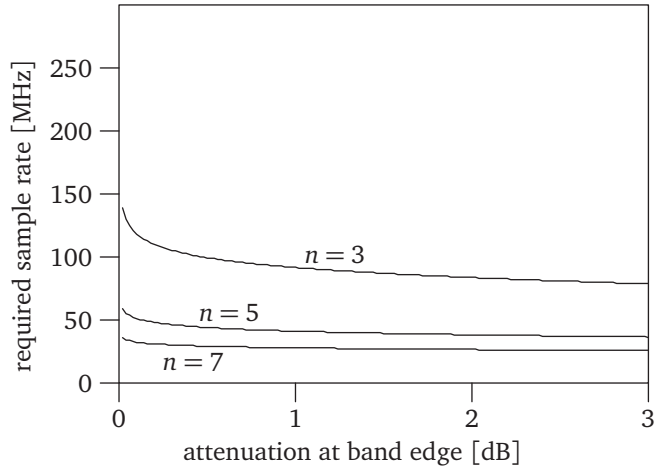
For the software radio in the previous chapter, the required sample rate equalled twice the maximum input frequency, assuming zero power above that frequency. This is a simple way to calculate the sample rate requirement, but is only valid with a brick wall filter at the input.

For the baseband filter discussed in this section, a more realistic filter is used, and a different algorithm is necessary. As with other unwanted effects, the goal is to keep in-band interference below an acceptable level. Therefore, to find a minimum required sample rate, the sample rate is lowered from a sufficiently high value until the aliased signal – integrated over the bandwidth of the wanted signal – is stronger than the maximum allowed in-band interference. The result of this is shown in figure 3.11.

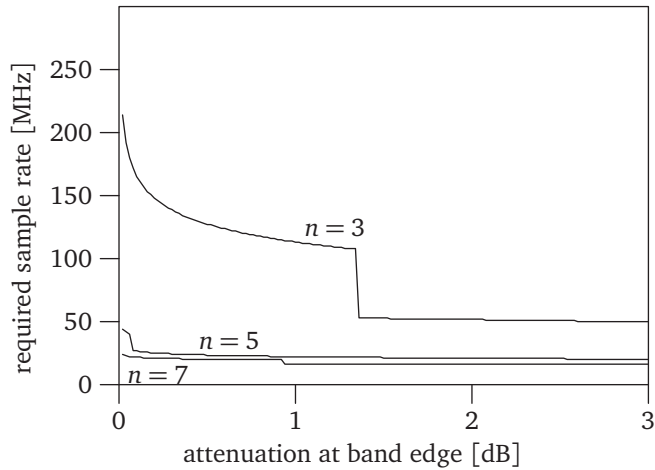
As stated above, for every band edge attenuation the sampling rate is lowered until the aliased interferer exceeds the maximum in-band noise level. For a higher band edge attenuation, the interferers are attenuated more, and the sample rate can be lowered further until the aliased interferer will exceed the maximum in-band noise level. When at a frequency where the level of the interferer changes, the sample rate can suddenly be lowered significantly. This explains the discontinuities seen in figure 3.11b.

#### 3.5.2 SFDR

The required SFDR is the ratio between the maximum signal level at the ADC input and the maximum allowed in-band interference. This can also be calculated for different filters, and the result is shown in figure 3.12. The most stringent requirement is for Bluetooth, since the 10 MHz filter passband includes strong adjacent channels. This is independent of the filter's stopband attenuation and therefore the curve in figure 3.12a is flat. For Hiperlan/2, there is a dependency, but a modest



(a) Bluetooth



(b) Hiperlan/2

Figure 3.11: required sample-rate vs. band-edge attenuation for Butterworth filters of order  $n$ .

amount of filtering already attenuates the blockers enough for them not to dominate the ADC's input signal.

### 3.5.3 SNR

Figure 3.13 shows the required signal-to-noise ratio. Again, the most stringent requirement is for Bluetooth.

The SNR requirement increases with more filtering. This is because with more filtering a lower sample rate can be used (as discussed in section 3.5.1), and with less oversampling more resolution is required. For this calculation, the maximum of the two samplerates in figures 3.11 a and b is used.

### 3.5.4 Resolution

For sinewaves, the formula  $SNR = 6.02 \cdot n + 1.76$  can be used to convert from SNR (in dB) to the resolution  $n$ . For Bluetooth, figure 3.13a shows that with a 5<sup>th</sup> order Butterworth filter with 0.5 dB band-edge attenuation, an SNR of 48 dB is required. This requires a resolution of 8 bits.

For Hiperlan/2, one has to take into account the peak-to-average power ratio (PAPR) of the OFDM signal. The PAPR of an OFDM signal can be calculated as follows [32]:

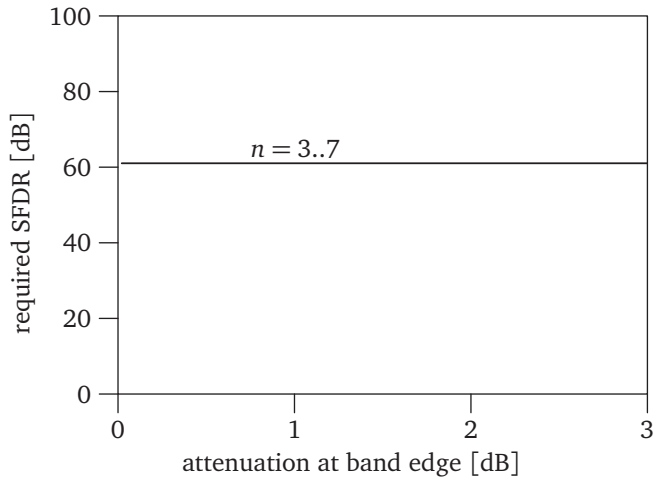
$$PAPR = \frac{N \cdot A_{max}^2}{\sigma_u^2} \quad (3.10)$$

where  $N$  is the number of OFDM subcarriers,  $A_{max}$  the maximum amplitude of one subcarrier, and  $\sigma_u^2$  the average energy per symbol.

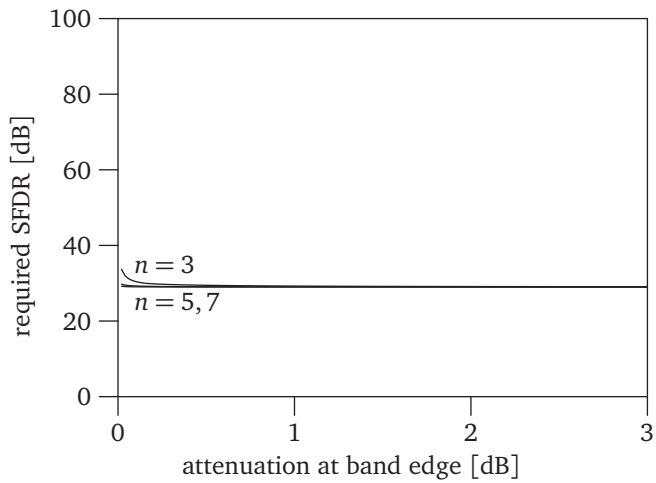
For 64-QAM, which gives the highest  $A_{max}^2/\sigma_u^2$  of the subcarrier modulation types available in Hiperlan/2,  $A_{max} = 7\sqrt{2}$  and  $\sigma_u^2 = 42$  [32]. Hiperlan/2 has 52 subcarriers, so  $N = 52$ . Therefore, the PAPR for Hiperlan/2 is

$$PAPR = \frac{52 \cdot (7\sqrt{2})^2}{42} = 121.3 = 20.8 \text{ dB}$$

However, the largest peaks are very unlikely to occur, and some bit errors due to clipping can be allowed. Therefore, often a value of 12 dB (e.g. [33]), or even lower is used. Here, a PAPR of 12 dB will be assumed.

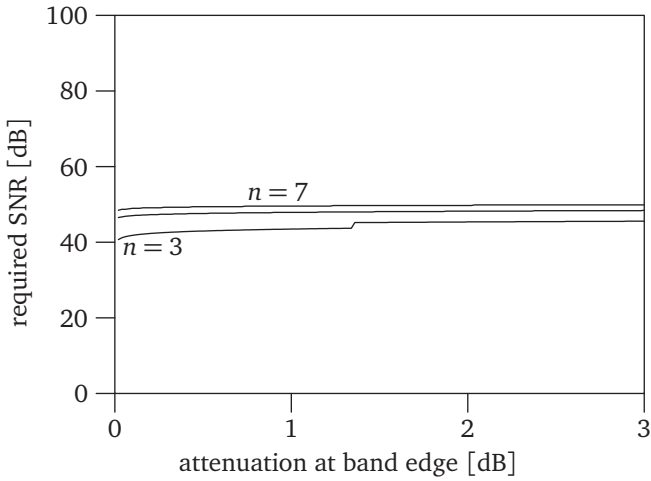


(a) Bluetooth

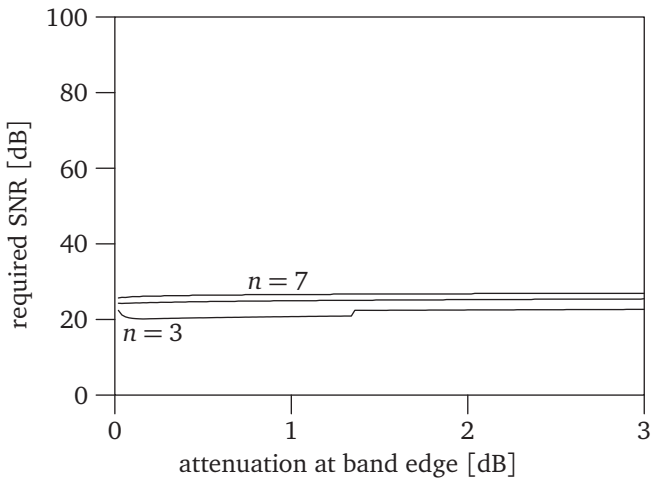


(b) Hiperlan/2

Figure 3.12: required spurious-free dynamic range vs. band-edge attenuation for Butterworth filters of order  $n$ .



(a) Bluetooth



(b) Hiperlan/2

Figure 3.13: required signal-to-noise ratio vs. band-edge attenuation for Butterworth filters of order  $n$ . The sample rate is as shown in figure 3.11, at the same band edge attenuation and filter order. The maximum of the two value for Bluetooth and Hiperlan/2 is used.

Again assuming a 5<sup>th</sup> order Butterworth filter with 0.5 dB band-edge attenuation, for Hiperlan/2 an SNR of 25 dB is required. Taking into account the PAPR, a resolution of 6 bits would therefore be sufficient for Hiperlan/2.

It should be noted that some published receivers for Hiperlan/2 (or for IEEE 802.11a, which has almost the same physical layer) employ ADCs with a resolution of 10 bits, e.g. [34, 35]. No explanation is given for this choice, except that some room is needed to allow for residual DC offset [34]. Presumably, another reason is that part of the variable gain is implemented digitally.

### 3.5.5 Variable gain

The above analysis only pertained to weak signals. For most adjacent and non-adjacent interference tests, the wanted signal is only 3 dB above the minimum sensitivity level. However, according to the standards, the receiver should be able to receive wanted signals up to -20 dBm. This can be solved by increasing the ADC resolution, but a more common solution is varying the gain in front of the ADC.

Another reason for making this gain variable, is that both Bluetooth and Hiperlan/2 have to be supported and their signal levels may not be the same.

Figure 3.14 shows the levels of the signals that are present at the downconverter's input during the relevant tests. The maximum allowed noise levels are integrated over the downconverter's 16.6 MHz pass-band, which explains the low signal-to-noise ratio for Bluetooth, compared to the 18 dB that is required for the demodulator.

Tests (a) through (d) are for Bluetooth. (a) is the sensitivity test, (b) is a test with blockers (corresponding to the right graph in figure 3.10a), (c) another test with blockers (corresponding to the left graph in figure 3.10a) and (d) is the test for the maximum usable signal. Tests (e) through (g) are for Hiperlan/2. The 54 Mb/s substandard was chosen, because this requires the highest SNR. (e) is the sensitivity test, (f) includes blockers (that have been attenuated by filters; otherwise corresponding to figure 3.10b) and (g) is the test for the maximum usable signal.

In this figure, the required ADC SNR of 52 dB is also shown. This is the SNR in a 8.3 MHz bandwidth, which explains why it is higher than the SNR shown in figure 3.13, which shows the SNR over the whole ADC bandwidth. For all of the seven tests, a possible 'scaling' of the ADC has been drawn. For every test except (b), there is some freedom in doing

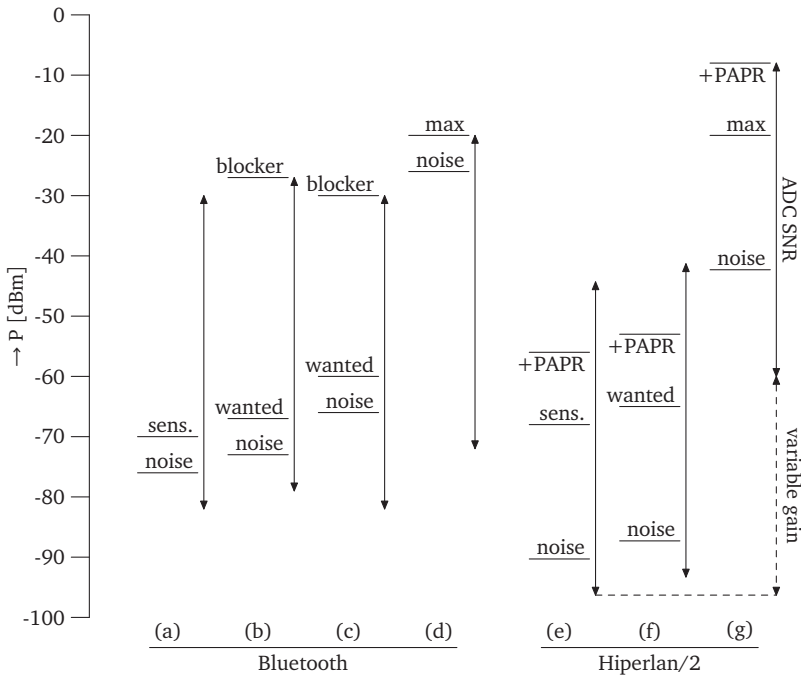


Figure 3.14: The input-referred levels of signals that are present during various tests, used to illustrate the required variable gain. The line labeled 'sens.' represents the level used in the sensitivity test; 'noise' the maximum noise the demodulator will tolerate (integrated over the full 16.6 MHz bandwidth); 'wanted' the level of the wanted signal during blockers tests and 'blocker' the level of the blocker during these tests. The tests (a) through (g) are described in the text.



so. Because the (uncorrelated) noise of two ADC's will be added before demodulation, the required ADC SNR is 3 dB higher than the maximum in-band noise level in figure 3.14 would suggest. Another 3 dB has been added to allow for other noise sources besides the ADC.

