Analog-Digital Codesign
using
Coarse Quantization

André B.J. Kokkeler

Composition of the Graduation Committee:

Prof. Dr. Ir. A.J. Mouthaan, University of Twente (chairman)
Prof. Dr. Ir. Th. Krol (promoter)
Prof. Dr. Ir. C.H. Slump (promoter)
Dr. Ir. G.J.M. Smit (assistent-promoter)
Dr. Ir. M.J. Bentum, ASTRON
Prof. Dr. Ir. W.C. van Etten, University of Twente
Prof. Dr. Ir. B. Nauta, University of Twente
Prof. Dr. Ir. A.-J. van der Veen, TU Delft

University of Twente
The Netherlands

Group of Computer Architecture,
Design & Test for Embedded Systems.
P.O. Box 217, 7500 AE Enschede, The Netherlands.

Keywords: power amplifiers, digital predistortion, quantization, dithering, radio telescope, tied array adder, analog-digital codesign.

Printed by Ipskamp PrintPartners, Enschede, The Netherlands.
ISBN 90-365-2156-4

ANALOG-DIGITAL CODESIGN USING COARSE QUANTIZATION

DISSERTATION

to obtain
the doctor's degree at the University of Twente,
on the authority of the rector magnificus,
prof. dr. W.H.M. Zijm,
on account of the decision of the graduation committee,
to be publicly defended
on Thursday, April 28, 2005 at 15.00

by

André Bernardus Joseph Kokkeler

born on 9 february 1965,

in Denekamp, The Netherlands

This dissertation is approved by:

Prof. Dr. Ir.	Th.	Krol	(promoter)	and
Prof. Dr. Ir.	C.H.	Slump	(promoter)	and
Dr. Ir.	G.J.M.	Smit	(assistent-promoter)	

Abstract

With regards to electronic systems, two important trends can be observed. The first trend is generally known as Moore's law. According to this law, the digital processing capacity per chip is increasing rapidly; approximately a factor two every 18 months. Another part of the first trend is that the performance increase of integrated linear or analog processing is slow; approximately a factor two every 4.7 years. One of the consequences of these two aspects is that, in systems containing an analog- and a digital part (mixed-signal systems), more and more functionality is shifted from the analog processing domain into the digital processing domain.

The second trend that can be observed is that the rate of data exchange between electronic systems is increasing rapidly, whether this is via tracks on printed circuit boards, wires or wireless. These higher data rates require components with large bandwidth and small noise contributions in the analog part of the electronic system and components running at a high clock frequency and high resolution in the digital part. At the interface of the analog- and digital parts of the system, the requirements from both sides are applicable. For that reason, the design of data converters from analog to digital (ADCs) and from digital to analog (DACs) is an important part of the overall system design. In practice, especially the design of the ADC is demanding.

For a specific set of applications, the requirements for the ADC can be relaxed by reducing the resolution of the conversion from analog to digital. Within these specific applications, signal characteristics rather than instantaneous values of the signal are determined. Reducing the resolution to an extreme extend is called 'coarse quantization'.

When designing systems with an analog- and a digital part, two different disciplines have to cooperate: the analog design discipline and the digital design discipline. The analog- and digital design processes are guided by an Analog-Digital Codesign methodology. In the existing Codesign methodology (the Specify-Explore-Refine-Verify or SERV methodology), the mixed-signal system is separated into an analog- and a digital part in an early stage of the design process. This separation is one of the major design specifications and is generally not reconsidered in the course of the project. However, when considering coarse quantization, system performance is not clear during the functional specification phase of the project because of unknown effects of coarse quantization. For that reason, a design methodology is proposed that allows revision of the functional specification, based on a performance evaluation. This design methodology is originally developed for the design exploration of (digital) hardware and software and is called the 'Y-chart approach'. We use this approach for the design of mixed-signal systems.

Analog-Digital Codesign, guided by the Y-chart approach, leads to mixed-signal systems with reduced costs compared to systems designed with the traditional SERV methodology. The Y-chart approach also enables the use of coarse quantization as an additional design parameter to further reduce costs. This is illustrated by two case studies.

The first case study concentrates on the design of a digital predistorter for Power Amplifiers (PAs) in telecommunication transmitters. In this thesis, we show that coarse quantization is applicable within predistorters, leading to the use of low cost ADCs. Furthermore, if we allow system timing parameters which are part of a functional specification, to be frozen only after a performance evaluation, the costs of the digital processing part can be reduced as well.

In the second case study, we reconsider the design of a part of a Radio Telescope, used for Radio Astronomy. This part is called the Tied Array Adder and it sums signals from different telescopes. Within a Tied Array Adder, digital summation of signals in combination with coarse quantization is applicable but leads to loss of efficiency compared to a Tied Array Adder where summation is done in the analog domain. The system specification can only be frozen after a performance analysis of the overall system because the effects of coarse quantization are not known during functional specification.

Both case studies show that coarse quantization can lead to mixed-signal systems with lower costs but system parameters will change. The explicit reconsideration of functional specifications, facilitated by the Y-chart approach, is therefore essential for the introduction of coarse quantization.

Samenvatting

Met betrekking tot electronische systemen kunnen twee belangrijke trends worden waargenomen. De eerste trend is algemeen bekend als de wet van Moore. Volgens deze wet neemt de digitale verwerkingscapaciteit per chip zeer snel toe: ongeveer een factor twee elke 18 maanden. Een andere waarneming binnen de eerste trend is dat de toename in prestaties van geïntegreerde lineaire- of analoge verwerking traag is: ongeveer een factor twee elke 4.7 jaar. Een gevolg van deze twee aspecten is, dat bij systemen die uit een analoog en een digitaal deel bestaan (mixed-signal systemen) steeds meer functionaliteit verplaatst wordt van het analoge deel naar het digitale deel.

De tweede trend die kan worden waargenomen is dat de snelheid waarmee informatie uitgewisseld wordt tussen electronische systemen zeer snel toeneemt, of dit nu verloopt via sporen op printplaten, kabels of draadloos. Door deze hoge snelheden zijn componenten vereist met grote bandbreedte en lage ruisbijdragen in het analoge deel en componenten met hoge klokfrequenties en hoge resoluties in het digitale deel. Voor het grensvlak van de analoge- en digitale delen zijn de eisen van beide kanten van toepassing. Om die reden vormt het ontwerp van data omzetters van analoog naar digitaal (ADCs) en van digitaal naar analoog (DACs) een belangrijk onderdeel van het systeemontwerp. In de praktijk is voornamelijk het ontwerp van de ADC veeleisend.

Voor een bepaalde verzameling toepassingen kunnen de eisen voor de ADC versoepeld worden door de resolutie van de analoog naar digitaal omzetting te verlagen. Binnen deze toepassingen worden eigenschappen van het signaal bepaald in plaats van de instantane waarde. Het verlagen van de resolutie in extreme mate wordt ’grove kwantisatie’ genoemd.

Tijdens het ontwerp van systemen met een analoog en een digitaal deel moeten mensen vanuit twee verschillende disciplines met elkaar samenwerken: de analoge ontwerpdiscipline en de digitale ontwerpdiscipline. De analoge- en digitale ontwerpprocessen worden geleid door een Analoog-Digitaal Codesign methode. Volgens de bestaande Codesign methode (de Specify-Explore-Refine-Verify of SERV methode) wordt het mixed-signal systeem in een vroegtijdig stadium van het ontwerpproces gesplitst in een analoog en een digitaal deel. Deze splitsing is één van de belangrijkste ontwerpspecificaties en wordt in het algemeen niet heroverwogen gedurende het project. Echter, wanneer grove kwantisatie overwogen wordt, zijn de prestaties van het systeem niet bekend tijdens het functioneel ontwerp door de vooralsnog onbekende effecten van grove kwantisatie. Om die reden wordt een ontwerpmethode geïntroduceerd die het toestaat om functionele specificaties te herzien, gebaseerd op een prestatie evaluatie. Deze ontwerpmethode is oorspronkelijk ontwikkeld voor ontwerp-verkennings studies van (digitale) hardware en software en wordt de 'Y-chart' benadering genoemd. We passen deze benadering toe voor het ontwerp van mixed-signal systemen.

Analoog-Digitaal Codesign, gebruikmakend van de Y-chart benadering, leidt tot goedkopere mixed-signal systemen in vergelijking tot systemen ontworpen via de traditionele SERV methode. De Y-chart benadering staat tevens het gebruik van grove kwantisatie toe als middel om de kosten van het systeem verder te verlagen. Dit wordt geïllustreerd met behulp van twee case studies.

De eerste case studie richt zich op het ontwerp van digitale predistortie van vermogens-versterkers in zenders voor telecommunicatie toepassingen. In dit proefschrift wordt aangetoond dat grove kwantisatie toegepast kan worden binnen predistortie-systemen waardoor goedkope ADCs gebruikt kunnen worden. Als we tevens toestaan dat systeem timing parameters, die een onderdeel zijn van de functionele specificatie, pas bevroren worden na een prestatie evaluatie, dan kunnen de kosten van het digitale verwerkingsdeel ook nog verlaagd worden.