On the right, the required variable gain of 36 dB is shown. Of course, this need not be implemented as a variable gain amplifier in front of the ADC, but after a corresponding increase in ADC SNR, can also be implemented digitally.

### 3.5.6 Conclusion

With a 5<sup>th</sup>-order Butterworth lowpass filter with 0.5 dB attenuation at 8.3 MHz, the ADC needs to have a sample rate of 50 MHz (as seen in figure 3.11a), with a resolution of 8 bits (corresponding to the 47.5 dB SNR seen in figure 3.13a). This can be achieved by currently available integrated ADCs. A 5<sup>th</sup>-order Butterworth filter with 0.5 dB attenuation at 8.3 MHz has its -3 dB point at 10.2 MHz.

These requirements are dominated by Bluetooth. This is because 16 Bluetooth channels fit within the Hiperlan/2 passband, and therefore a strong interfering signal might pass the filter unattenuated.

## 3.6 Summary and conclusions

Several architectures for software defined radio were discussed. As summarised in table 3.8, those without an RF pre-filter in general have prohibitively large linearity requirements on the front-end, whereas those with filters appear feasible. Therefore, such a filter is required.

The more conventional architecture of a combined zero-IF/low-IF receiver, combined with switchable filters is the most promising. This architecture and its components will therefore be discussed in the next chapter.

### 3. SOFTWARE DEFINED RADIO

---

order of RF filters	0	2	4	6
noise figure <sup>2</sup>	10.6 dB	10.1 dB	10.1 dB	10.1 dB
IIP <sub>2</sub>	81 dBm	52 dBm	14 dBm	-17 dBm
IIP <sub>3</sub>	45.5 dBm	22 dBm	5 dBm	0 dBm
3-dB CP	0 dBm	-14 dBm	-23 dbm	-28 dBm

Table 3.8: Requirements for the combined Bluetooth / Hiperlan/2 receiver. The RF filter is a Butterworth filter with its 0.5-dB points at the band edges (2400 / 2483.5 MHz for Bluetooth and 5125 / 5750 MHz for Hiperlan/2).

---

<sup>2</sup>Note: noise figure is for whole receiver, including ADCs

## Chapter 4

# A wideband downconverter

As shown in the previous chapters, a wideband downconverter is required as part of a software defined radio receiver. A combined low-IF and zero-IF architecture was presented as the most viable solution. This chapter shows the realisation of such a downconverter.

Some attention has already been given in the literature to the design of wideband, multistandard receiver front-ends, e.g. [36, 37]. These designs are however not in CMOS, which hampers integration with the ever-expanding digital parts of contemporary receivers. At the time of the start of our IC design (in January 2003), we were unaware of any published low-noise wideband front-ends in CMOS. Some wideband LNAs however, have been published, e.g. [38, 39, 40]. Also, several CMOS receivers for satellite reception have been published [41, 42], but as they are to be used in conjunction with outdoor LNCs, their noise figures are rather high (e.g. 16 dB for [42]). This is far too high for wireless applications where the front-end is working directly at RF. The above is summarised in table 4.1.

As said, the design started in January 2003. Since then, a lot of new work has been published. The presented work will be compared to newer designs in section 4.10.

This chapter presents a flexible wideband high-linearity downconverter, implemented in standard 0.18  $\mu\text{m}$  CMOS technology. It has previously been published in [43].

In the next section some design considerations will be discussed. In sections 4.3, 4.4 and 4.5 the circuit is presented. Sections 4.6 and 4.7 contain simulation and measurement results and finally section 4.9 presents some conclusions.

ref.	technology	type	BW [GHz]	NF [dB]
[36]	Si bipolar	downconverter	> 3	3.1
[37]	SiGe BiCMOS	downconverter	0.9-2.4	3.5
[38]	0.25 $\mu$ CMOS	LNA	0.002-1.6	2.4
[39]	0.18 $\mu$ CMOS	LNA	6.8	3.3
[40]	0.18 $\mu$ CMOS	LNA	0.9	4.6
[41]	0.18 $\mu$ CMOS	receiver	0.92-2.15	???
[42]	0.18 $\mu$ CMOS	receiver	0.95-2.15	16

Table 4.1: some previously published designs

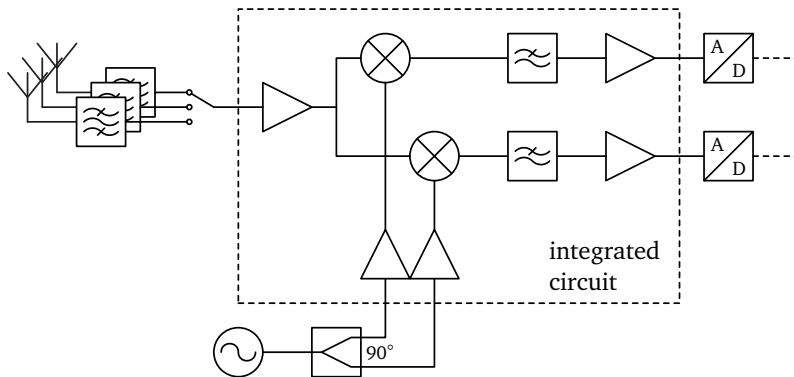


Figure 4.1: system overview

## 4.1 Design considerations

The requirements on the front-end have been derived in the previous chapter. They are summarised in table 3.8 on page 58. From the filter options shown there, the one with fourth-order filters will be the target for this chapter.

Next to these requirements, some others need to be mentioned. First, most commercially available RF-filters and duplexers have  $50\Omega$  input and output impedances. Therefore, in order to maximise chip re-use and flexibility, the input impedance of the downconverter was chosen to be  $50\Omega$ , even though this is non-optimal considering power consumption. This flexibility can be exploited by using several filters and a switch on one PCB, as shown in figure 4.1.

Furthermore, an input that can accept both single-ended and differential-

mode input signals is very much desired to allow for an even wider range of antennas and filters. A fully balanced design also has the advantage of lower even-order distortion, which is important for wideband downconverters. It also reduces other common mode interferences and improves the power supply rejection.

Usually, RF receivers are optimised for low power consumption. In contrast, we have taken the approach to optimise for flexibility. Clearly, this will not lead to minimum power consumption, but for the chosen vehicle (a laptop platform, with more battery capacity than for instance a handheld device), this is less of a concern. Also, note that several hundred milliwatt power consumption may be acceptable, given the fact that high resolution A/D converters and digital signal processing in software will typically consume  $>1$  Watt of power. Furthermore, one piece of equipment, flexible enough to serve many purposes is strongly preferred by end-users, even if this is not power-optimal. For example, many laptop users accept up to 10 W power consumption for playing mp3 music, even though dedicated mp3 players with vastly lower power consumption are available.

## 4.2 Circuit overview

An overview of the downconverter in  $0.18\ \mu\text{m}$  CMOS can be seen in figure 4.2. As discussed in the previous chapter, the downconverter has two mixers for quadrature downconversion. For this, passive mixers have been chosen, mainly because of their relatively good linearity and low  $1/f$  noise, which will be motivated in section 4.4.

Because a passive mixer works by switching currents, the LNA has current mode outputs. These outputs cannot be shared by two mixers, so the LNA has two separate outputs. At the end, the transimpedance amplifiers convert the current into voltage again at intermediate frequency.

The next three sections discuss the constituent parts of the downconverter.

## 4.3 Low-Noise Amplifier

One of the requirements for the LNA is wide-band input matching. In narrow-band LNAs, input-matching is usually achieved with a matching network consisting of inductors and capacitors. However, this only gives matching over a small bandwidth. Another option is resistive input

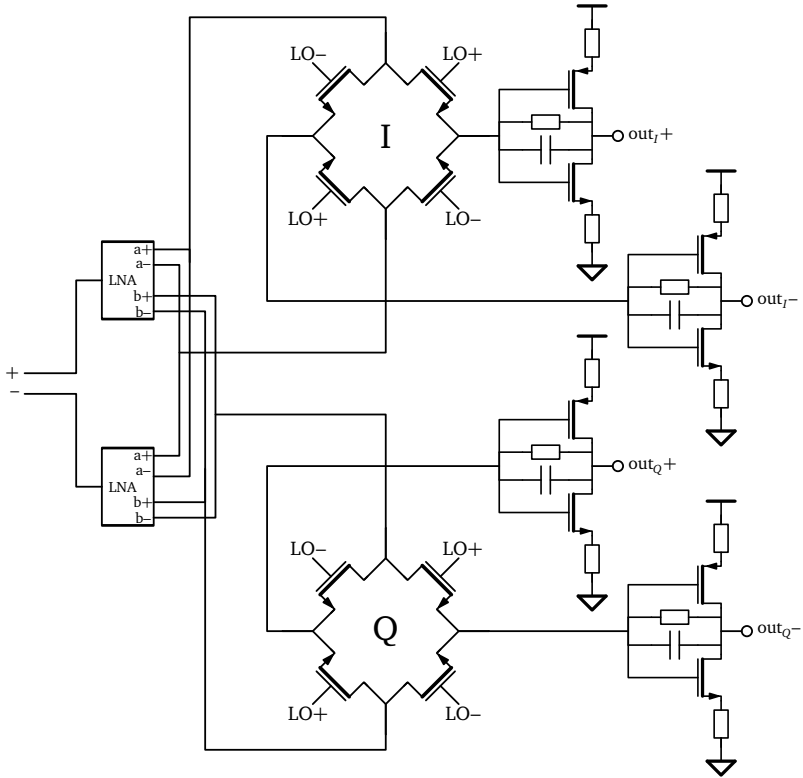


Figure 4.2: Overview of the implemented system, excluding the LO buffers. The schematic of the two LNA blocks is shown in figure 4.5.

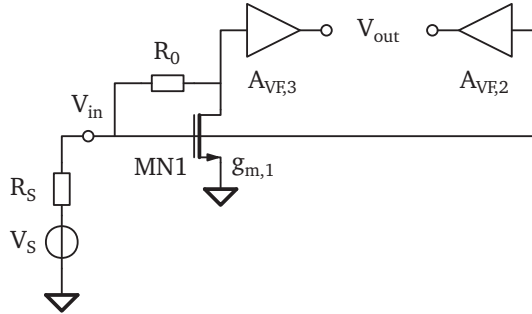


Figure 4.3: The employed noise cancelling topology (biasing not shown).

matching, using either a passive resistor or a transconductor. This is inherently wide-band, but normally suffers from high noise contribution. In this design, noise cancelling [38] is used to decouple input matching and noise figure.

Figure 4.3 shows the noise cancelling topology that will be used. Because this stage is an inverting amplifier, the signal is in anti-phase on the gate and drain of MN1. On the other hand, the noise current in MN1's channel produces in-phase noise voltages at these nodes (through the voltage divider consisting of  $R_0$  and the impedance of the signal source). Both signal and noise on these two nodes are inverted before they reach the two outputs of the LNA. Therefore, the input signal is present on the outputs in anti-phase, while the noise of MN1 is in-phase. This is exploited to add signal contributions while cancelling the noise of MN1. In other words, for a certain gain ratio between the two branches, the channel noise of the input transistor will be cancelled at the differential output.

Several other noise cancelling LNA topologies are known (see [44]), but the one in figure 4.3 has several advantages over the others. First, by employing two identical LNAs and cross coupling their outputs, a balanced LNA will be formed. This has better common mode rejection than using two independent LNAs. Second, as the current source biasing MN1 (which is not shown in figure 4.3) is in parallel to MN1's channel, its noise is cancelled as well.

As discussed previously, the LNA needs current outputs. Therefore, transconductor output stages are added to the LNA, as shown in figure 4.4.

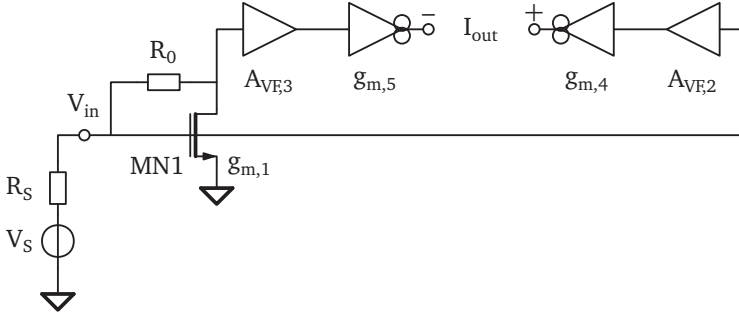


Figure 4.4: overview of one half of the LNA. Compared to figure 4.3, current output stages have been added.

Next, some design equations are presented. First, to achieve input matching, the following is required:

$$\frac{1}{R_S} = g_{m,1} \quad (4.1)$$

To achieve noise cancelling, the output current due to MN1's channel noise current at the two outputs should be equal:

$$\begin{aligned} I_{n,1} \cdot (R_0 + R_S) \cdot A_{VF,3} \cdot g_{m,5} &= I_{n,1} \cdot R_S \cdot A_{VF,2} \cdot g_{m,4} \\ \frac{R_0 + R_S}{R_S} &= \frac{A_{VF,2} g_{m,4}}{A_{VF,3} g_{m,5}} \end{aligned} \quad (4.2)$$

where  $I_{n,1}$  is MN1's channel noise current and the other parameters are as depicted in figure 4.4.

The transconductance from the input to the differential output is given by the following equations.

$$g_m = g_{m,+} - g_{m,-} \quad (4.3)$$

where

$$\begin{aligned} g_{m,+} &= A_{VF,2} g_{m,4} \\ g_{m,-} &= (1 - g_{m,1} R_0) \cdot A_{VF,3} g_{m,5} \end{aligned}$$

The condition for output balancing is

$$g_{m,+} + g_{m,-} = 0 \quad (4.4)$$





The common-source input stage with MN1 is responsible for input matching. A resistive load ( $R_1$ ) instead of a current source improves the bandwidth, at the cost of lowering the voltage gain of this stage from  $1 - g_{m,1}R_0$  to  $\frac{1 - g_{m,1}R_0}{1 + \frac{R_0}{R_1}}$ . Note that since  $R_1$  is in parallel to MN1's channel, its noise is cancelled as well.

With resistive loading of MN1, part of MN1's noise current flows into  $R_1$ . However, the ratio of the induced noise voltages at the gate and drain of MN1 is unchanged and therefore the noise cancelling condition of equation 4.2 is still valid.

In contrast, the conditions for input matching and output balancing are changed. The new input matching condition is:

$$\frac{1}{R_S} = g_{m,1} + \frac{1}{R_1} \frac{1 - g_{m,1}R_0}{1 + \frac{R_0}{R_1}} \quad (4.5)$$

The new output balancing condition is:

$$g_{m,+} + g_{m,-} = 0 \quad (4.6)$$

where

$$\begin{aligned} g_{m,+} &= A_{VF,2}g_{m,4} \\ g_{m,-} &= \frac{1 - g_{m,1}R_0}{1 + \frac{R_0}{R_1}} A_{VF,3}g_{m,5} \end{aligned}$$

Although the addition of  $R_1$  gives one more degree of freedom, combining equations 4.5, 4.2 and 4.6 shows that the input matching, noise cancelling and output balancing conditions still cannot be met simultaneously.

Another stage worth mentioning is the one marked  $A_{VF,2}$  in figure 4.4. To fulfil the noise cancelling condition of equation 4.2,  $A_{VF,2} \cdot g_{m,4}$  needs to be considerably larger than  $A_{VF,3} \cdot g_{m,5}$ . For this reason, and to lower relative noise contributions of later stages, a high value for  $A_{VF,2}$  is needed. Also, because of noise concerns the transistors in this stage need to have a high  $g_m$ . An inverter has a higher  $g_m/I$  than a resistively loaded amplifier, but its pmos transistor has a high input capacitance and its gain is less well defined. A trade-off between bandwidth and  $g_m/I$  therefore resulted in the hybrid stage shown in the lower left of figure 4.5.

Simulations show that the noise cancelling can bring the noise figure of the LNA below 3 dB. With high LNA gain, a front-end with close to 3 dB noise figure could be designed, but at the cost of linearity and bandwidth. However, noise figure was deemed less important than  $IIP_3$  and high bandwidth. Therefore we choose to accept 6 dB NF for the front-end.

The output stages consist of inverters. To improve linearity, these stages are degenerated. In a normal inverter the gates of both the NMOST and PMOST are at the same voltage. This would lead to a lower gate overdrive voltage  $V_{gt} = V_{gs} - V_t$ , which is bad for bandwidth and linearity. Therefore, gate overdrive voltages have been increased by the coupling capacitors and the voltage divider.

All transistors in the LNA have the same gate overdrive voltage. Based on a DC simulation of one transistor, a gate overdrive voltage was selected that gives low higher-order derivatives of the drain current versus gate-source voltage. This should give good linearity performance and resulted in a relatively high  $V_{gt}$  of 0.35 V.

Notwithstanding the use of differential circuits, extensive on-chip supply decoupling is employed to further enhance the power supply rejection.

An annotated schematic is shown in figure 4.6. Among other things, it shows the current consumption per stage.

## 4.4 Mixer

Fully balanced passive mixers were used to achieve high linearity and low  $1/f$  noise. See figure 4.2.

Both mixers consist of four switch transistors. These switches are driven by CMOS inverters acting as LO buffers. Because of the high output impedance of the LNA (current source), a low on-resistance of the switch transistors and the low input impedance of the following IF amplifier, variations in the channel conductivity of the switch transistors have little impact on the signal. This has two advantages. First, because variations in the conductivity caused by large signals have less impact, linearity is improved. Second, both  $1/f$  and thermal noise in the channel current of the switches have less impact, thus allowing smaller transistors to be used, lowering the load presented to the LO buffers, and thus improving LO bandwidth. Especially the lower  $1/f$  noise here is a big plus for zero-IF reception.

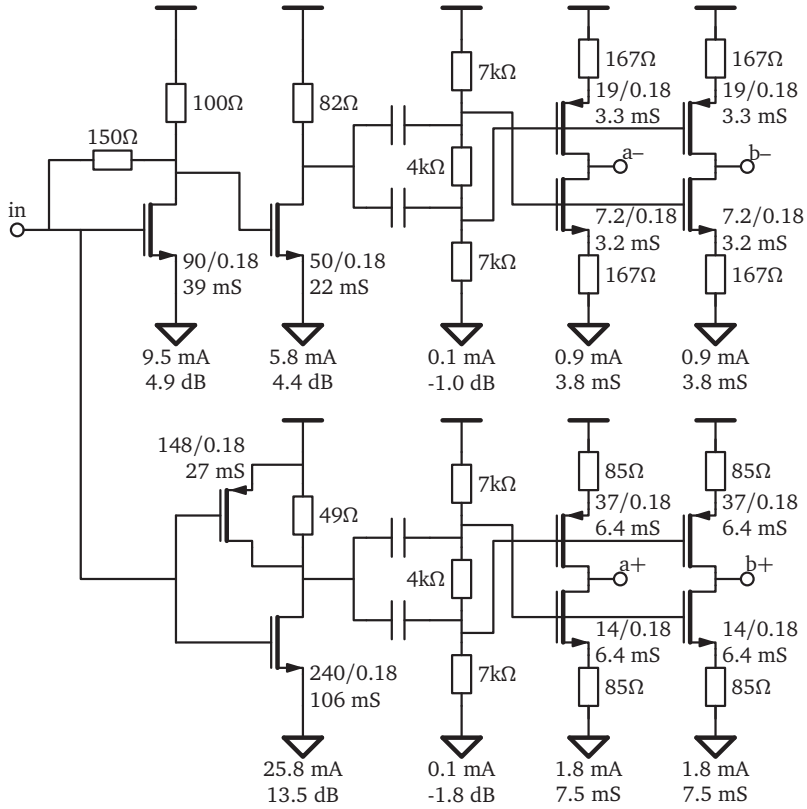


Figure 4.6: annotated schematic of one half of the LNA. For each transistor the  $W/L$  (in  $\mu\text{m}$ ) and transconductance are given and for each stage the current consumption and either the voltage gain (in dB, at 1 GHz) or the transconductance are given. All values have been obtained from simulation of the final design.  $V_{\text{dd}} = 1.8$  V typical.

As mentioned in section 4.2, the LNA has two separate outputs for the two mixers. This is because the current mode output cannot be shared by both. A secondary reason for having separate outputs is to prevent the existence of a low-ohmic current-path between the inputs of two transimpedance amplifiers during the time that both the I- and the Q-path are switched on. This would change the feedback network of the transimpedance amplifiers, thereby amplifying the opamp noise. This effect is comparable to the one described in [45], except that in this case the low-ohmic path would exist continuously instead of just during switch-over.

The LO signals to the mixer are buffered using inverters. At the input of these buffers, the incoming LO signals are terminated into  $50\Omega$  resistors to ground. The outputs of the buffers are AC coupled to the mixers, and the bias voltage at that point can be set externally.

## 4.5 IF filter and amplifier

The IF amplifier and filter is implemented as a transimpedance amplifier with a parallel RC-combination as a feedback network. As shown in section 3.5, a 5<sup>th</sup> order filter with a  $-3$  dB bandwidth of 10.2 MHz is required. However, to enable experiments with this filter, it will be implemented off-chip. In order not to interfere with this off-chip filter, a relatively high bandwidth of 16 MHz has been chosen for the on-chip first-order filter.

To improve the LNA/mixer linearity, the IF amplifier has a low input impedance up to high frequencies. To improve the linearity of the IF amplifier itself, the transistors have been degenerated.

The amplifier was designed for a  $1/f$  noise corner frequency well below 100 kHz. This was done using the brute-force approach of using large transistors. More elaborate techniques exist to lower low-frequency noise, but because the core of this work is in the downconverter, these alternatives have not been investigated.

## 4.6 Simulations

This section shows some results of simulating the complete downconverter, both the circuit as designed, and the netlist extracted from the layout. All simulations were performed using MOS model 9[46] with Cadence's Spectre simulator.

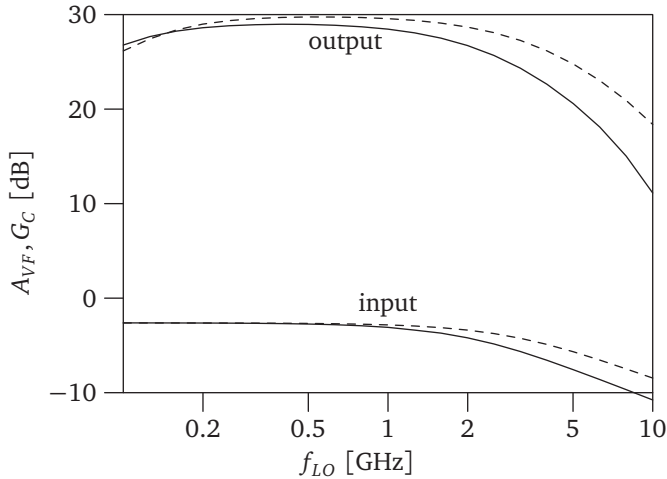


Figure 4.7: simulated voltage gain from source to downconverter input and voltage conversion gain from source to output (input frequency:  $f_{LO}+3$  MHz). Dashed lines are for the circuit simulation; solid lines for simulation of the layout extraction.

For the periodic steady state (PSS) analysis, the local oscillator signal is the large signal, and its frequency is swept. A periodic AC (PAC) analysis is used to obtain the conversion gain from source to output, with the input frequency 3 MHz above the LO frequency.

Figure 4.8 shows the simulation setup.  $V_{\text{source}}$  is the loaded source voltage.  $V_{\text{in}}$  is the voltage at one of the two downconverter inputs.  $V_{\text{out}}$  is the differential voltage at one of the downconverter outputs.

The voltage conversion gain (from source to output) is shown in figure 4.7 (upper dashed line). From this, it can be seen that the  $-3$  dB bandwidth is 3.5 GHz. The lower dashed line is the transfer from source to input. This shows that the bandwidth of the input node is 5.0 GHz. As the bandwidth from input node to output is 6.5 GHz, it can be concluded that the input node limits the bandwidth.

Figure 4.7 also shows the frequency transfer for the circuit including bondpads and parasitic capacitances as extracted from the layout (solid lines). This shows a bandwidth of 2.4 GHz, lower than the one from the circuit simulation. Again, the bandwidth appears limited by the input node. The input node has a bandwidth of 3.1 GHz, and the input-to-output bandwidth is 4.5 GHz.

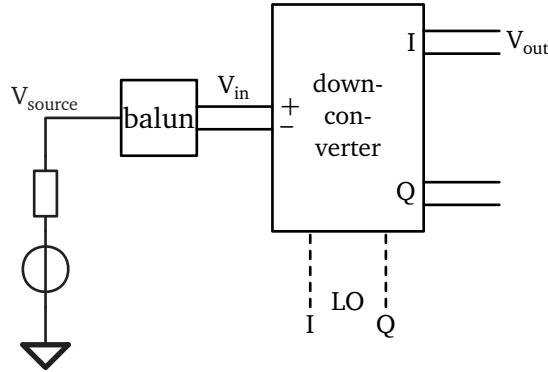


Figure 4.8: Setup including a balun, illustrating the voltages used in figure 4.7.  $V_{\text{source}}$  is the loaded source voltage.  $V_{\text{in}}$  is the voltage at *one* of the two downconverter inputs.  $V_{\text{out}}$  is the differential voltage at one of the downconverter outputs.

It is clear from these numbers that the achieved bandwidth will be insufficient to cover the 5.5 GHz band for Hiperlan/2. This will be discussed further in section 4.10.

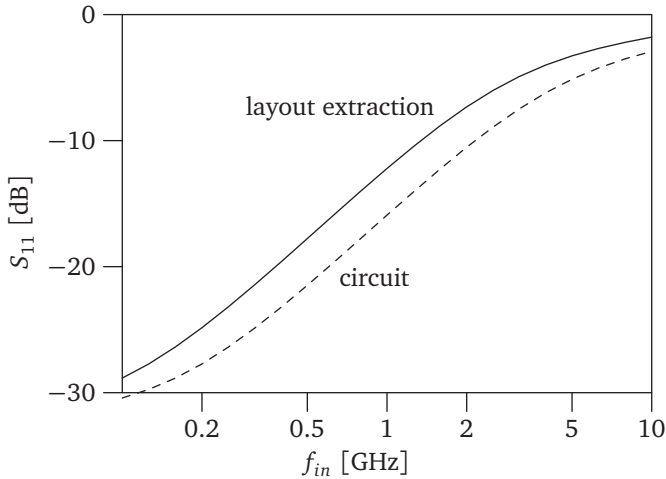
Since the input node already has a low bandwidth, it is interesting to study it in more detail. Figure 4.9 shows the simulated  $S_{11}$  of *one* of the two LNA inputs. This was done to facilitate a direct comparison with measurements, and a single-ended  $S_{11}$  measurement is far less involved than a differential one.