In de tweede case studie heroverwegen we het ontwerp van een deel van een Radio Telescoop die wordt gebruikt voor Radio Astronomie. Dit deel wordt de 'Tied Array Adder' genoemd en het telt signalen van meerdere telescopen bij elkaar op. Binnen een Tied Array Adder kan digitale optelling van signalen in combinatie met grove kwantisatie worden gebruikt hetgeen echter leidt tot verlies van gevoeligheid in vergelijking met een Tied Array Adder waar de signalen worden opgeteld in het analoge domein. De systeem specificaties kunnen pas bevroren worden na een evaluatie van de prestaties van het gehele systeem doordat de effecten van grove kwantisatie niet bekend zijn tijdens de functionele specificatie fase.

Beide case studies tonen aan dat grove kwantisatie kan leiden tot mixed-signal systemen tegen lagere kosten maar dat systeem parameters beïnvloed zullen worden. Het expliciet heroverwegen van functionele specificaties, hetgeen toegestaan is in de Y-chart benadering, is daarom essentieel voor het toepassen van grove kwantisatie.

Acknowledgements

First of all, I would like to thank Thijs Krol for giving me the opportunity to do my Ph.D. He always showed his confidence that I would succeed eventhough I sometimes was more reserved. Kees Slump and Gerard Smit I want to thank for carefully reading my thesis and for sifting all the awful expressions. Without their help, this thesis would have been incorrect, incomprehensible and unreadable. I want to thank Bert Molenkamp for introducing me in the 'way of life' at the university and Lodewijk Smit for the discussions and help on the process of producing a thesis. Actually, I have to show my gratitude to all members of the Embedded Systems group. They all contribute to the atmosphere in which work becomes a very pleasant pastime.

Philips Semiconductors provided me with the results of measurements on real power amplifiers. This way I could verify our approach using realistic data. I would like to thank Joop van de Sluis and Pieter Lok for their cooperation and willingness to share the information with me.

I want to thank my former collegues at Ericsson, both the R&D department in Enschede and the collegues I worked with in Sweden. A special word of thanks goes to Hugo de Graaf. He always inspired me to try and find ways which had not been traveled before.

I want to thank Arnold van Ardenne from ASTRON because the second case study in this thesis is based on his work. Furthermore, he encouraged me to research a selected topic in depth, besides the need to develop multi-disciplinary skills. I want to thank Peter Fridman for sharing his view on the effects of coarse quantization in Tied Arrays. I want to thank André Gunst for the many joyful and clarifying discussions we have had concerning the effects of quantization.

At the end I want to thank Myriam for her loving and unconditional support. Myriam, Tim and Carlijn, you have made this work worth while.

List of Acronyms

AC	Alternating Current
ACI	Adjacent Channel Interference
ADC	Analog-to-Digital Converter
AGC	Automatic Gain Control
AM	Amplitude Modulation
AMS	Analog and Mixed-Signal (extension)
ASTRON	Netherlands Foundation for Research in Astronomy
BE	Back End
BER	Bit Error Rate
CALLUM	Combined Analogue Locked Loop Universal Modulator
CF	Crest Factor
DAC	Digital-to-Analog Converter
DC	Direct Current
DFT	Discrete Fourier Transform
DCB	Digital Continuum Backend
DNL	Differential Non Linearity
DSP	Digital Signal Processor
DSSS	Direct Sequence Spread Spectrum
DZB	Digital Z(= last) Backend
EDGE	Enhanced Data GSM Environment
ETSI	European Telecommunication Standards Institute
EVN	European VLBI Network
FE	Front End
FFT	Fast Fourier Transform
FCC	Federal Communications Commision
GSM	Global System for Mobile communications
IEEE	Institute of Electrical and Electronics Engineers
IF	Intermediate Frequency
INL	Integral Non Linearity
I/Q	In-phase/Quadrature-phase
ITU	International Telecommunication Union

ITU-R	International Telecommunication Union - Radiocommunications Standardization Sector
IVC	IF to Video Conversion
LAN	Local Area Network
LINC	Linear amplification with Non-linear Components
LNA	Low Noise Amplifier
LO	Local Oscilator
LS	Least Squares
LSB	Least Significant Bit
MERLIN	Multi Element Radio Linked Interferometer Network
MP3	MPEG-1/2 Layer-3
MPEG	Moving Pictures Expert Group
MSE	Mean Square Error
NWO	Nederlandse Organisatie voor Wetenschappelijk Onderzoek
OFDM	Orthogonal Frequency Division Multiplexing
PA	Power Amplifier
PM	Phase Modulation
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RAM	Random Access Memory
RF	Radio Frequency
RMS	Root Mean Square
ROM	Read Only Memory
RTL	Register Transfer Level
SLD	Square Law Detector
SERV	Specify Explore Refine Verify
SNR	Signal to Noise Ratio
TAA	Tied Array Adder
TPD	Total Power Detector
UHF	Ultra High Frequency
UMTS	Universal Mobile Telecommunications Service
VCO	Voltage Controlled Oscilator
VLBA	Very Long Baseline Array
VLBI	Very Large Baseline Interferometry
WCDMA	Wideband Code Division Multiple Access
WSRT	Westerbork Synthesis Radio Telescope

Contents

Chapter 1

Introduction

In the 1950s, the invention of the transistor launched an electronic revolution. The transistor is a solid-state device, made from silicon which replaced vacuum tubes. Tubes consist of wires, metal plates, a glass capsule and a vacuum. The transistor is smaller, cheaper and dissipates less heat than a vacuum tube, but can be used in the same way as a vacuum tube.

The first transistors were packaged in separate containers and early ’transistorized’ computers contained about 10.000 transistors. One can imagine that the assembly of a computer consisting of 10.000 or more separate containers is expensive and cumbersome.

Building computers with more than 100.000 transistors became possible after the invention of the integrated circuit. The basic idea is to fabricate an entire circuit, consisting of multiple transistors onto the same piece of silicon, rather than to assemble discrete components made from separate pieces of silicon. The terminals of different transistors are interconnected to form circuits via small tracks of metal. Many transistors can be layed out on a single wafer of silicon. A wafer contains multiple instantiations of the same circuit. The silicon area dedicated to a single circuit is called a ’chip’.

As time went on, it became possible to pack more and more transistors on the same chip. This trend was investigated quantitatively by Gordon Moore in 1965 ([1]). Moore investigated for successive generations of semiconductor technologies, the cost per component for different numbers of components on a single chip. The costs of a single chip is fixed, so the more components can be packed on a chip, the less the costs per component will be. However, as components are added, decreased yields tend to raise the costs per component. Moore discovered that the number of components per chip, leading to minimum component costs was doubling each year and he expected this trend to proceed on the long term. He appeared to be right although the pace has slowed to a doubling every 18 months since the 1970s. This trend has become well-known as Moore’s law.

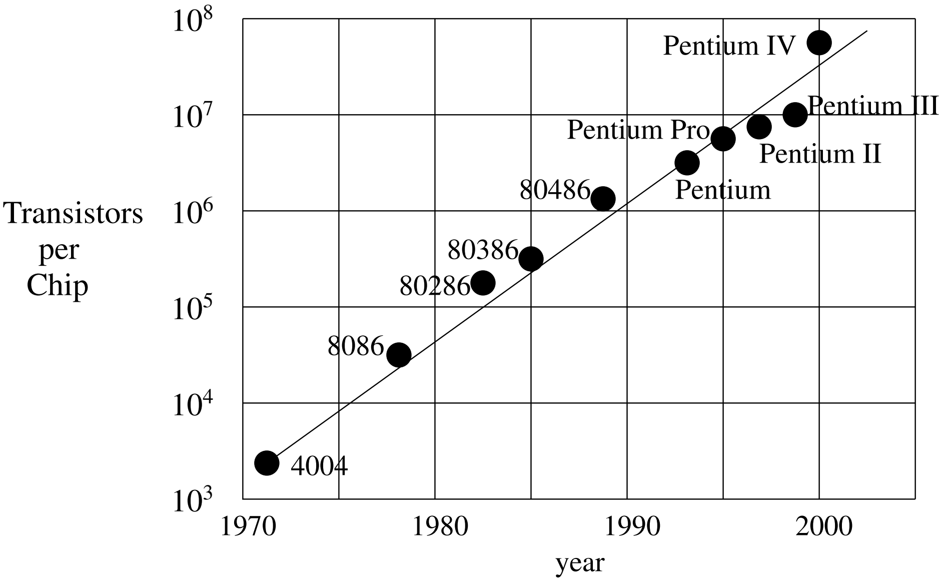


Figure 1.1: Growth of transistor count of the Intel Processor family

Moore's law is illustrated by the evolution of the Intel processor family (see figure 1.1, from [2], p. 30).

The most striking consequences of Moore's law are:

- The same functionality is realized in course of time in smaller packages against lower costs, more reliably, dissipating less power. Furthermore, the computer becomes onmipresent. Computers are built into many, many devices for monitoring and control. Examples are the processing devices used in telecommunication equipment like cell-phones, automotives, identification devices etcetera.

- Because of the miniaturization, logic and memory elements are placed closer together and therefore, the electrical pathlength is shortened, increasing operating speed.

In Moore's article ([1]), a second, less cited, yet important observation (prediction) was presented. Moore's basic conclusions were that integration will not change analog systems as radically as digital systems. Analog systems are systems where signals are continuous, both in time and amplitude. In digital systems, signals are discrete in time and amplitude.

The lack of large-value capacitors and inductors is the greatest fundamental limitation to integrated analog electronics. Moore also stated literally that 'inductors and capacitors will be with us for some time'.