Table 4.2 shows the current consumption of the downconverter's building blocks. The total of 112 mA at a supply voltage of 1.8 V corresponds to a power consumption of 202 mW for the whole downconverter.

## 4.7 Measurements

The front-end was realised in a  $0.18 \mu\text{m}$  standard CMOS process (see figure 4.10 for a photograph). The active chip area is  $800 \times 650 \mu\text{m}$ , most of which is taken by filter capacitors.

Measurements were done on a packaged chip (HVQFN24 package). It was mounted on a PCB made of Rogers RO4003 substrate with a thickness of 0.8 mm. The PCB is shown in figure 4.11. The PCB includes two 33 pF coupling capacitors at the RF input. This gives a -3 dB point at 48

Figure 4.9: Simulated  $S_{11}$  for one LNA input

building block	supply current
LNA	93.2 mA
LO buffers (LO@5.5 GHz)	2.5 mA
IF amplifiers	16.4 mA
total	112.2 mA

Table 4.2: simulated supply current of the downconverter's building blocks

MHz. Four 33 pF coupling capacitors are used at the LO inputs as well. Furthermore, to set the LO inputs' DC value, an 18.9 V voltage source is connected to the four LO inputs via four 1k $\Omega$  resistors. Together with the on-chip 50 $\Omega$  termination resistance, this forms a voltage divider that sets the DC level at 0.9 V. The high resistance value of 1 k $\Omega$  was chosen in order not to load the LO signal. The PCB also includes four amplifiers for the output ports. These are based on THS4271 opamps, and have a voltage gain of 10. A schematic of this amplifier is shown in figure 4.12.

Conversion from single-ended signal sources to the downconverter's differential RF and LO ports was done using hybrids. Up to 3 GHz Tyco H-183-4 hybrids were used, and above that frequency Krytar 4010124. Quadrature LO signals were generated using a 0 $^\circ$  splitter and an adjustable delay line.

For all measurements, the given input levels are corrected for losses



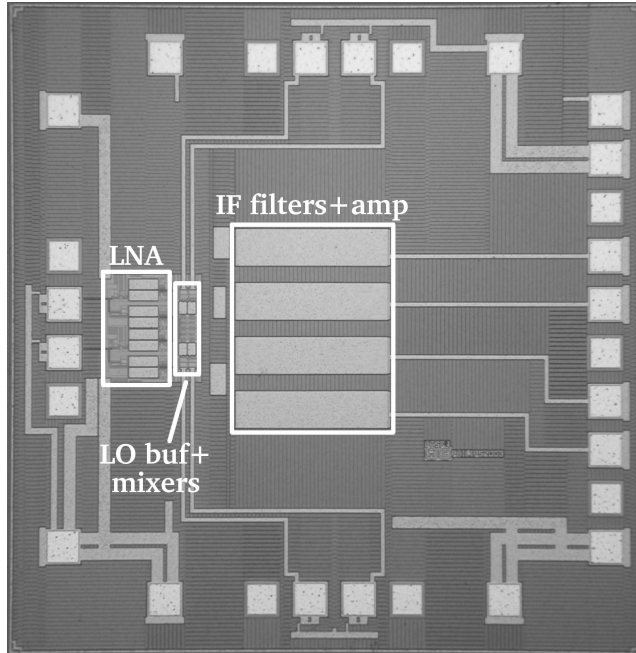


Figure 4.10: Chip micrograph. Total size is  $1.6 \times 1.7$  mm. Active area is  $800 \times 650$   $\mu\text{m}$ .

in cables and in the hybrids. Furthermore, the gain of the opamp amplifiers on the PCB has been subtracted from the total measured conversion gain to arrive at the conversion gain of the downconverter. Corrections have been made for the insertion loss of the hybrids.

Figure 4.13 shows the measured voltage conversion gain as a function of input frequency, showing 200 MHz–2.2 GHz  $-3$  dB bandwidth. The lower cut-off frequency is determined by the coupling capacitors in the LNA. At higher frequencies the conversion gain is still considerable, albeit at increased noise figure.

The same figure also shows the voltage conversion gain resulting from a simulation of the layout extraction. Compared to this, the measured upper  $-3$  dB point is slightly degraded: 2.2 GHz vs. 2.4 GHz. The lower  $-3$  dB point changes considerably: from  $<100$  MHz to 200 MHz. This can partially be explained by the 33 pF on-board coupling capacitors. Furthermore, with 25.6 instead of 28.4 dB, the voltage gain is

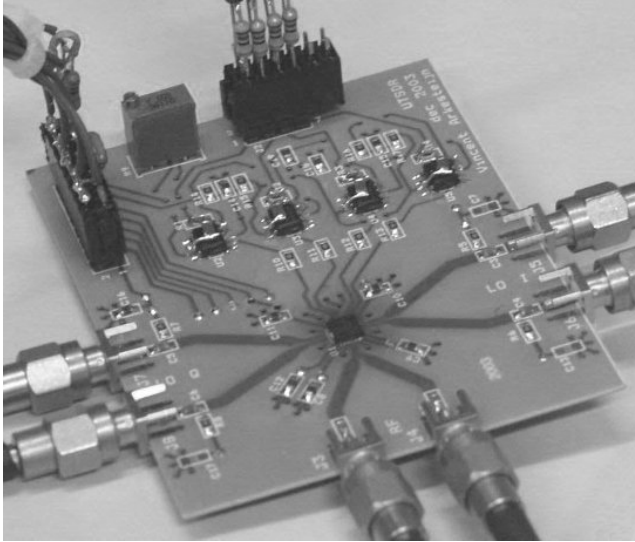


Figure 4.11: the printed circuit board used for measurements

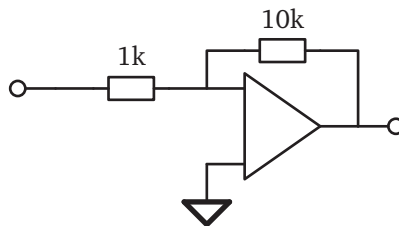


Figure 4.12: the opamp-based amplifier on the PCB

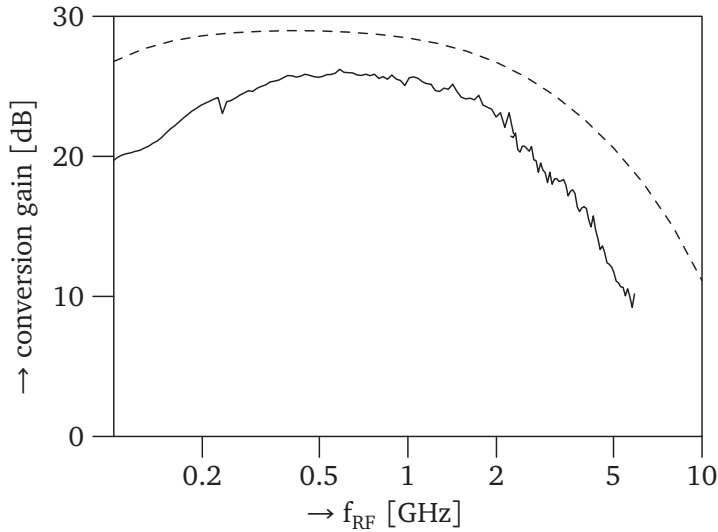


Figure 4.13: voltage conversion gain vs. input frequency (output frequency=5 MHz) ( $V_{dd}=1.8$  V). The dashed line is the result of simulating the layout extraction.

lower than simulated.

Apart from losses in the PCB, one hypothesis is that this is caused by spread of the  $g_m$  of the transistors. The gain is influenced by the  $g_m$  of two cascaded stages, if we restrict ourselves to the path that dominates the gain (the bottom path in the LNA schematic in figure 4.5 on page 65). Then, the gain difference of 2.8 dB corresponds to 1.4 dB per stage. This can be caused by a  $g_m$  that is 15% lower than the nominal value, which is plausible. Resistor spread would be another explanation, as of course, a combination of  $g_m$  and resistance spread would be.

Figure 4.14 shows the measured  $S_{11}$  (solid line). As discussed, this is measured on *one* of the two LNA inputs. This measurement was performed using an RF probe on an unpackaged IC. Calibration of the network analyser was done up to, but excluding, the probe. The dashed line is the result of simulating the layout extraction. For input impedances close to  $50\Omega$ , a small impedance change results in a large change in  $S_{11}$ . Therefore, differences at low values of  $S_{11}$  are to be expected. At higher  $S_{11}$ , a difference of roughly 2 dB is seen. An insertion loss of 1 dB in the RF probe and its SMA connector would already yield a 2 dB lower  $S_{11}$ ,

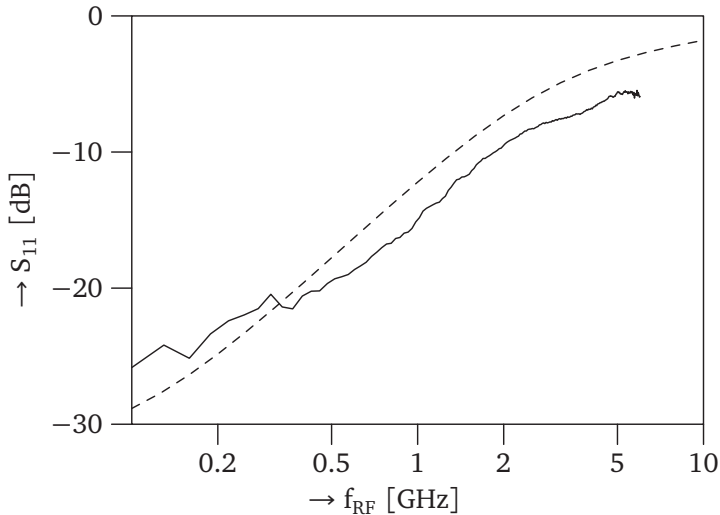


Figure 4.14: Measured  $S_{11}$  vs. input frequency ( $V_{dd} = 1.8$  V). The dashed line is the result of simulating the layout extraction.

which is one possible explanation.

Figure 4.15 shows the DSB noise figure, at two different supply voltages: 1.4V and 1.8V. The supply voltage of the LO buffers has been kept at 1.8 V for both measurements. The noise figure is measured using one output, before the on-board opamps. Assuming most noise is introduced before the signal is split into the I and Q paths, this equals the actual noise figure of the quadrature downconverter.

The dashed line connects the three points of the simulated noise figure (simulated with  $V_{dd} = 1.8$  V). Unfortunately, simulation software problems currently prevent us from performing more simulations. These problems were caused by technology library differences between the version used during the design and the versions available at the time of writing this dissertation.

With lower gain than simulated, a higher noise figure is expected. However, the relatively small difference in noise figure between simulation and measurement (6.5 vs. 6 dB at 1 GHz) indicates that only a small portion of the gain difference is due to stages in the front of the receiver. The larger difference between simulated and measured noise figure at higher frequencies indicates a lower bandwidth. As the simulated noise figure is not from a layout extraction, this can be explained

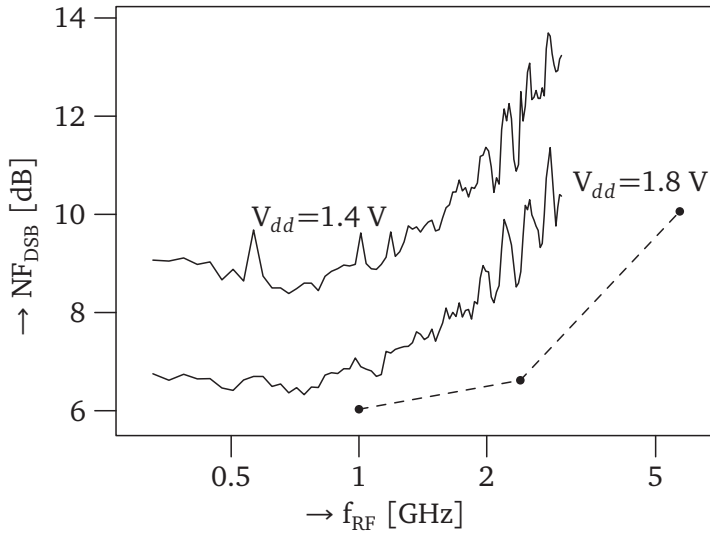


Figure 4.15: DSB noise figure versus input frequency (output frequency=10 MHz). The dashed line represents the result of the circuit simulation (only three points are available).

by the bandwidth difference between the simulations of the circuit and of the layout extraction, as seen in figure 4.7.

Figure 4.16 shows the output noise of the downconverter. This was measured using a differential probe on either the I or the Q output. Note the  $1/f$  noise corner frequency of <50 kHz. The transition around 0.1 MHz is due to the used spectrum analyser, which switches over to a different measurement technique at that frequency. At higher frequencies, the noise drops off due to the on-chip first-order IF filter, with its -3 dB point at 16 MHz.

Figure 4.17 shows an  $IM_3$  plot, measured with an LO frequency of 1 GHz and two input signals at 1005 and 1006 MHz. The  $IIP_3$  is +1 dBm ( $OIP_3$ : 13 dBV), which is considerably better than typically found for narrowband receivers. The -1 dB compression point is -16 dBm and  $IIP_2$  is +35 dBm. This last number varies depending on the baluns used, and could therefore probably be improved further by using baluns with better balancing.

A summary of the measurement results can be found in table 4.3.

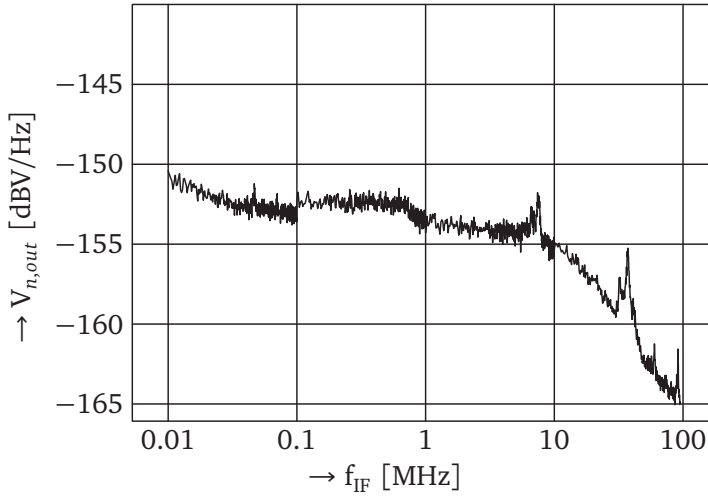


Figure 4.16: output noise vs. frequency. LO=1 GHz,  $V_{dd}=1.8$  V

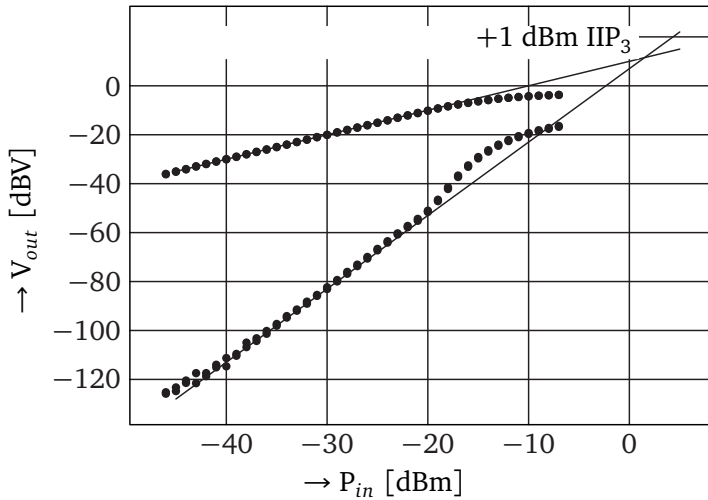


Figure 4.17: Two-tone 3<sup>rd</sup> order intermodulation distortion. input at 1005 and 1006 MHz,  $V_{dd}=1.8$  V.

	Vdd = 1.4 V <sup>1</sup>	Vdd=1.8 V
-3 dB BW	0.2 – 2.2 GHz	0.2 – 2.2 GHz
G <sub>C</sub>	21 dB	25 dB
NF <sub>DSB,min</sub>	8.5 dB	6.5 dB
IIP <sub>3</sub>	+1 dBm	+1 dBm
IIP <sub>2</sub>	+31 dBm	+35 dBm
-1 dB CP	-14.5 dBm	-16 dBm
LO radiation @1 GHz	-47 dBm	-47 dBm
P	130 mW	200 mW

Table 4.3: Key performance measurements of the downconverter

## 4.8 Combination with postprocessing

As mentioned before, the research described in this dissertation was performed in conjunction with that presented in [2]. Within this common project, the goal of the demonstrator platform was to build a software defined radio receiver capable of receiving both Bluetooth and Hiperlan/2.

For this demonstrator, a laptop platform was chosen. One reason was the relatively high available power (compared to other mobile terminals); the other reason was the availability of ample processing power.

This section briefly describes the software (written by Roel Schiphorst) and the digital hardware (designed by Geert Jan Laanstra and Henry Kuipers). It also summarises the results of testing the combination with the downconverter. More detailed information can be found in [2].

### 4.8.1 Software

The receive software implements part of the channel selection, the demodulation, and estimation of parameters such as frequency, phase offset and timing.

For measuring the computational requirements, for both standards a user scenario has been derived. These scenarios model typical transmission parameters such as duty cycle. To test the receiver, transmit functions have been implemented as well.

Table 4.4 shows the computational load of the transmit and receive functions for the two standards. All software is written in C and run on a Pentium 4 at 2.8 GHz.

<sup>1</sup>supply of LO buffers at 1.8 V

function	processor cycles/second
Bluetooth TX	714 MHz
Bluetooth RX	381 MHz
Hiperlan/2 TX	500 MHz
Hiperlan/2 RX	1225 MHz

Table 4.4: Processor cycles used by the various functions

### 4.8.2 Hardware

Figure 4.18 shows the setup that is used to test the combination of the analogue front-end with the digital hardware and software.

The Bluetooth and Hiperlan/2 signals are generated using an Agilent E4438C vector signal generator (VSG). This generator has a memory for a set of I and Q values, which it can generate and upconvert to RF. However, this memory is too small for performing meaningful BER tests. Therefore, a PC with DACs has been connected to the analogue I and Q inputs of the generator. A Krytar 4010124 hybrid was used as a balun that connects the generator to the PCB with the downconverter.

RF pre-filters were not included, because in a measurement setup the effect they have on blocker signals can easily be simulated by changing the output power level of the generator that generates the blocker signal.

The downconverter's LO signals are generated using an HP 8665B signal generator. This generator's signal is split using a  $0^\circ$  splitter. Then, a Radiall R499103000 coaxial phase shifter (a length-adjustable transmission line) is used in one path to generate quadrature signals. As with the RF input signal, Krytar 4010124 hybrids are used to generate balanced signals for the downconverter's LO ports.

As mentioned, the downconverter board includes opamp-based IF amplifiers with a voltage gain of 20 dB. The employed opamps (THS4271) have a unity-gain bandwidth of 1.4 GHz, and in our setup had a tendency to oscillate. Apart from ample supply decoupling, this was remedied by inserting  $47\Omega$  resistors in the output lines. These resistors also provide a characteristic source impedance for the filters that follow, but together with the load they do introduce 6 dB of attenuation.

After the downconverter board, four LC lowpass filters filter the signals. The schematic of one such filter is shown in figure 4.19. In simulations, these filters have a -3 dB point at 10.3 MHz and a -0.5 dB point at 9.2 MHz. According to the calculations in section 3.5, a 5<sup>th</sup> order filter is sufficient. However, the 7<sup>th</sup> order filters had already been selected and built before these calculations were available.



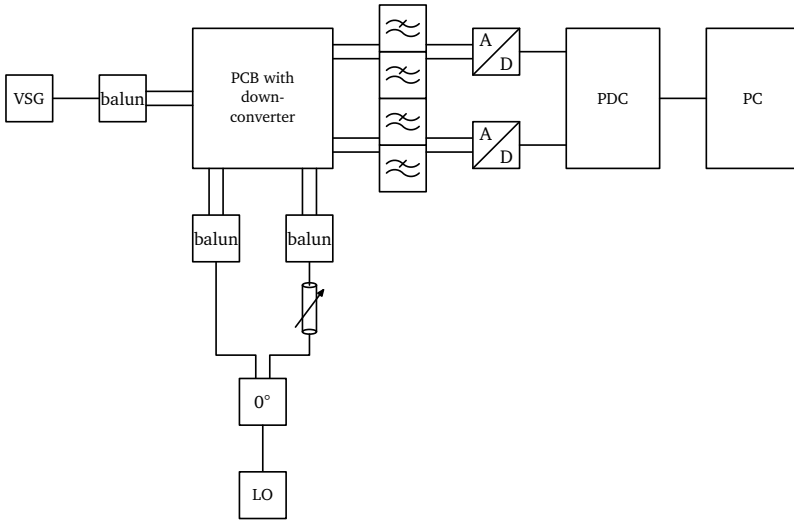


Figure 4.18: the setup used for testing the combination of downconverter with ADCs and post-processing

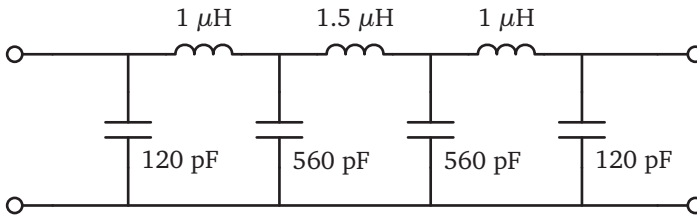


Figure 4.19: schematic of one of the four LC lowpass filter that filters the downconverter's output signals, as shown in figure 4.18

For the two ADCs, evaluation boards for the Analog Devices AD9432 are used. This IC has a nominal resolution of 12 bits, an SNR of 67 dB, and a maximum sample rate of 105 MHz. It is used at a sample rate of 80 MHz. This exceeds the requirements that were derived in section 3.5 (which were a resolution of 8 bits and a sample rate of 50 MHz). Its power consumption is typically 850 mW.

Because the PC cannot handle the full data rate from the ADCs, sample rate conversion is done in hardware. The ADC's output is therefore fed to an Intersil ISL5416 programmable downconverter (PDC). For Hiperlan/2 reception, it performs anti-alias filtering and decimation to a sample rate of 20 MHz. For Bluetooth, it also filters and downmixes one of the 16 channels present in the passband to baseband. Then the output sample rate is 5 MHz, which is more than sufficient for one Bluetooth channel.

The demonstrator is targeted at a laptop platform. For test purposes however, to facilitate experimentation, desktop PCs have been used instead.

The setup as described here lacks variable gain amplifiers. For a fully functional receiver, these would need to be added. To derive the range of input signals that the current receiver can handle, some calculations will be made.

The AD9432 has a full-scale differential input voltage of  $2 V_{pp}$ , or -3 dBV rms. The gain of the front-end is 22 dB for the downconverter (at 2.4 GHz; see figure 4.13), plus 20 dB for the opamps, minus 6 dB for the resistors/filters/load combination as described above. This results in a total front-end voltage gain of 36 dB. Therefore, the maximum input signal to the receiver in order not to overload the ADC is  $-3-36=-39$  dBV or -26 dBm.

The ADC's SNR is 67 dB over the 40 MHz bandwidth, corresponding to a noise floor of -70 dBV. In the 8.3 MHz passband, the noise floor is therefore -77 dBV. For the two ADC's together, this is -74 dBV. Referred to the input, this corresponds to  $-74-36=-110$  dBV or -97 dBm.

However, the downconverter itself adds noise too. At 2.4 GHz the DSB noise figure is 9 dB (see figure 4.15). The receiver input noise is -174 dBm/Hz, or -102 dBm in 16.6 MHz. The input-referred noise added by the downconverter is therefore  $-102+9=-93$  dBm.

The on-board THS4271 opamps in the amplifier of figure 4.12 have an input-referred noise voltage of  $3 \text{ nV}/\sqrt{\text{Hz}}$ . Together with the noise of the resistors, the input-referred noise of one of these amplifiers is  $5.3 \text{ nV}/\sqrt{\text{Hz}}$ . For the differential signal, this becomes  $\sqrt{2} \cdot 5.3 \approx 7.6 \text{ nV}/\sqrt{\text{Hz}}$

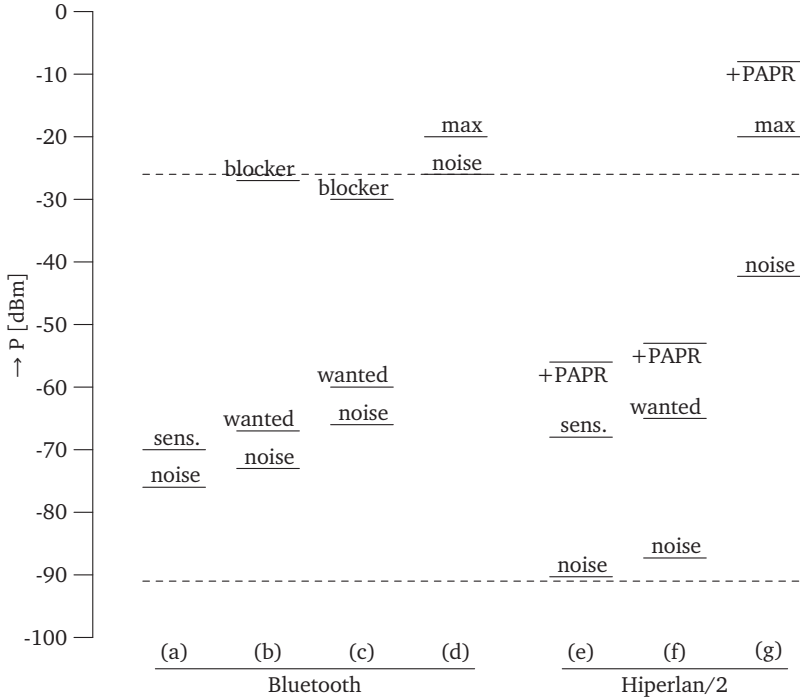


Figure 4.20: The input-referred levels of signals that are present at the ADC input during various tests, as in figure 3.14 on page 56. The two dashed lines represent the maximum input signal and the noise floor of the receiver.

or  $-162 \text{ dBV}/\sqrt{\text{Hz}}$ . As with the ADCs, the opamp's noise from the I and Q paths is added, resulting in another 3 dB for a total of  $-159 \text{ dBV}/\sqrt{\text{Hz}}$ . In the 8.3 MHz bandwidth, this is -90 dBV. Referred to the input of the receiver, this is  $-90-22=-111 \text{ dBV}$  or -99 dBm.

Adding these contributions of -97 , -93 and -99 dBm yields a total input-referred noise of -91 dBm for the whole receiver. This is 2 dB more than the noise of only the downconverter.

The maximum input signal without clipping and the noise floor of the receiver are drawn in figure 4.20, together with the signal levels during various tests. From this figure, we can conclude that all tests except those for the maximum usable signal ((d) and (g)) should pass. This will be tested in the next section.

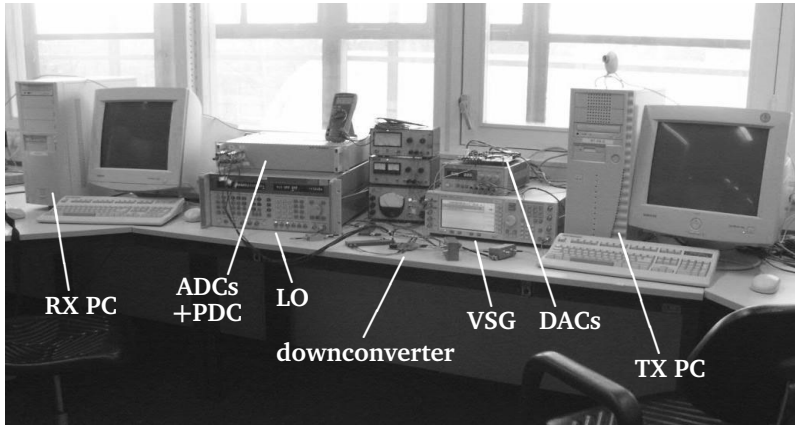


Figure 4.21: photograph of test setup combining the downconverter described in this chapter with digital post-processing described in section 4.8 and [2].