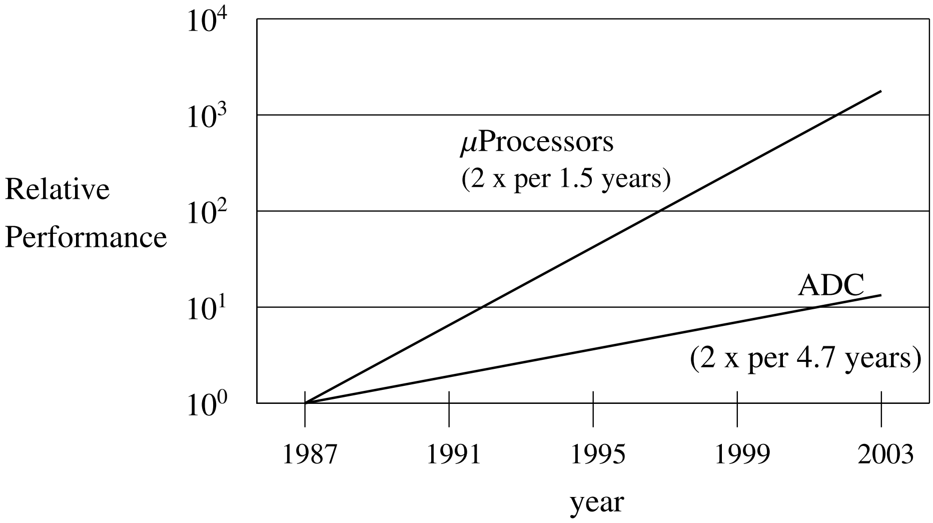


Figure 1.2: Relative performance of microprocessors and ADCs

Moore's second prediction became true as well. Digital performance metrics have grown significantly faster than corresponding metrics for analog circuits, especially Analog-to-Digital Converters (ADCs). From an implementation perspective, the ADC is more demanding than its functional counterpart the DAC (Digital-to-Analog Converter). For that reason, comparisons between the evolution of digital processing capacity and analog processing capacity often use the evolution of the resolution of ADCs as a reference. An overview of performance trends for ADCs is given in [3] and [4]. In figure 1.2, the relative performance of microprocessors and ADCs over the years 1987 to 2003 is presented (from [4]).

The relative performance of the microprocessors is determined by the number of instructions per second that can be executed by a single microprocessor. For the different generations of microprocessors, comparable instructions are considered. The relative performance of ADCs is determined by the resolution of the quantization that can be obtained at a certain sampling speed. In the course of time, chip area, power consumption and production costs are equal.

We see that in a period of 15 years, the relative performance of the ADCs has fallen behind a factor 150. Concluding, we basically see two trends which reflect the difference in evolution between digital and analog performance metrics:

- The performance increase of integrated digital processing is fast; a factor 2 every 18 months.
- The performance increase of integrated linear or analog processing is slow; a factor 2 every 4.7 years.

1.1 Trends and consequences

Many, if not all, applications require analog interfaces beside the digital processing part. Digital Audio equipment like MP3 players, need to generate analog signals to stimulate speakers to generate sound. A mobile telephone has to convert speech into digital data and vice versa. More importantly, however, mobile devices need to send and receive data over a wireless channel via an analog interface. In general, sensors (for example wireless receivers, cd-readers) generate analog signals and actuators (wireless transmitters, cd writers) require analog signals. The information processing, however, is often done in the digital domain. Systems that both have an analog- and a digital part are generally refered to as 'mixed-signal' systems. Below, we will now give a more detailed definition of mixed-signal systems.

1.1.1 Definition of mixed-signal systems

Many electronic systems can be characterized as mixed-signal systems; they consist of a digital and an analog part. Within the digital part, data is represented with limited predefined accuracy. The data is said to have 'discrete-amplitude' (see [5], p. 5). The discrete levels are numbered and generally coded into bits. In practice, the number of quantization levels is limited (say N_{levels}) and can be coded into ${}^2\log(N_{levels})$ bits. The number of bits used for coding the quantization levels is called the 'resolution' of the signal. A time-varying signal generally is a stream of data values which represents the amplitude-discrete data at discrete time intervals (discrete-time).

In the analog part of a mixed-signal system, data can in principle be represented with infinite precision through a voltage or a current (continuous-amplitude). Errors in the signal representation are caused by imperfect behavior of analog components. Two examples of imperfect behavior are non-linear behavior and the addition of noise to the original signal. Non-linear behavior is often modeled as the addition of distortion to a linear component and in many applications this distortion component is modeled as noise. The ratio of the power of the original signal and the noise (Signal-to-Noise Ratio or SNR) is an important measure in analog electronics.

Variations of signals in analog systems continuously result in corresponding variations in the voltage or current (continuous-time).

In mixed-signal systems, data is exchanged between the digital and the analog domain via Analog to Digital Converters (ADCs) and Digital to Analog Converters (DACs), see [5]. An ADC has two tasks: sampling and quantization. Sampling converts the continuous-time signal into a discrete-time signal. Quantization converts the continuous-amplitude signal into a discrete-amplitude signal. An analog signal can be sampled before quantization or quantized before sampling. A DAC generally converts digital data into an appropriate analog current or voltage. The two stages from amplitude-discrete to amplitude-continuous and from time-discrete to time-continuous are generally not identified separately. The number of discrete amplitude levels an ADC or a DAC can identify is called the resolution and is mostly expressed in the number of bits used to encode the discrete levels.

The design of the digital part of a mixed-signal system differs from the design of the analog part of the system. The basic difference is that within digital design, signal quality is generally less affected by operations on the signal. In many applications, the digital representation format is over-dimensioned, liberating the digital designer from continuously monitoring signal quality during the design process. Within the analog design process, the correct representation of the signal constantly is a matter of concern.

1.1.2 Trends in the design of mixed-signal systems

The general trend in the design of mixed-signal systems is to:

- make optimum use of the first part of Moore's observation; the performance in the digital domain increases rapidly.

- avoid the second part of Moore's observation; the performance in the analog domain increases slowly.

This translates into a trend where functionality is shifted as much as possible from the analog domain into the digital domain. Prominent examples of this trend are 'Software Radios' and 'Digital IF'. Software Radio refers to a software reconfigurable radio with programmable radio interfaces and digital baseband signal processing stages. An example of an architecture for advanced software radio platforms is given in [6]. Digital IF refers to the application of digital techniques in the design of wireless receivers and transmitters where the characters 'IF' stand for Intermediate Frequencies. Digital IF is one of the most important enabling technologies of Software Radios. We will concentrate on Digital IF applied in receivers for telecommunication applications.

In figure 1.3 a typical receiver block diagram is presented.

The signal of interest is captured at the antenna and its frequency is generally located in the frequency area ranging from a few MHz to a few GHz. This frequency range is generally indicated as 'RF', which stands for 'Radio Frequencies'. For telecommunication applications, the bandwidth of the signal is in the order of 1 MHz to 10 MHz. This signal is first amplified by a Low Noise Amplifier (LNA). By mixing the output of the LNA with a Local Oscilator signal with a frequency in

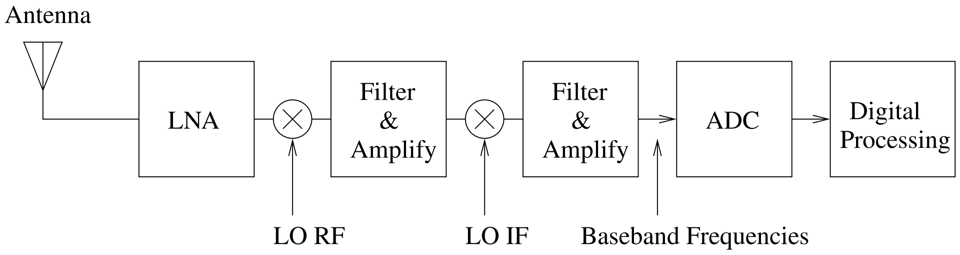


Figure 1.3: Typical superheterodyne receiver block diagram

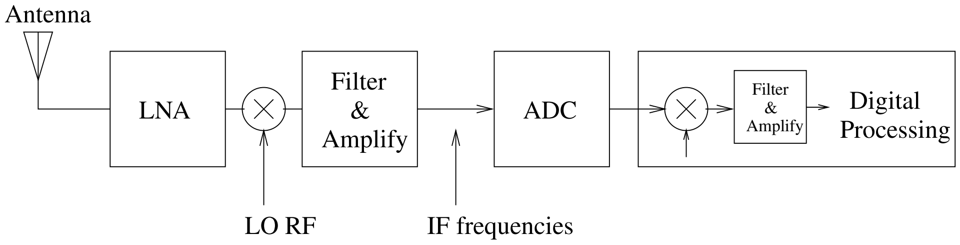


Figure 1.4: Typical receiver block diagram using digital IF

the RF range (LO RF), the signal of interest is downconverted to a frequency range between 10 MHz and 100 MHz. This intermediate frequency range is generally indicated as 'IF frequencies'. The signal is filtered and amplified and a second mixer downconverts the signal to Baseband frequencies, ranging from nearly 0 Hz to about 10 MHz, using a Local Oscilator signal with a frequency in the IF range (LO IF). This two-stage approach, from RF-frequencies to IF frequencies to Baseband, is typical for the so-called 'superheterodyne receiver'. Alternatively, receivers which directly convert from RF frequencies to Baseband frequencies are called 'direct conversion receivers'. Although more complicated from a functional point of view, superheterodyne receivers are preferable from an implementation point of view. We therefore concentrate on superheterodyne receivers. After the last mixing stage, the signal is filtered, amplified and converted into the digital domain by means of an ADC. The signal is then further processed in the digital domain. The signal at the input of the ADC, in general, has a bandwidth in the order of 10 MHz.

When using digital IF, the location of the ADC is shifted towards the analog frontend, see figure 1.4.

The ADC is sampling the signal at IF frequencies. The last conversion stage and necessary filtering is done in the digital domain (see for an example [7]). The signal at the input of the ADC is located at a frequency ranging from 10 MHz to 100 MHz. In general, requirements for the analog part of a mixed-signal system are related to the requirements for the digital part. Because of the

Nyquist/Shannon Sampling theorem (see [8] and [9]), the minimum required repetition rate of samples in the digital domain equals twice the bandwidth in the analog domain. This implies that the analog input bandwidth and the sampling frequency of the ADC have to be increased considerably when using digital IF.

For digital IF applied in transmitters, configurations can be sketched similar to the ones shown in figures 1.3 and 1.4. The functional counterpart of the ADC in the transmitter is the DAC in the receiver. The consequences for the DAC in the digital IF transmitter are similar to the consequences for the ADC in a digital IF receiver. The DAC will have to operate at a higher clocking rate and needs to have a larger analog bandwidth.

Concluding, digital IF requires data converters (ADCs and DACs) with larger bandwidth (higher sampling rate).

Besides the trend to shift more and more functionality from the analog domain into the digital domain, there is another trend which has a major impact on the design of mixed-signal systems. Users require, especially in telecommunication applications, higher data rates for each new generation of equipment. This can be seen from the evolution of the GSM standard to EDGE (see [10]) and UMTS (see [11]). The corresponding maximum data rates are 200 kilobits per second (kbps), 270 kbps and 3.84 Megabits per second (Mbps) respectively. The trend can also be seen from the evolution of the IEEE802.11b standard ([12]) to IEEE802.11g ([13]). The corresponding maximum data rates are 11 Mbps and 54 Mbps respectively.

There are basically two techniques to increase the data rates. The spectral window, assigned to the telecommunication standard, can be increased and/or more advanced modulation schemes can be used. Increasing the spectral window means that a larger part of the scarce electro-magnetic spectrum is required. Therefore, the use of the second technique to increase the data rate, advanced modulation schemes, is often more attractive.

As mentioned before, requirements for the analog part of a mixed-signal system (bandwidth, SNR) are related to the requirements for the digital part (sampling rate, resolution). Concerning the bandwidth/sampling rate, we have already seen that the minimum required repetition rate of samples in the digital domain equals twice the bandwidth in the analog domain. Concerning the SNR/resolution, it is indicated in [5], p. 12, that an increase of the SNR of data converters with 6.01 dB requires an additional resolution in the digital representation of signals of 1 bit.

For both techniques to increase the data rate, increasing the assigned spectral window and advanced modulation schemes, the effects on the analog and digital parts of the system can roughly be indicated. An increased spectral window leads to an increase of the bandwidth of the analog components and to an increased repetition rate of the digitally represented signal. Hence, the data converters (ADC

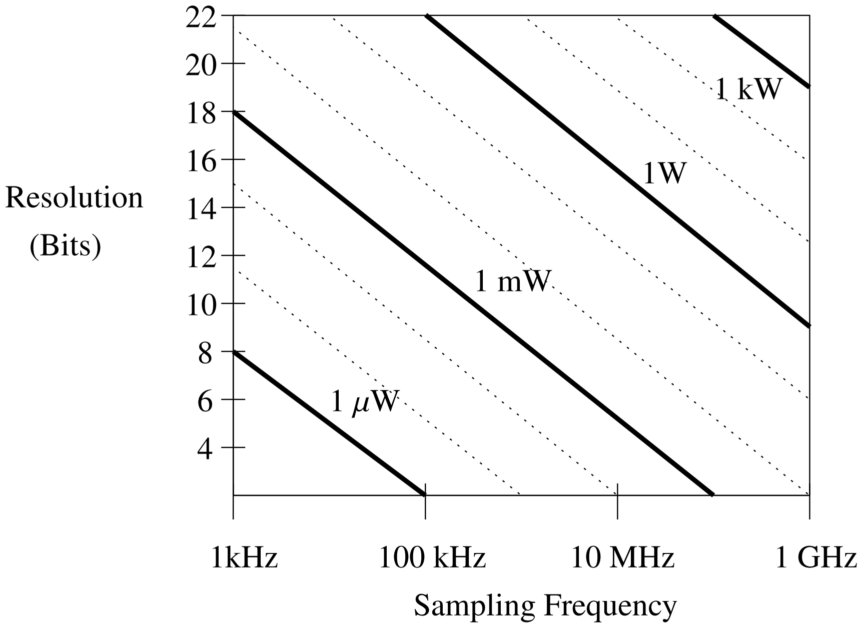


Figure 1.5: Power requirements of ADCs (from [4]).

and DAC) have to operate at higher frequencies. Using more advanced modulation schemes generally requires a larger SNR which means for the analog part of a receiver that mainly the noise injected by the first amplifier stage of a receiver (the LNA) and the non-linearity of the successive stages should be reduced. For the digital part this results in data converters with an increased resolution.

We see that for both options (increasing the spectral window and more advanced modulation schemes to increase data rates), the performance of the data converters has to increase. Besides this, we have already seen that, when introducing digital IF, data converters with increased performance are necessary as well. The ever increasing requirements on data converters are a problem. This is illustrated in figure 1.5.

In this figure, we see the power consumption of ADCs for different resolutions and sampling frequencies. If either the resolution or the sampling frequency increases, the power consumption increases accordingly. A factor of 10 increase in sampling frequency leads to a factor of 10 increase in power consumption while approximately 3 bits additional resolution also leads to a tenfold power consumption. For example, when going from baseband sampling to digital IF, a tenfold increase of sampling frequency is quite common. The corresponding power increase (factor 10) is often not acceptable which makes digital IF not an attractive option in many applications. Similarly, increasing the spectral window for telecommunication applications will lead to higher sampling rates and consequently to higher

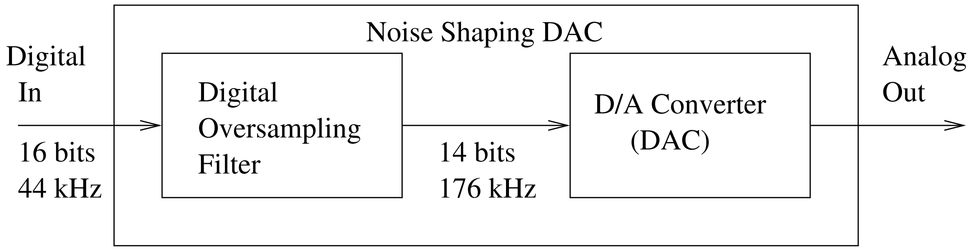


Figure 1.6: Noise shaping DAC

power consumption. Furthermore, more advanced modulation schemes will lead to data converters with more resolution inevitably leading to higher power consumption.

So, currently we cannot introduce digital IF on a large scale, we cannot increase spectral windows significantly and we cannot apply more advanced modulation schemes without a significant increase in power consumption. Furthermore, because of the second part of Moore's observation, we do not expect a significant increase of the performance of data converters in the near future either.

1.2 Digitally assisted analog

According to [4], one way to escape from this situation is 'digitally assisted analog'. The principle behind digitally assisted analog circuits is to delegate analog requirements concerning allowed non-linearity, to the digital domain. The effects of the reduced power dissipation and/or chip-area because of the relaxed analog requirements surpass the effects of the increased power dissipation and/or chip-area in the digital domain. An example of digitally assisted analog circuits are digitally predistorted Power Amplifiers (PAs). By means of digital processing, the inherent non-linear behavior of the analog power amplifier is reduced. Digital predistortion of Power Amplifiers will be explained in more detail in section 1.6.1. Other prominent examples of digitally assisted analog circuits are noise shaping DACs and sigma-delta ADCs (see [5], Chapter 10). The basic idea is to exchange accuracy with speed. An example of a noise shaping DAC is given in figure 1.6 (from [5], p. 369).

The final DAC in figure 1.6 is preceeded by a digital oversampling filter. Within this filter, additional samples, 'between' the original samples are determined by interpolation, increasing the effective sampling rate. The number of bits, used for the representation of the digital values (the resolution), is decreased which increases noise level. However, because of the increased sampling rate, out-of-band noise can be removed by means of clever digital filter techniques.

The oversampled digital signal is then converted into an analog signal by means of a 14 bits DAC. The component count of a 14 bits DAC is lower than the component count of a 16 bits DAC which outweighs the increased digital complexity. In a sigma-delta ADC, a similar but reverse approach is used. The analog signals are coarsely quantized at high frequency and a successive digital signal processing stage, called a low-pass decimator, reduces the sampling frequency while increasing the resolution. We will not explain the noise-shaping DAC and sigma-delta converters in more detail (see [5]).