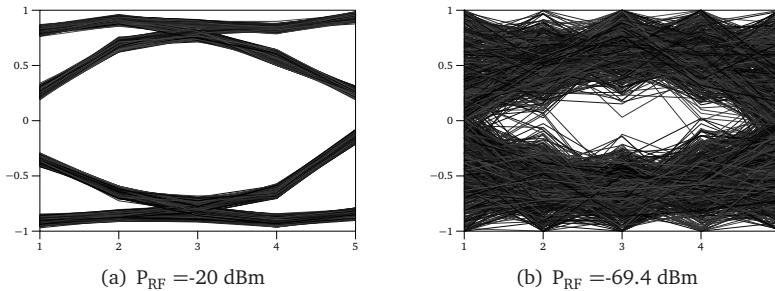


Figure 4.22: eye diagrams of downconverted Bluetooth signals

### 4.8.3 Experimental results

A photograph of the setup described in the previous section is shown in figure 4.21. The PC on the right generates the baseband Bluetooth and Hiperlan/2 signals. These are upconverted using the Agilent E4438C vector signal generator (VSG). After downconversion, the signals are then filtered, converted to digital, filtered, decimated and transferred to the PC on the left.

Using this setup, it is possible to receive both Bluetooth and Hiperlan/2-modulated signals. However, because of the lower gain and higher noise

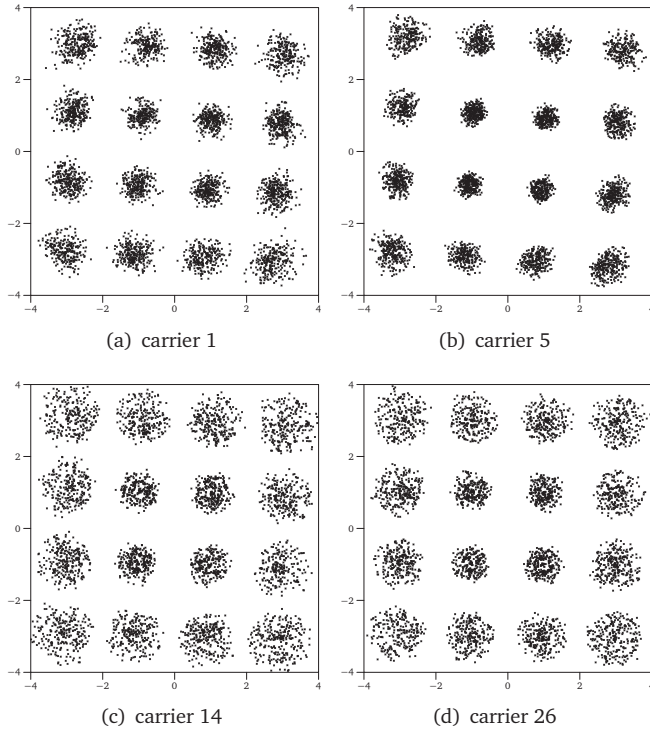


Figure 4.23: four constellation diagrams of different subcarriers of a downconverted Hiperlan/2 signal.  $P_{RF} = -40$  dBm.

figure at 5.5 GHz (as seen in figures 4.13 and 4.15), the Hiperlan/2 experiments have been performed at 2.4 GHz.

Figure 4.22a shows a received Bluetooth eye diagram with an input level of -20 dBm. According to figure 4.20, this maximum usable signal level test (d) should have failed. It did not however, which can be explained because after clipping of a frequency modulated signal, it can still be demodulated. Figure 4.22b shows an eye diagram with an input level of -69.4 dBm. At this level, the received BER is 0.1%, which should have been reached at the reference sensitivity level of -70 dBm.

Figure 4.23 shows four received Hiperlan/2 16-QAM signal constellations. This measurement was done with an RF input power of -40 dBm. It is noteworthy that the noise as seen in these graphs is not the same for all subcarriers, and is stronger for constellation points with a

standard	req. sensitivity	measured sensitivity
Bluetooth	-70 dBm	-69.4 dBm
HL/2 (9 Mb/s)	-83 dBm	-71.0 dBm
HL/2 (18 Mb/s)	-79 dBm	-66.0 dBm
HL/2 (36 Mb/s)	-73 dBm	-53.8 dBm
HL/2 (54 Mb/s)	-68 dBm	n/a

Table 4.5: required and measured sensitivity

higher amplitude.

Sensitivity tests have been performed by lowering the RF signal levels until the error rate drops below a certain threshold (0.1% bit error rate for Bluetooth, 10% packet error rate for Hiperlan/2). Table 4.5 summarises the results of these tests. The required sensitivity has almost been met for Bluetooth, but not for Hiperlan/2.

If this were caused by extra additive white Gaussian noise or unforeseen losses somewhere in the system, the extra noise should be equally large for all Hiperlan/2 subcarriers and for all constellation points, which is not the case. Furthermore, in that case the same difference between required and measured sensitivity would be expected for all Hiperlan/2 substandards, which is also not the case.

It should be noted that additional noise was also seen during baseband tests, where the DAC outputs were connected directly to the ADCs, without the vector signal generator and downconverter inbetween. However, the added noise in these tests was smaller.

Additional research is required to find the cause of the insufficient sensitivity. However, time constraints prevented us from investigating this more, and also from performing blocker tests.

## 4.9 Conclusions

A wideband downconverter front-end has been designed and realised in 0.18  $\mu\text{m}$  CMOS. It achieves 25 dB conversion gain,  $>2$  GHz bandwidth, an  $\text{IIP}_3$  of +1 dBm ( $\text{OIP}_3$ : 13 dBV) and an  $\text{IIP}_2$  of +35 dBm, at 200 mW power consumption. The minimum noise figure is 6.5 dB and the 1/f corner frequency is below 50 kHz.

Table 4.6 shows the requirements that were derived in the previous chapter together with the obtained results for the downconverter. It can be seen that  $\text{IIP}_2$  and compression point are adequate for receivers with

	required with 4 <sup>th</sup> -order filters	required with 6 <sup>th</sup> -order filters	achieved
noise figure	10.1 dB <sup>2</sup>	10.1 dB <sup>2</sup>	9 dB; >10 dB <sup>3</sup>
IIP <sub>2</sub>	14 dBm	-17 dBm	35 dBm
IIP <sub>3</sub>	5 dBm	0 dBm	1 dBm
3-dB CP	-23 dbm	-28 dBm	>-16 dBm

Table 4.6: requirements for receivers with two different orders for the RF pre-filters, together with the achieved performance for the downconverter described in this chapter

either 4<sup>th</sup>-order or 6<sup>th</sup>-order RF pre-filters. The achieved IIP<sub>3</sub> is sufficient for a receiver with 6<sup>th</sup>-order filters.

The noise figure however, is barely sufficient in itself to meet sensitivity requirements. In the presented receiver setup, IF processing degraded the noise figure by another 2 dB, which resulted in a receiver with a total noise figure of 11 dB at 2.4 GHz. Adding RF switches, filters and baluns would add at least another 2 or 3 dB and therefore the required sensitivity can not be met.

Overall, the results indicate that a high-linearity flexible wideband downconverter is feasible in CMOS, but has its price especially in power consumption and higher noise figure.

The downconverter has been combined with demodulation on a standard PC. This combination is able to receive Bluetooth and Hiperlan/2 signals. However, the sensitivity requirements were not met.

## 4.10 Discussion

The receiver presented in this chapter fails to meet noise requirements, in both the 2.4 and the 5.5 GHz band. Furthermore, the power consumption of the downconverter is high compared to narrowband downconverters.

200 mW power consumption for the downconverter is almost negligible compared to the typical power consumed by the Pentium 4 processor used for channel filtering and demodulation. However, it is quite a bit more than that of typical single-standard downconverters. Of course their functionality is different, but it is still interesting to ask where and why this power is spent.

<sup>2</sup>for whole receiver

<sup>3</sup>at 2.4 and 5.5 GHz; for front-end only

#### 4. A WIDEBAND DOWNCONVERTER

ref.	technology	type	BW [GHz]	NF [dB]	IIP <sub>2</sub> [dBm]	IIP <sub>3</sub> [dBm]	P <sub>diss</sub> [mW]
[47]	90nm CMOS	LNA	0–5	2	n/a	-14	42
[48]	0.13 $\mu$ m CMOS	LNA	0.1–6.5	3	n/a	n/a	12
[49]	90nm CMOS	receiver	0.8–6	5	60	-3.5	70
[50]	90nm CMOS	LNA	2.5–4.0	4–5.4	n/a	-8	8
[51]	90nm CMOS	LNA	0–6	2.5	n/a	-17 – -8	9.8
[52]	0.13 $\mu$ m CMOS	LNA	1–7	2.4	n/a	-4.1	25
[53]	90nm CMOS	front-end	2–5.8	3.4	n/a	-21	85
[54]	90nm CMOS	transceiver	0.1–2.5	7	60	-6	48
this	0.18 $\mu$ m CMOS	downconv.	0.2–2.2	6.5	35	+1	200

Table 4.7: A comparison of this work with more recent publications

As shown in table 4.2, most power is spent in the LNA. Figure 4.6 shows the bias current per stage (note that this figure shows only one half of the LNA). More than half of the LNA's power is spent in the stage around MN2 and MP2. As discussed in section 4.3, this stage needs to have a large gain in order to fulfil the noise cancelling condition, and low noise contribution. To meet these requirements a high  $g_m$  is necessary, which leads to high power consumption.

Apart from this, and irrespective of the used topology, improving linearity by using a high  $V_{gt}$  has led to higher power consumption.

Regarding the noise figure, one question then is whether the required noise figure is achievable at 5.5 GHz. Judging from the existence of Hiperlan/2 and IEEE 802.11a transceivers in 0.18 $\mu$ m CMOS (e.g. [55, 56, 57]), it is. However, these receivers use tuned LNAs, which are not suitable for wideband receivers.

As discussed in the beginning of this chapter, at the time that the design of the downconverter started, there were no published low-noise wideband downconverters in CMOS. Since then however, several have been published, in newer technologies. Table 4.7 shows a comparison of the results shown in this chapter with results that were published later. The data suggests that this work has been improved upon.

Restricting the comparison to designs that include downconversion ([49, 53, 54]), it is clear that bandwidth, noise figure and power dissipation have indeed been surpassed, by [49]. The design presented there also uses noise cancelling, but in a different topology. Another approach was described in [54], where five separate LNAs have been used to cover the frequency range. It should also be noted that these parameters, and especially the power dissipation, depend on the implemented function-



ality. For instance, unlike the design discussed in this chapter, those in [49, 53, 54] do include a local oscillator.

The only two publications listed in table 4.7 that also mention  $IIP_2$ , show a higher value. As noted before however, the measured  $IIP_2$  of our design seems to be limited by the wideband baluns that are used. Narrowband baluns probably have better balancing within their passband, which is one option for improving this. Although the requirements derived in chapter 3 and set out in table 4.6 are easily met, the  $IIP_2$  does warrant further investigation. This is because in chapter 3 a non-linear LNA was assumed, followed by a perfect (and perfectly balanced) mixer. However, when the mixer's non-idealities are also taken into account, different mechanisms play a role [58], and requirements may become higher.

Finally, note that this work is still unsurpassed with respect to  $IIP_3$ . This is important for software defined radio receivers, because it allows less RF pre-filtering. Not only can the filters then be wider, but the order can also be lower, which is important for tunable (RF MEMS) filters.



## Chapter 5

# Summary and conclusions

### 5.1 Summary

There is a tendency towards devices that support an increasing number of radio standards and substandards. This calls for a flexible radio, in which little functionality is fixed in hardware. Chapter 1 introduced the concepts of software radio and software defined radio, that both aim at this flexibility. It also described the design vehicle that is used throughout this dissertation: a combined receiver for Bluetooth and Hiperlan/2.

Chapter 2 discussed the software radio, where the entire RF range is digitised and all channel selection and demodulation is performed digitally. Starting from the specifications for the two radio standards, ADC requirements were derived. These requirements appeared to be unrealistic, and therefore a software radio receiver is currently not feasible. Jitter requirements for the ADC's sampling clock were also derived. These turn out to be far less of a problem than would be expected according to a standard textbook jitter analysis.

Chapter 3 discussed the software defined radio, where the RF signals are downconverted before they are digitised. The downconverter's requirements are derived with varying amounts of RF pre-filtering, starting with no filtering at all. Not filtering leads to unrealistic demands on the downconverter's linearity. Increasing the filter's order and lowering its bandwidth gradually relaxes these requirements by attenuating strong out-of-band interferers. After filtering and downconversion, the signals are digitised. The requirements of the ADC needed for this are also derived, depending on the amount of baseband filtering.

Chapter 4 described the design of a wideband downconverter in

0.18 $\mu$ m CMOS technology. The downconverter contains a wideband LNA that employs noise-cancelling to combine wideband input matching with low noise contribution. Passive mixers are used for their good linearity and low  $1/f$  noise. The designed downconverter was combined with ADCs, and with channel filtering and demodulation software running in real time on standards PCs. This resulted in a functional receiver for Bluetooth and Hiperlan/2, although sensitivity requirements were not met.

### 5.2 Conclusions

- A software radio receiver for GHz-range frequencies is currently infeasible, because ADCs with the required specifications are not available.
- The jitter requirements for the sampling clock of this same software radio receiver are feasible, and similar to those for the LO of a mixer-based receiver.
- A software defined radio receiver without RF pre-filtering is currently infeasible, because downconverters with the required linearity are not available. Therefore, RF pre-filtering is required for a software defined radio receiver.
- A wideband receiver front-end for a software defined radio receiver is feasible in CMOS, but noise and power consumption of the presented design need to be improved upon.