1.3 Coarse quantization

In the sigma-delta ADC and noise-shaping DAC examples, the repetition rate of samples is increased and the resolution (number of bits) is decreased. Increasing the bandwidth while reducing the resolution is necessary in applications where sampled data is used to represent instantaneous values of signals. For example, the instantaneous amplitude of a real-valued signal or the instantaneous amplitude and phase of a complex-valued signal. In some receiver applications, knowledge concerning the instantaneous values of a signal is not of utmost importance. In these applications, only *characteristics* of a signal need to be determined. An example of such an application is a square law detector (see [14], p. 32), also called Total Power Detector (TPD). The output of the TPD at a specific moment in time is proportional to the average power in a timeslot that ends at that specific moment. A more general application is the determination of the power spectral density of a signal within a certain time slot. In these applications, the resolution of the ADC might be reduced without increasing the sampling rate. Refering to figure 1.5, we see that this is attractive from a 'power consumption point of view', especially if the resolution can be reduced drastically. In the remainder of this thesis, we will refer to quantization where the resolution has been reduced drastically (say below 5 bits) as coarse quantization.

Because of the quantization of analog signals into digital signals, an error is introduced. This error is called 'quantization error'. The power of the quantization error depends on the resolution of the ADC. If the resolution is high, the power of the quantization error is low. In case of coarse quantization, the power of the quantization error is relatively high. Another effect is that in case of high-resolution quantization, the quantization error has no correlation to the original signal. In case of coarse quantization, there is a relation between the original signal and the quantization error. This is illustrated in figure 1.7.

In the left column we see, from top to bottom, the original sinusoidal signal, the quantized signal and the quantization error, in case of 16 bit quantization. We see that the quantization error shows noise-like behavior and is therefore often

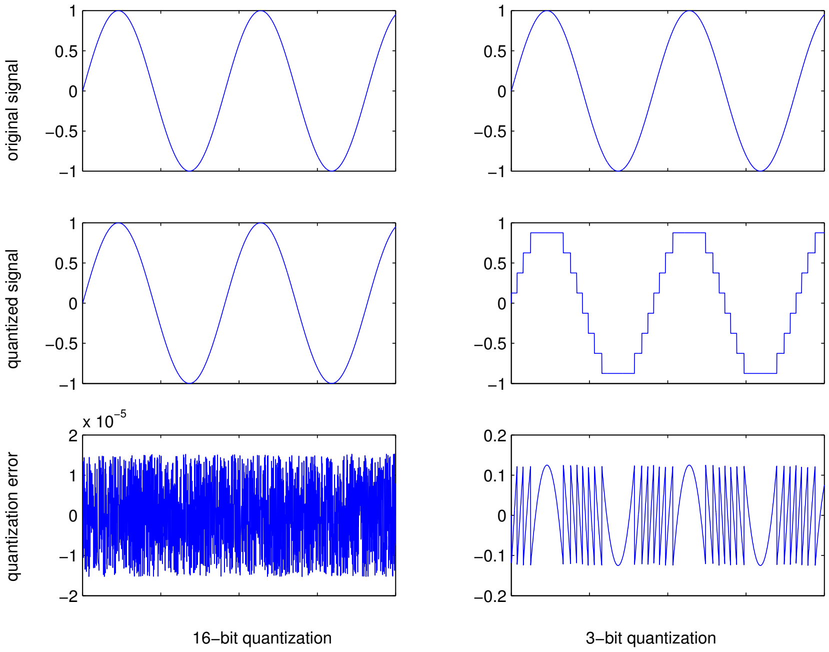


Figure 1.7: Effects of quantization for different resolutions

called quantization noise. One of the main properties of the quantization noise is that it is uncorrelated with the original signal. There is no correspondence or resemblance with the original signal. So, quantization just adds noise to the original signal and there are standard rules for calculating the SNR of the signal after quantization (see section 5.3.3.1). In the right column of figure 1.7, we again see the original signal at the top and below we see the quantized signal and the quantization error in case of 3 bit coarse quantization. If we concentrate on the quantization error, it is clearly visible that there is correlation between the error and the original signal and that the error is not a noise-like signal. Because of this effect, coarse quantization introduces errors in determining signal characteristics. For example, if a Total Power Detector is preceded by a coarsely quantizing ADC, the measured power level will not be correct. So, reducing the resolution will basically have 2 effects (see for an example section 5.3.3.1):

- The noise on the measured signal characteristic will increase.
- There will be a systematic error on the measured signal characteristic.

Depending on the application, the increased noise level might be acceptable and the systematic error might be known. In these cases, reducing the resolution of the ADC simplifies its design resulting in smaller chip areas and consequently, lower production costs. Reducing the resolution also reduces the power consumption as can be see from figure 1.5. From this figure it is clear that reducing the resolution as much as possible, leads to the lowest possible power consumption.

In many applications, the power of the quantization errors should be small compared to the original signal. In these cases the resolution should be sufficiently high. For example, quantization of audio signals should be done with relatively high resolution (14 bits) to preserve quality. For these applications, coarse quantization is not an option.

For applications where increased noise levels are acceptable, coarse quantization might be an option. However, the systematic error introduced by coarse quantization needs to be known. For many applications these effects are not known and are not simple to deduce. For a very specific application, crosscorrelation of single-bit quantized noise signals, the effects of coarse quantization are known (see [15]) and coarse quantization can be exploited.

The difficulty in analyzing its effects has prevented the application of coarse quantization without increasing the sampling rate, on a large scale. There is one major exception: Radio Astronomy receivers. Within Radio Astronomy receivers, the signals to be quantized show noise-like behavior and the effects of coarse quantization can be adequately determined. Furthermore, the effects of the increased noise levels can be reduced by longer integration times (using more samples). An example is the receiver system of the Westerbork Synthesis Radio Telescope which is equiped with 2-bit quantizers (see section 8.1).

Summarizing, for many applications, coarse quantization without increasing the sampling rate is not an option because increased noise levels are not tolerable. For those applications where increased noise levels might be acceptable, the effects of coarse quantization on a system level are not always exactly known and are difficult to analyze.

On the other hand, coarse quantization has significant advantages from a realization point of view like lower production costs and less power dissipation. So, from this point of view, it is desirable to, at least, consider coarse quantization. Because coarse quantization has effects at system level, it has to be considered during the system design. A system design methodology should therefore facilitate an explicite analysis of coarse quantization.

1.4 Design methodologies

The purpose of this section is to identify a design methodology that facilitates the explicit analysis of coarse quantization at system level. A design methodology is defined as a working set of design methods and use of associated design tools ([16]).

The increasing system complexity and the commercial emphasis on shorter design times has led to system design methodologies and supporting tools. Systems often consist of an analog part and a digital part and the digital part on its turn consists of hardware and software. So, during the system design, three different expertises (analog, digital hardware and software) are involved. The basic goal of a design methodology is to guide the activities within the three expertise areas in such way that it leads to an optimal design using the available resources efficiently. Basically these methods assist to determine how the system functionality should be divided over the analog part, the digital hardware and software.

In the early 1990s, a need for a design methodology and accompanying tools arose for systems whose functionality is determined by (digital) hardware and software. Several methods and corresponding tools were developed and they are labeled as Hardware-Software Codesign methods and tools. A good overview of developments within Hardware-Software Codesign is found in [17].

More recently, the first steps towards Analog-Digital Codesign are taken. The difficulty in Analog-Digital Codesign is to cosimulate the analog and digital parts. This is even more difficult in case sections of the analog part operate at RF frequencies. Computer Aided Design tool vendors are developing tools for Analog-Digital Codesign. Mentor Graphics has announced a tool for co-simulation of the analog and digital parts. Cadence recently presented a platform which acts as a base for different tools from both the analog and digital domain. The platform eases the exchange of information between the different domains. These tools support a more or less bottom-up approach of Analog-Digital Codesign. First a design is created and the tools support the analysis of the created design. Based on the results, the design is adapted. This can be iterated until the design meets the specifications.

A top-down approach or system-level synthesis, was first developed for Hardware-Software Codesign (see [18] and [19]) and recently extended to mixed-signal systems (see [20], [21], [22] and [23]). Within Hardware-Software Codesign, there are tools and languages for the specification, allocation and partitioning of functionality onto hardware and software. Allocation is defined as the 'selection of the type and number of components as well as the determination of their interconnection in such way that the functionality of the system can be implemented, the constraints are satisfied and the objective cost function is minimized' ([18]). Partitioning defines 'the mapping between the set of behaviors in the specification and the set of allocated components in the selected architecture' (also [18]). In the remainder of this thesis, we will refer to 'Allocation' as 'Architecture Specification' and to 'Partitioning' as 'Mapping'.

One of the main topics within the top-down approach is to describe the system behavior in a structured and abstract way, more precisely, in a single language. Describing functional behavior in software languages biases the implementation in favor of software, while hardware languages like Verilog and VHDL bias results toward hardware ([17]). New system-level design languages like SystemC ([22]) have been developed to overcome these problems. SystemC is a set of C++ classes and methods that provides a means to describe the structure and behavior of hardware/software systems from abstract to register transfer level (RTL) models. The most powerful feature of SystemC is that it enables the description of concurrency and parallelism. However, SystemC lacks a standard support for analog and mixed-signal systems, or more generally, for the modeling and simulation of continuous-time systems and so of mixed continuous-time/discrete-event systems. For that reason a SystemC-AMS Study Group has been formed to develop analog and mixed-signal (AMS) extensions to SystemC. This work has begun only recently and the result of a first attempt to develop a consistent set of functional specifications for SystemC-AMS are reported in [23].

Summarizing, (digital) Hardware-Software Codesign is an expertise which has somewhat matured since the early 1990s and tools are available. Analog-Digital Codesign is an expertise area which still has to be opened and first initial steps to develop tools have been made.

Until now we have basically concentrated on (simulation) tools and tools for top-down system level synthesis. We have not yet dealt with design methodologies. The different methodologies can basically be categorized into two approaches: the sequential approach and the iterative approach. The sequential design methodology is described in [16], [19], [23] and [21]. The iterative design methodology is described in [24], [25] and [26]. Both methodologies will be described below.

1.4.1 Sequential design methodology

The sequential design methodology is using the Specify-Explore-Refine (SER) design paradigm ([19]). In [20] an explicit verification stage after the refinement is proposed. The resulting methodology is therefore using the Specify-Explore-Refine-Verify (SERV) paradigm. The four different steps are explained below:

- Specify: The functionality of the aimed system is decomposed creating a conceptual model. Preferably, a language should be used which facilitates simulation and verification of the system. This process results in a functional specification which does not contain implementation details.

- Explore: In the exploration phase, design alternatives are explored. The initial specification is transformed into an implementation oriented specification. System components are allocated and their performance metrics are defined. These performance metrics relate to both functional and non-functional metrics like costs, energy and execution time. Then the functional specification is partitioned among the system components.

- Refine: The initial specification is refined. This refined specification is based on the design decisions taken in the exploration phase and the consequences of implementation details on the functional behavior is reflected. The aim is a system-level description which is mainly functional, posessing those characteristics that result from implementation choices. The functional behavior of this refined model should satisfy the functional requirements of the intial model.
- Verify: In the verification phase, the functional behavior of the refined model is checked against the functional requirements of the initial model.

The SERV activities are followed by the final hardware- and software design. Tools are developed to support this approach for Hardware-Software Codesign ([19]) and for Analog-Digital Codesign ([20], [21] and [22]).

The SERV paradigm imposes a more or less top-down approach during the system level design phase and the design exploration. It enforces to complete and freeze the specification before the exploration phase can start. The final stage is a pure verification stage. It is meant to check whether the resulting specifications meet the primary specification. According to this methodology, there is no explicit adaptation of the original (purely functional) specifications, based on the exploration results. However, a very good solution which nearly satisfies all requirements is very often acceptable. In practice, the initial requirements will then be stretched. The SERV paradigm does not support this explicit feedback to the initial requirements. It therefore is more suited for the design of systems with a fixed application and stringent requirements.

1.4.2 Iterative design methodology

An approach which explicitely addresses the possibility to jointly optimize the requirements of the application and the implementation, is the Y-chart approach (see [24], [25] and [26]). The Y-chart approach is illustrated in figure 1.8.

A clear distinction is made between applications and architectures ([26]). By means of an explicit mapping stage, the target application is mapped onto the candidate architecture. The performance of a mapped application is evaluated quantitatively in the performance analysis stage, resulting in performance numbers. Based on the performance numbers, the designer may decide to change either the architecture, the requirements of the application or the mapping, after which a

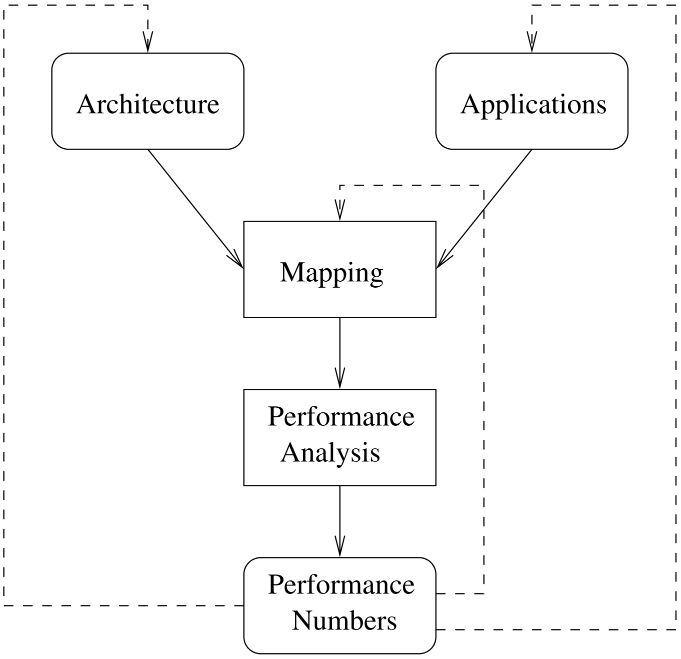


Figure 1.8: The Y-chart approach

new performance analysis will follow. This approach is also useful in cases where a platform (architecture) has to be defined, suitable for a set of applications.

1.4.3 Comparison of design methodologies

Within both methods, the SERV paradigm and the Y-chart approach, the necessary steps for the system level design are described and it is indicated in which order these steps have to be taken. Some steps are not explicitely mentioned in one methodology while they are mentioned in the other (e.g. architecture specification is mentioned in the SERV paradigm but not in the Y-chart approach) and some similar activities are given different names (e.g. the allocation process in the SERV paradigm relates to the mapping process in the Y-chart approach).

To express the difference between the two approaches, we have to represent the SERV and Y-chart approach in a uniform way. This is illustrated in figure 1.9.

On the right upper side we see the specify phase (SERV paradigm). It consists of a ’Functional Specification’ activity which results in an ’Application’ (Y-chart approach). On the left upper side we see the explore phase (SERV). An ’Architecture Specification’ activity (SERV) results in an ’Architecture’ (Y-chart). The ’Application’ is mapped onto the ’Architecture’ (Y-chart). As already mentioned, ’Mapping’ is similar to ’Allocation’ (SERV). A ’Performance Analysis’

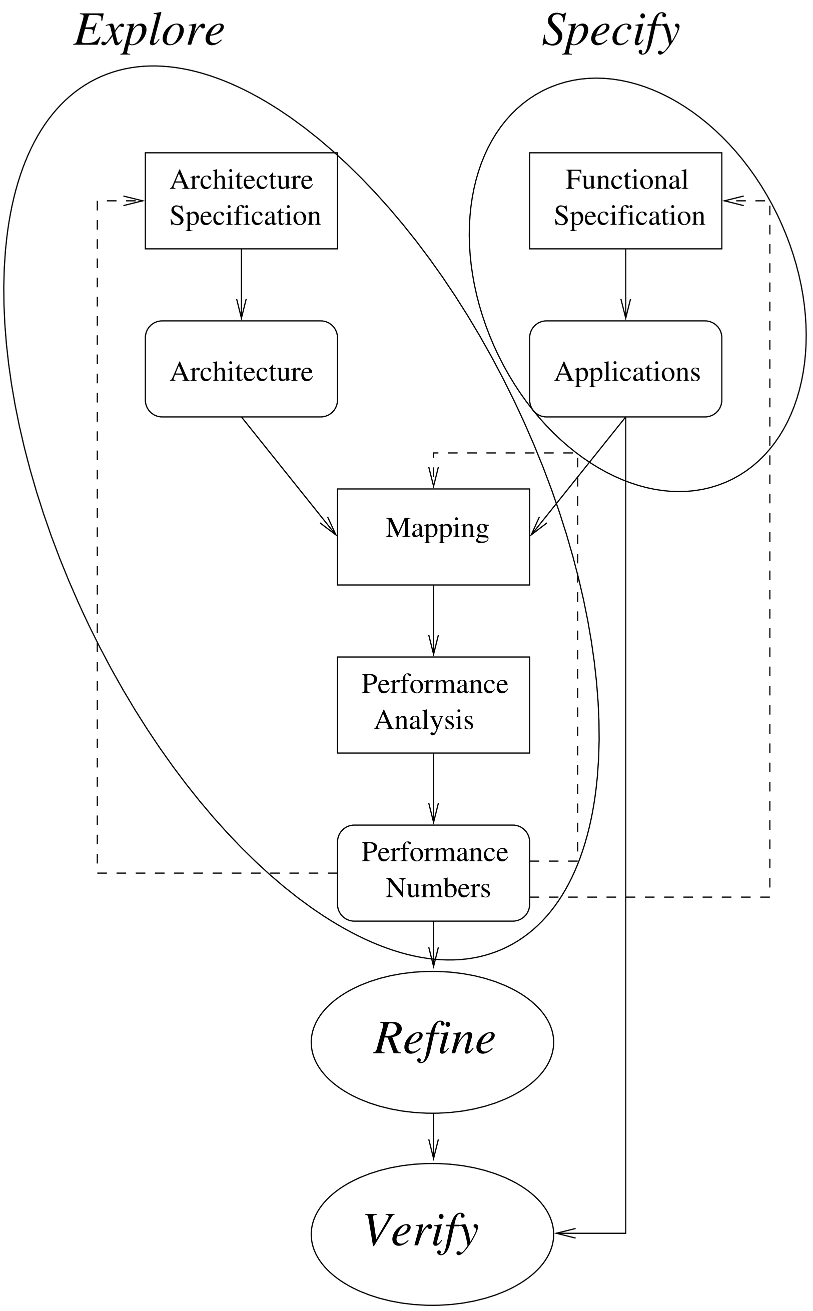


Figure 1.9: Uniform representation of the SERV paradigm and Y-chart approach

of the mapped application results in ’Performance Numbers’. According to both approaches, the ’Architecture Specification’ and ’Mapping’ stages of a next iteration might be influenced by ’Performance Numbers’. However, within the Y-chart approach, explicit feedback from ’Performance Numbers’ to the ’Functional Specification’ activity is allowed while this option is not indicated in the SERV paradigm. The Refine and Verify phases (SERV) are not addressed explicitely in the Y-chart approach. The SERV paradigm is therefore more tailored towards system level design and the Y-chart approach towards system level design exploration.