### 5.3 Recommendations for further research

The availability of a fast high-resolution ADC, or of a tunable RF filter with sufficient order and Q to lower the downconverter's linearity requirements would be an obvious breakthrough towards a software (defined) radio. Failing that, some other suggestions for future research are as follows.

- The filter that is required to lower the downconverter's linearity requirements can also be designed to act as a resonant uptransformer. This would lower the LNA's power consumption for the same noise figure.
- Investigate other noise cancelling LNA topologies.

# Samenvatting

Radiozenders en -ontvangers komen in veel apparatuur voor, zoals die voor mobiele telefonie of draadloze netwerken. Hierbij is er een trend naar apparatuur die een groter aantal radiostandaarden ondersteunt. Dit pleit voor een flexibele radio, waarbij slechts een klein deel van de functionaliteit vast ligt.

In hoofdstuk 2 wordt de 'software radio' besproken. Hierbij wordt het gehele RF-spectrum gedigitaliseerd, en vindt kanaalselectie en demodulatie digitaal plaats. Uitgaande van de specificaties van de twee radiostandaarden worden de eisen aan de analoog-digitaalomzetter (ADC) afgeleid. Deze eisen blijken onrealistisch, en daarom is een 'software radio' op dit moment niet haalbaar. De eisen die aan de jitter op het klok-sigitaal van de ADC worden gesteld, worden ook afgeleid. Deze blijken veel minder problematisch dan wat men op grond van bestaande theorie zou verwachten.

In hoofdstuk 3 wordt de 'software defined radio' besproken. Hierbij wordt het hoogfrequente ingangssigitaal eerst omlaaggemengd voordat het wordt gedigitaliseerd. De eisen die aan de hiervoor benodigde downconverter worden gesteld, worden afgeleid bij variërende mate van filtering. Zonder filtering blijken de lineariteitseisen onrealistisch hoog. Het verhogen van de filterorde en verlagen van de bandbreedte versoepelt deze eisen doordat sterke stoorsignalen worden verzwakt. Na het filteren en omlaaggemengden worden de signalen gedigitaliseerd. De eisen die aan de hiervoor benodigde ADC worden gesteld, worden afgeleid, afhankelijk van de mate van basisbandfiltering.

In hoofdstuk 4 wordt het ontwerp van een breedbandige downconverter in 0.18  $\mu\text{m}$  CMOS-technologie beschreven. De downconverter bevat een breedbandige voorversterker met ruisonderdrukking. Er worden passieve mixers gebruikt vanwege hun hoge lineariteit en lage laagfrequente ruis. De ontworpen downconverter is gecombineerd met ADC's, en met kanaalselectie- en demodulatiesoftware op normale PC's.

## 5. SUMMARY AND CONCLUSIONS

---

Dit leverde een werkende ontvanger voor Bluetooth en Hiperlan/2 op.

# Dankwoord

Een van de leuke kanten van een promotie-onderzoek is de groep mensen om je heen. Sommigen inspirerend, anderen kritisch, maar allemaal slim en bereid om mee te denken. Enkelen van hen wil ik hier bedanken.

Ten eerste mijn promotor, Bram Nauta. Zonder zijn inhoudelijke kennis, inspiratie en motivatie was dit proefschrift er niet geweest. Ook de inzet mijn dagelijks begeleider en assistent-promotor Eric Klumperink was essentieel. Zijn vakkennis, tomeloze inzet en kritische begeleiding hebben dit proefschrift mede gevormd.

Regelmatig hadden we overleg met de mensen van 'vloer negen'. Roel Schiphorst, Fokke Hoeksema en Kees Slump keken vanuit een ander gezichtspunt naar de materie, en dat leverde vaak interessante inzichten op.

Iets meer op afstand stond de gebruikersraad. Twee maal per jaar kwamen we bijeen. De leden gaven zinvolle feedback, en het was interessant om te zien hoe er vanuit het bedrijfsleven naar het onderwerp werd gekeken. En bijna net zo belangrijk: deze bijeenkomsten leverden altijd weer een deadline op om een stukje onderzoek af te ronden.

Philips Research heeft het ontworpen IC gefabriceerd. Domine Lee-naerts heeft dat proces uitstekend begeleid.

Vele studenten hebben D1-projecten, IOO's en afstudeeronderzoeken uitgevoerd die in meer of mindere mate hebben bijgedragen aan dit proefschrift.

Cor Bakker, Frederik Reenders, Henk de Vries en Gerard Wienk zorgden op uitstekende wijze voor de technische ondersteuning. Erlan H. heeft bijgedragen aan de metingen. Annemiek Janssen, Gerdien Lam-mers en hun talrijke voorgangers stonden altijd klaar voor administratieve hulp.

De aanwezigheid van mijn collega-promovendi was ook belangrijk. Ik noem hier alleen Stephan Blaakmeer, Paul Geraedts, Jan-Rutger Schra-

der, Arnoud van der Wel en Simon Louwsma, maar ook de anderen hebben direct of indirect bijgedragen aan dit proefschrift.

Naast al deze mensen die inhoudelijk bijdroegen, waren er minstens zo veel die voor afleiding zorgden, zowel binnen als buiten de universiteit. Ik ga ze niet opsommen – je weet toch wel dat ik jou bedoel.

Aan al deze mensen: dank jullie wel!



# Appendix A

## Intermodulation distortion

### A.1 Frequencies

This section shows how to calculate the frequencies at which two signals produce intermodulation distortion.

$f_{i,1}$  is the frequency of first interferer. Together with the second interferer at  $f_{i,2}$  it will generate an intermodulation product at a frequency  $f_w$ .

For convenience, the following parameters are defined:

$$\begin{aligned}\phi_{i,1} &= 2\pi f_{i,1}t \\ \phi_{i,2} &= 2\pi f_{i,2}t \\ \phi_w &= 2\pi f_w t\end{aligned}$$

Two sine waves at  $f_{i,1}$  and  $f_{i,2}$  will via some  $n^{\text{th}}$  order non-linearity lead to both harmonic distortion and intermodulation distortion:

$$(A_1 \sin \phi_{i,1} + A_2 \sin \phi_{i,2})^n = \sum_{i=0}^n \binom{n}{i} A_1^{n-i} \sin^{n-i} \phi_{i,1} A_2^i \sin^i \phi_{i,2} \quad (\text{A.1})$$

The terms for  $i = 0$  and  $i = n$  are due to harmonic distortion, while those for  $0 < i < n$  are due to intermodulation distortion.

#### A.1.1 Second order

For  $n = 2$ , A.1 becomes (omitting the terms due to harmonic distortion):

$$(A_1 \sin \phi_{i,1} + A_2 \sin \phi_{i,2})^2$$

$$\begin{aligned}
&\Rightarrow 2A_1 \sin \phi_{i,1} A_2 \sin \phi_{i,2} \\
&= A_1 A_2 \cos(\phi_{i,1} - \phi_{i,2}) - A_1 A_2 \cos(\phi_{i,1} + \phi_{i,2}) \quad (\text{A.2})
\end{aligned}$$

### A.1.2 Third order

For  $n = 3$ , A.1 becomes (again, omitting the terms due to harmonic distortion):

$$\begin{aligned}
&(A_1 \sin \phi_{i,1} + A_2 \sin \phi_{i,2})^3 \\
&\Rightarrow 3A_1^2 A_2 \sin^2 \phi_{i,1} \sin \phi_{i,2} + 3A_1 A_2^2 \sin \phi_{i,1} \sin^2 \phi_{i,2} \\
&= -\frac{3}{4} A_1 A_2^2 \sin(2\phi_{i,2} + \phi_{i,1}) + \frac{3}{4} A_1 A_2^2 \sin(2\phi_{i,2} - \phi_{i,1}) \\
&\quad + \frac{3}{2} A_1 A_2^2 \sin \phi_{i,1} \\
&\quad - \frac{3}{4} A_1^2 A_2 \sin(2\phi_{i,1} + \phi_{i,2}) + \frac{3}{4} A_1^2 A_2 \sin(2\phi_{i,1} - \phi_{i,2}) \\
&\quad + \frac{3}{2} A_1^2 A_2 \sin \phi_{i,2} \quad (\text{A.3})
\end{aligned}$$

## A.2 Levels

Existing literature contains ample discussion on the calculation of intermodulation intercept points, e.g. [24]. However, most text books only deal with interferers of equal power, while this thesis sometimes deals with interferers of unequal power. Therefore, an equivalence formula will be derived here.

For second order intermodulation, equation A.2 shows that all terms are proportional to  $A_1 A_2$ . With two interferers of equal amplitude  $A_{eq}$ , these terms would be proportional to  $A_{eq}^2$  instead. Therefore, all intermodulation calculations can be performed with

$$A_{eq} = \sqrt{A_1 A_2},$$

and therefore

$$P_{eq} = \sqrt{P_1 P_2}. \quad (\text{A.4})$$

For third order intermodulation, a similar derivation holds. As equation A.3 shows, all intermodulation products are proportional to either  $A_1^2 A_2$  or  $A_1 A_2^2$ . For the terms proportional to  $A_1^2 A_2$  (those at  $2f_{i,1} + f_{i,2}$ ,  $|2f_{i,1} - f_{i,2}|$  and at  $f_{i,2}$ ):

$$A_{eq} = \sqrt[3]{A_1^2 A_2},$$

and therefore

$$P_{eq} = \sqrt[3]{P_1^2 P_2}. \quad (\text{A.5})$$

For intermodulation products at  $2f_{i,2} + f_{i,1}$ ,  $|2f_{i,2} - f_{i,1}|$  and at  $f_{i,1}$ ,  $P_1$  and  $P_2$  have to be interchanged.

Using equations A.4 and A.5, intermodulation intercept points can be calculated in the normal fashion.



# Bibliography

- [1] J. Mitola, "The software radio architecture," *IEEE Communications Magazine*, vol. 33, no. 5, pp. 26–38, May 1995.
- [2] R. Schiphorst, "Software-defined radio for wireless local-area networks," Ph.D. dissertation, University of Twente, 2004.
- [3] "Specification of the bluetooth system version 1.1," Tech. Rep., February 2001.
- [4] "ETSI TS 101 475 v1.2.2 (2001-02) technical specification broadband radio access networks (BRAN); HIPERLAN type 2; physical (PHY) layer," Tech. Rep.
- [5] T. Hentschel, M. Henker, and G. Fettweis, "The digital front-end of software radio terminals," *IEEE Personal Communications*, vol. 6, no. 4, pp. 40–46, Aug. 1999.
- [6] N. C. Davies, "A high performance HF software radio," in *Eighth International Conference on HF Radio Systems and Techniques*, 2000, pp. 249–256.
- [7] R. J. Lackey and D. W. Upmal, "Speakeasy: the military software radio," *IEEE Communications Magazine*, vol. 33, no. 5, pp. 56–61, May 1995.
- [8] P. G. Cook and W. Bonser, "Architectural overview of the SPEAKeasy system," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 650–661, Apr. 1999.
- [9] H. Yoshida, S. Otaka, T. Kato, and H. Tsurumi, "A software defined radio receiver using the direct conversion principle: Implementation and evaluation," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2000, pp. 1044–1048.

- [10] K. Muhammad, D. Leipold, B. Staszewski, Y.-C. Ho, C. M. Hung, K. Maggio, C. Fernando, T. Jung, J. Wallberg, J.-S. Koh, S. John, I. Deng, O. Moreira, R. Staszewski, R. Katz, and O. Friedman, "A discrete-time bluetooth receiver in a  $0.13\mu\text{m}$  digital CMOS process," in *Proceedings of the 2004 IEEE International Solid-State Circuits Conference*, 2004, pp. 268–269.
- [11] K. Muhammad, R. B. Staszewski, and D. Leipold, "Digital RF processing: Toward low-cost reconfigurable radios," *IEEE Communications Magazine*, vol. 43, no. 8, pp. 105–113, Aug. 2005.
- [12] Z. Ru, E. A. M. Klumperink, and B. Nauta, "On the suitability of discrete-time receivers for software-defined radio," in *IEEE International Symposium on Circuits and Systems*, 2007, pp. 2522–2525.
- [13] V. J. Arkesteijn, E. A. M. Klumperink, and B. Nauta, "ADC clock jitter requirements for software radio receivers," in *Proceedings of the 2004 IEEE 60<sup>th</sup> Vehicular Technology Conference*, Sept. 2004, pp. 1983–1985.
- [14] —, "Jitter requirements of the sampling clock in software radio receivers," *IEEE Transactions on Circuits and Systems II*, vol. 53, no. 2, pp. 90–94, Feb. 2006.
- [15] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, Jan. 1949.
- [16] B. Razavi, *principles of Data Conversion System Design*. IEEE Press, 1995.
- [17] R. van de Plassche, *Integrated Analog-to-Digital and Digital-to-Analog Converters*. Kluwer Academic Publishers, 1994.
- [18] D. H. Shen, C.-M. Hwang, B. B. Lusignan, and B. A. Wooley, "A 900-MHz RF front-end with integrated discrete-time filtering," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 12, pp. 1945–1954, Dec. 1996.
- [19] M. Shinagawa, Y. Akazawa, and T. Wakimoto, "Jitter analysis of high-speed sampling systems," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 1, pp. 220–224, Feb. 1990.