1.4.4 Analog-Digital Codesign methodologies

Until now, we have dealt with design methodologies in a very general sense. The question that arises is: How can the two design methodologies support the design exploration of mixed-signal systems and more specifically, of mixed-signal systems using coarse quantization?

The system level design of mixed-signal systems using the SERV paradigm is described in [20]. It is proposed to first partition the application into an analog- and digital part and then start two ‘explore’ phases, one for the analog part and one for the digital part. This is indicated in figure 1.10.

The additional ’Analog Digital Partitioning’ activity determines which part is realized in the digital domain and which part in the analog domain. In the design of the analog part, the quality of the signals has to be considered continuously. The quality is characterized by the Signal-to-Noise Ratio (SNR) of the signal and the spectral contents. In the design of the digital part of a mixed-signal system, the quality of the signal representation is of much less concern. The aspect that is important for the representation of signals is the number of bits because performance characteristics like power consumption, chip area and costs, depend on it. In many applications, the representation format is over-dimensioned, liberating the designer from continuously monitoring signal quality during the design process. For that reason, two separate design disciplines have arisen: analog design and digital design. The sequential design methodology using the SERV paradigm is based in these two separate design disciplines. The resolutions of the ADCs and DACs in the mixed-signal system have to be determined based on the Analog Digital Partitioning. The resolution of the converters is not an explicit design parameter which can be tuned to optimize the result of the mapping phase. In practice, it will be tempting to use converters with (too) high resolution to avoid the influence of quantization. Furthermore, there is no explicit feedback from the analog and digital performance numbers to the Analog Digital Partitioning and functional specification, which prevents the analysis of quantization effects. Another disadvantage of the SERV paradigm is that there are separate exploration phases for

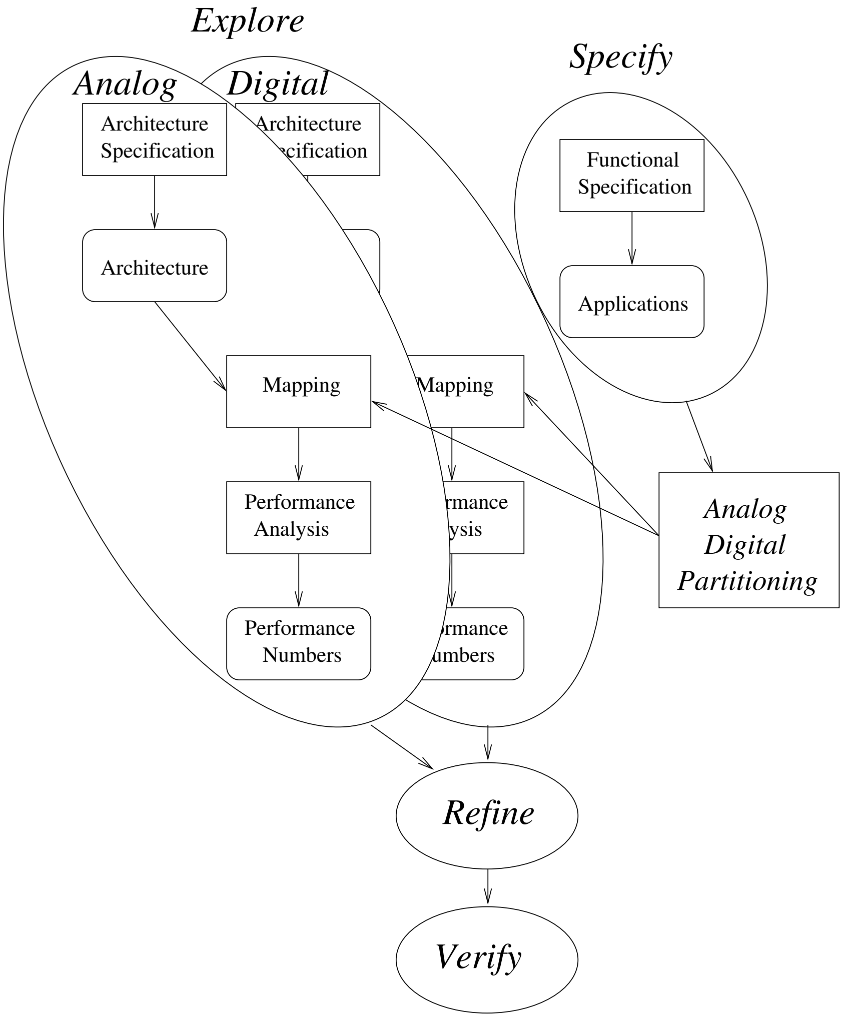


Figure 1.10: The SERV paradigm for mixed-signal systems

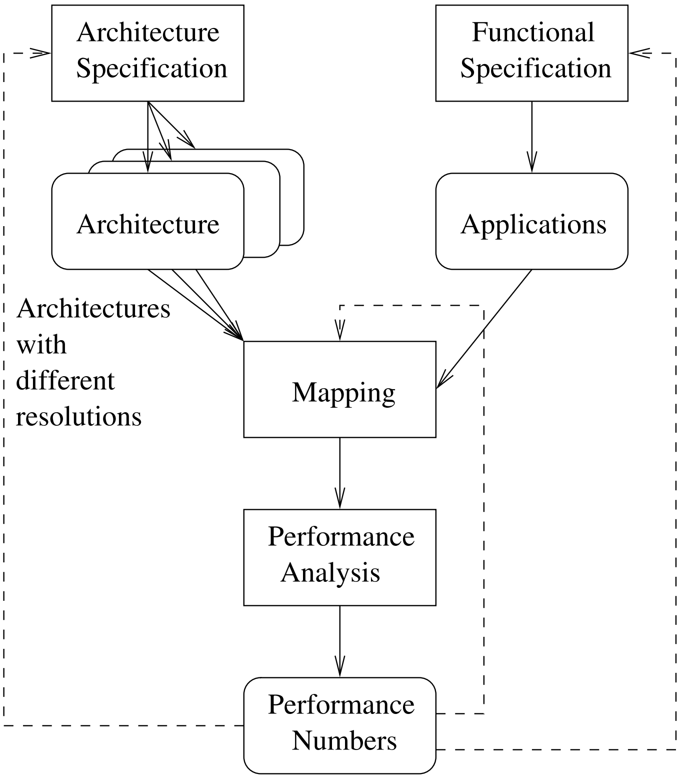


Figure 1.11: Analysis of coarse quantization using the Y-chart approach

the analog and digital design. Within these phases, there is optimization, based on performance numbers. However, there is no explicit coupling between the analog- and digital design. This prevents the joint optimization of the analog- and digital parts. Our conclusion is that the SERV paradigm does not explicitely encourage the use of coarse quantization. For that reason, we do not use the SERV paradigm to investigate the usefulness of coarse quantization in mixed-signal systems.

Within the Y-chart approach, architecture specification and mapping are pure system design activities where analog- and digital design teams have to *cooperate* to define the alternative architectures and to map the functionality onto these alternatives. In the Y-chart approach, the analog and digital design teams are forced to cooperate instead of being coexistent. The system level design of mixed-signal systems truly becomes Analog-Digital Codesign.

When using the Y-chart approach in combination with coarse quantization, the resolutions of ADCs and DACs can be choosen as explicit design parameters. Using different resolutions in the architecture specification activity results in alternative architectures (see figure 1.11). These architectures can be analyzed within one iteration of the Y-chart approach.

The application can then be mapped onto the set of architectures. After the performance analysis, the designer can decide to change the set of architectures, change the specification of the application and/or change the mapping.

Based on the previous analysis we can draw two conclusions:

- The Y-chart approach explicitely supports the analysis of the effects of coarse quantization in the Analog-Digital Codesign process.

- The Y-chart approach allows feedback from performance numbers to the functional specification. During the architecture specification (in which different resolutions are chosen, resulting in different architectures) the effects of coarse quantization are not yet clear and it is not clear whether the specifications of the application will be met. An architecture which nearly satisfies all requirements might be acceptable which means that, based on the performance numbers, the application has to be respecified.

1.5 Problem Statement

In section 1.1, we have identified that the performance of mixed-signal systems is limited by the performance of data converters, especially Analog-to-Digital Converters. We have also seen that 'Moore's law' will certainly not solve this problem. On the contrary, the performance of digital systems will increase more rapidly than the performance of the analog system.

For specific applications, there is a way to escape from this situation: coarse quantization without increasing the sampling rate, see section 1.3. These specific applications are characterized by the fact that their aim is to determine accurate signal characteristics rather than using accurate digital representations of instantaneous values of signals.

In section 1.4 we have identified an Analog-Digital Codesign methodology, the Y-chart approach, which supports the design exploration of mixed-signal systems using coarse quantization.

The main hypothesis of this thesis is that coarse quantization is a useful technique to improve the performance of mixed-signal systems whose aim is to determine signal characteristics. We will illustrate the usefulness of coarse quantization by means of two case studies. In these case studies, we will do a system level design exploration using the Y-chart approach.

1.6 Introduction of the case studies

The two case studies are: the design of a digital predistorter for a Power Amplifier and the design of a digital Tied Array Adder (TAA) for a multi-element Radio Astronomy receiver.