- 
- [20] M. Patel, I. Darwazeh, and J. J. O'Reilly, "Bandpass sampling for software radio receivers, and the effect of oversampling on aperture jitter," in *Proceedings of the 55th Vehicular Technology Conference*, May 2002, pp. 1901–1905.
- [21] N. Da Dalt, M. Harteneck, C. Sandner, and A. Wiesbauer, "On the jitter requirements of the sampling clock for analog-to-digital converters," *IEEE Transactions on Circuits and Systems I*, vol. 49, no. 9, pp. 1354–1360, Sept. 2002.
- [22] R. van de Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*, 2nd ed. Kluwer Academic Publishers, 2003.
- [23] S. Lindfors, A. Parssinen, and K. A. I. Halonen, "A 3-V 230-MHz CMOS decimation subsampler," *IEEE Transactions on Circuits and Systems II*, vol. 50, no. 3, pp. 105–117, Mar. 2003.
- [24] B. Razavi, *RF Microelectronics*. Prentice Hall, 1998.
- [25] D. C. Lee, "Analysis of jitter in phase-locked loops," *IEEE Transactions on Circuits and Systems II*, vol. 49, no. 11, pp. 704–711, Nov. 2002.
- [26] R. C. H. van de Beek, C. S. Vaucher, D. M. W. Leenaerts, E. A. M. Klumperink, and B. Nauta, "A 2.5-10-GHz clock multiplier unit with 0.22-ps RMS jitter in standard 0.18- $\mu\text{m}$  CMOS," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 11, pp. 1862–1872, Nov. 2004.
- [27] J. van der Tang, D. Kasperkovitz, and A. van Roermond, *High-frequency oscillator design for integrated transceivers*. Kluwer Academic Publishers, 2003.
- [28] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 539–550, Apr. 1999.
- [29] ———, "Performance trends for analog-to-digital converters," *IEEE Communications Magazine*, vol. 37, no. 2, Feb. 1999.
- [30] S. M. Louwsma, E. J. M. van Tuijl, M. Vertregt, and B. Nauta, "A 1.35 GS/s, 10b, 175 mW time-interleaved AD converter in 0.13 $\mu\text{m}$  CMOS," in *Digest of Technical Papers Symposium on VLSI Circuits*, June 2007.

- [31] H. J. De Los Santos, G. Fischer, H. A. C. Tilmans, and J. T. M. van Beek, "RF MEMS for ubiquitous wireless connectivity," *IEEE Microwave Magazine*, vol. 5, no. 4, pp. 36–65, Dec. 2004.
- [32] Z. Wang, X. Ma, and G. B. Giannakis, "OFDM or single-carrier block transmissions?" *IEEE Transactions on Communications*, vol. 52, no. 3, pp. 380–394, Mar. 2004.
- [33] M. A. Margarit, D. Shih, P. J. Sullivan, and F. Ortega, "A 5-GHz BiCMOS RFIC front-end for IEEE 802.11a/HiperLAN wireless LAN," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 7, pp. 1284–1287, July 2003.
- [34] P. Zhang, T. Nguyen, C. Lam, D. Gambetta, T. Soorapanth, B. Cheng, S. Hart, I. Sever, T. Bourdi, A. K. Tham, and B. Razavi, "A 5-ghz direct-conversion cmos transceiver," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, pp. 2232–2238, Dec. 2003.
- [35] I. Vassiliou, K. Vavelidis, T. Georgantas, S. Plevridis, N. Haralabidis, G. Kamoulakos, C. Kapnistis, S. Kavadias, Y. Kokolakis, P. Merakos, J. C. Rudell, A. Yamanaka, S. Bouras, and I. Bouras, "A single-chip digitally calibrated 5.15-5.825-GHz 0.18- $\mu\text{m}$  CMOS transceiver for 802.11a wireless LAN," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, pp. 2221–2231, Dec. 2003.
- [36] Adiseno, M. Ismail, and H. Olsson, "A wide-band RF front-end for multiband multistandard high-linearity low-IF wireless receivers," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 9, pp. 1162–1168, Sept. 2002.
- [37] J. Ryyanen, K. Kivekas, J. Jussila, L. Sumanen, A. Parssinen, and K. A. I. Halonen, "A single-chip multimode receiver for GSM900, DCS1800, PCS1900, and WCDMA," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 4, pp. 594–602, Apr. 2003.
- [38] F. Bruccoleri, E. A. M. Klumperink, and B. Nauta, "Wide-band CMOS low-noise amplifier exploiting thermal noise canceling," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 2, pp. 275–282, Feb. 2004.
- [39] S. Andersson, C. Svensson, and O. Drugge, "Wideband LNA for a multistandard wireless receiver in 0.18  $\mu\text{m}$  CMOS," in *European Solid-State Circuits Conference*, 2003.



- 
- [40] Adiseno, H. Magnusson, and H. Olsson, "A 1.8-V wide-band CMOS LNA for multiband multistandard front-end receiver," in *European Solid-State Circuits Conference*, 2003.
- [41] P. Busson, P.-O. Jouffre, P. Dautriche, F. Paillardet, and I. Telliez, "A complete single-chip front-end for digital satellite broadcasting," in *International Conference on Consumer Electronics*, 2001.
- [42] B.-E. Kim, S.-Y. Kim, T.-J. Lee, J.-K. Lim, Y.-J. Kim, M. S. Jeong, K. Kim, S.-U. Kim, S.-H. Park, and B.-K. Ko, "A CMOS single-chip direct conversion satellite receiver for digital broadcasting system," in *2002 Symposium on VLSI Circuits Digest of Technical Papers*, 2002.
- [43] V. J. Arkesteijn, E. A. M. Klumperink, and B. Nauta, "A wideband high-linearity RF receiver front-end in CMOS," in *Proceedings of the 30<sup>th</sup> European Solid-State Circuits Conference*, Sept. 2004, pp. 71–74.
- [44] F. Bruccoleri, "Wide-band low-noise amplifier techniques in CMOS," Ph.D. dissertation, University of Twente, Nov. 2003.
- [45] W. Redman-White and D. M. W. Leenaerts, "1/f noise in passive CMOS mixers for low and zero IF integrated receivers," in *Proceedings of the 27<sup>th</sup> European Solid-State Circuits Conference*, Sept. 2001.
- [46] R. Velghe, D. Klaassen, and F. Klaassen, "MOS model 9," Philips Nat. Lab., unclassified report NL-UR 003/94, 1994.
- [47] J.-H. C. Zhan and S. S. Taylor, "A 5GHz resistive-feedback CMOS LNA for low-cost multi-standard applications," in *ISSCC Digest of Technical Papers*, 2006, pp. 200–201.
- [48] S. Chehrazi, A. Mirzaei, R. Bagheri, and A. A. Abidi, "A 6.5 GHz wideband CMOS low noise amplifier for multi-band use," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, Sept. 2005, pp. 801–804.
- [49] R. Bagheri, A. Mirzaei, S. Chehrazi, M. E. Heidari, M. Lee, M. Mikhemar, W. Tang, and A. A. Abidi, "An 800-MHz–6-GHz software-defined wireless receiver in 90-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 12, pp. 2860–2876, Dec. 2006.

- [50] S. C. Blaakmeer, E. A. M. Klumperink, D. M. W. Leenaerts, and B. Nauta, "A wideband noise-canceling CMOS LNA exploiting a transformer," in *Digest of Papers IEEE Radio Frequency Integrated Circuits Symposium*, June 2006.
- [51] J. Borremans, P. Wambacq, and D. Linten, "An ESD-protected DC-to-6GHz 9.7mW LNA in 90nm digital CMOS," in *ISSCC Digest of Technical Papers*, 2007, pp. 422–423.
- [52] R. Ramzan, S. Andersson, J. Dabrowski, and C. Svensson, "A 1.4V 25mW inductorless wideband LNA in 0.13 $\mu$ m CMOS," in *ISSCC Digest of Technical Papers*, 2007, pp. 424–425.
- [53] J.-H. C. Zhan, B. R. Carlton, and S. S. Taylor, "Low-cost direct conversion RF front-ends in deep submicron CMOS," in *Digest of Papers IEEE Radio Frequency Integrated Circuits Symposium*, June 2007, pp. 203–206.
- [54] G. Cafaro, T. Gradishar, J. Heck, S. Machan, G. Nagaraj, S. Olson, R. Salvi, B. Stengel, and B. Ziemer, "A 100 MHz - 2.5 GHz direct conversion CMOS transceiver for SDR applications," in *Digest of Papers IEEE Radio Frequency Integrated Circuits Symposium*, June 2007, pp. 189–192.
- [55] T. Maede, H. Yano, T. Yamase, N. Yoshida, N. Matsuno, S. Hori, K. Numata, R. Walkington, T. Tokairin, Y. Takahashi, M. Juji, and H. Hida, "A direct-conversion CMOS transceiver for 4.9-5.95GHz multi-standards WLANs," in *ISSCC Digest of Technical Papers*, 2004, pp. 90–91.
- [56] R. Ahola, A. Aktas, J. Wilson, K. R. Rao, F. Jonsson, I. Hyyrylainen, A. Brolin, T. Hakala, A. Friman, T. Makiniemi, J. Hanze, M. Sanden, D. Wallner, Y. Guo, T. Lagerstam, L. Noguier, T. Knuutila, P. Olofsson, and M. Ismail, "A single chip CMOS transceiver for 802.11 a/b/g WLANs," in *ISSCC Digest of Technical Papers*, 2004, pp. 92–93.
- [57] L. Perraud, C. Pinatel, M. Recouly, J.-L. Bonnot, N. Sornin, F. Benoist, M. Massei, and O. Gibrat, "A dual-band 802.11 a/b/g radio in 0.18 $\mu$ m CMOS," in *ISSCC Digest of Technical Papers*, 2004, pp. 94–95.
- [58] D. Manstretta, M. Brandolini, and F. Svelto, "Second-order intermodulation mechanisms in CMOS downconverters," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 3, pp. 394–406, Mar. 2003.

## List of publications

- [59] Vincent J. Arkesteijn, Eric A. M. Klumperink, and Bram Nauta. High-Q variable bandwidth passive filters for software defined radio. In *Proceedings of The Fourth International Symposium on Wireless Personal Multimedia Communications*, pages 1571–1572, September 2001.
- [60] Vincent J. Arkesteijn, Eric A. M. Klumperink, and Bram Nauta. High-Q variable bandwidth passive filters for software defined radio. In *Proceedings of the 2<sup>nd</sup> PROGRESS workshop on Embedded Systems*, pages 5–7, October 2001.
- [61] Vincent J. Arkesteijn, Eric A. M. Klumperink, and Bram Nauta. Variable bandwidth analog channel filters for software defined radio. In *Proceedings of the 12<sup>th</sup> ProRISC workshop on Circuits, Systems and Signal Processing*, pages 258–261, November 2001.
- [62] Vincent Arkesteijn, Roel Schiphorst, Fokke Hoeksema, Eric Klumperink, Bram Nauta, and Kees Slump. A software defined radio test-bed for WLAN front ends. In *Proceedings of the 3<sup>rd</sup> PROGRESS Workshop on Embedded Systems*, October 2002.
- [63] Vincent J. Arkesteijn, Eric A. M. Klumperink, and Bram Nauta. An analogue front-end architecture for software defined radio". In *Proceedings of the 13<sup>th</sup> ProRISC workshop on Circuits, Systems and Signal Processing*, pages 165–168, November 2002.
- [64] Vincent J. Arkesteijn, Eric A. M. Klumperink, and Bram Nauta. An analogue front-end test-bed for software defined radio. In *Proceedings of the 3<sup>rd</sup> International Symposium on Mobile Multimedia Systems & Applications*, pages 1–7, December 2002.
- [65] L. F. W. van Hoesel, F. W. Hoeksema, R. Schiphorst, V. J. Arkesteijn, and C. H. Slump. Frequency offset correction in a software defined

- hiperlan/2 demodulator using preamble section A. In *Proceedings of the 3<sup>rd</sup> International Symposium on Mobile Multimedia Systems & Applications*, pages 51–62, December 2002.
- [66] Vincent Arkesteijn, Roel Schiphorst, Fokke Hoeksema, Eric Klumperink, Bram Nauta, and Kees Slump. A combined receiver front-end for Bluetooth and HiperLAN/2. In *Proceedings of the 4<sup>th</sup> PROGRESS Symposium on Embedded Systems*, October 2003.
- [67] Vincent Arkesteijn. Software defined radio maakt draadloze communicatie flexibel. *Bits & Chips*, 5(18):42–43, November 2003.
- [68] Roel Schiphorst, Fokke Hoeksema, Vincent Arkesteijn, Kees Slump, Eric Klumperink, and Bram Nauta. a GPP-based software-defined radio front-end for WLAN standards. In *Proceedings of the 2004 IEEE Benelux Signal Processing Symposium*, pages 203–206, April 2004.
- [69] Vincent J. Arkesteijn, Eric A. M. Klumperink, and Bram Nauta. A wideband high-linearity RF receiver front-end in CMOS. In *Proceedings of the 30<sup>th</sup> European Solid-State Circuits Conference*, pages 71–74, September 2004.
- [70] Vincent J. Arkesteijn, Eric A. M. Klumperink, and Bram Nauta. ADC clock jitter requirements for software radio receivers. In *Proceedings of the 2004 IEEE 60<sup>th</sup> Vehicular Technology Conference*, pages 1983–1985, September 2004.
- [71] Vincent J. Arkesteijn, Roel Schiphorst, Fokke W. Hoeksema, Eric A. M. Klumperink, Bram Nauta, and Cornelis H. Slump. A combined analogue+digital software defined radio receiver front-end for Bluetooth and Hiperlan/2. In *Proceedings of the 5<sup>th</sup> PROGRESS Symposium on Embedded Systems*, October 2004. Best poster award.
- [72] Vincent J. Arkesteijn, Eric A. M. Klumperink, and Bram Nauta. A wideband inductorless CMOS front-end for software defined radio. In *Proceedings of the 14<sup>th</sup> ProRISC workshop on Circuits, Systems and Signal Processing*, pages 452–456, November 2004.
- [73] Vincent J. Arkesteijn, Eric A. M. Klumperink, and Bram Nauta. Jitter requirements of the sampling clock in software radio receivers. *IEEE Transactions on Circuits and Systems II*, 53(2):90–94, February 2006.

- 
- [74] Eric A.M. Klumperink, Rameswor Shrestha, Eisse Mensink, Vincent J. Arkesteijn, and Bram Nauta. Polyphase multi-path radio circuits for dynamic spectrum access. *IEEE Communications Magazine*, 45(5):104–112, May 2007.