1.6.1 Case study 1

Modern communication systems use sophisticated modulation techniques like QPSK, 64-QAM or multicarrier configurations (e.g. OFDM) in order to obtain high data rates and spectral efficiency. These modulation techniques require highly linear transmitters and receivers. Especially basestation-transmitters have to satisfy stringent linearity requirements. In basestation transmitters, signals from many users are multiplexed onto a single channel, leading to a signal with large peak signal excursions compared to the average signal level (large peak-to-average ratios). For terminals, this is less severe since the signal of only one user has to be transmitted leading to smaller peak-to-average ratios. In this case study, we therefore concentrate on basestation-transmitters. Designs of basestation-transmitters are generally expensive and power-inefficient. The last stage of a transmitter is the Power Amplifier (PA). The output power of a PA is relatively high. For example, for a PA in a UMTS basestation, typical output powers range between 20 to 50 Watts, where the PA gain is about 25 dB. A PA ideally has a linear transfer characteristic; the output is a constant factor times the input. In reality, a PA shows non-linear behaviour. When a non-linear PA is excited with a modulated carrier there are two effects. First, the spectrum of the signal will be expanded into adjacent channels causing Adjacent Channel Interference (ACI). Adjacent channels are those parts of the spectrum, allocated to other users. Second, spurious signals within the band of interest will complicate the detection of original signals at receivers which results in higher Bit Error Rates (BER). As an example, two spectra are presented in figure 1.12.

The relative power spectral densities of the original signal at the input of a non-linear amplifier and of the distorted signal at the output are given as a function of frequency, relative to the carrier. When looking carefully at the power within the original signal band (the primary channel), a small difference between the two spectra can be seen. Within the Adjacent channels, the increased power levels can easily be seen.

To limit the distortion, several linearization techniques exist. One of these techniques is predistortion. In figure 1.13 we consider a simplified model of predistortion in combination with a PA.

V_{in} is the input voltage of the predistortion-amplifier combination, V_{out} the output voltage. G is the linear gain of the amplifier and $g(V)$ the non-linear transfer

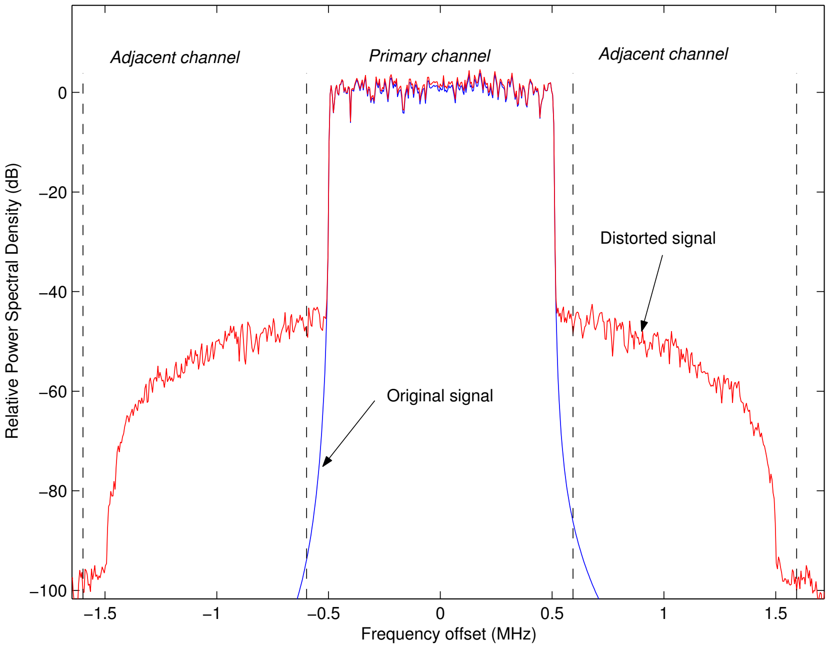


Figure 1.12: Spectra of an original and distorted signal

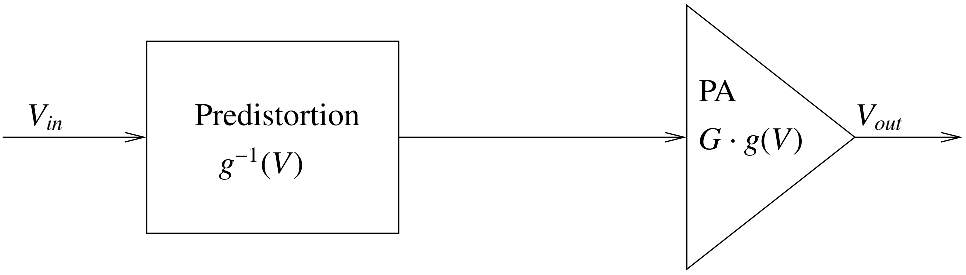


Figure 1.13: Simplified model of predistortion in combination with a PA

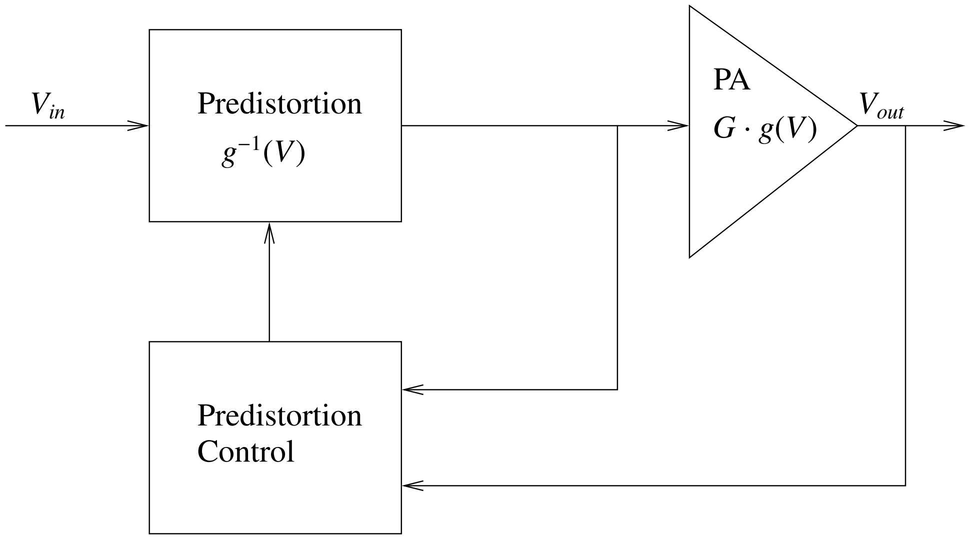


Figure 1.14: Predistortion with feedback

function (distortion). The general idea of predistortion is to distort the signal V_{in} with the inverse of the amplifier distortion $g^{-1}(V)$. The combination of predistortion and non-linear amplification by the PA results in linear amplification:

$$V_{out} = G \cdot g\left(g^{-1}(V_{in})\right) = G \cdot V_{in} \tag{1.1}$$

Unfortunately, the non-linearity of the PA changes in time e.g. due to temperature changes and aging. For that reason, during operation, the input and output of the PA have to be continuously monitored to track the changes in its behavior (see figure 1.14).

By comparing the signal at the input of the PA with the signal at the output, the non-linearity of the PA is continuously estimated and the predistorter is updated accordingly. There are several algorithms to estimate the non-linearity of the PA. Two recent examples can be found in [27] and [28]. These algorithms deal with predistortion in the digital domain. The Predistortion and Predistortion Control blocks in figure 1.14 are realized by digital logic. Since the PA operates on analog signals, the signals from the Predistortion block to the PA have to be converted from the digital domain into the analog domain and the output of the PA has to be converted from the analog domain into the digital domain. The algorithms mentioned above were designed without paying attention to the design of the feedback path and the ADC within it.

In this case study, we do not concentrate on the Predistortion block. It generally is implemented as a look-up table in Random Access Memory (RAM). We concentrate on the complexity of the ADC in the feedback path and on the Digital

Signal Processing algorithm in the Predistortion Control block which estimates the behavior of the PA and which has to calculate the settings of the Predistortion block. Our aim is to improve the overall performance of digital predistortion by introducing coarse quantization in the feedback path. Because the effects of coarse quantization cannot be determined using standard rules, a thorough analysis needs to be done and some of the system characteristics can only be determined after the analysis. We therefore use the Y-chart approach to guide the system level exploration. In this case study, we iterate through the Y-chart two times. In the first iteration we use the algorithm described in [27]. To improve performance numbers of the digital part of the system, we design a new digital signal processing algorithm, the Crosscorrelation predistorter. This algorithm is used in the second iteration of the Y-chart approach.

1.6.2 Case study 2

Our second case study has its origin in a completely different application domain: Radio Astronomy receivers. Telecommunication and Radio Astronomy have opposite interests concerning the use of the electromagnetic spectrum. The spectrum available to the operators of telecommunication networks, determines the data that can be transported. The more spectrum that is available, the more data can be transported leading to higher turnovers of the operators and telecommunication equipment manufacturers. If these companies are managed decently, this should lead to higher profits.

The interests of the telecommunications industry conflicts with the interest of Radio Astronomy. The more spectrum there is available to telecommunication, the less spectrum there is available to Radio Astronomy. Spectrum allocation matters are discussed at World Radio Conferences, usually every two years. The conferences are part of the ITU-R (International Telecommunication Union - Radiocommunications Standardization Sector, [29]). Furthermore, maximum interference levels like Adjacent Channel Interference levels, are specified, also protecting Radio Astronomy observations.

Another difference between the first and the second case study is that in the first case study we investigate the design of a linear transmitter and in the second case study we investigate the design of a linear receiver.

Within Radio Astronomy, radio emissions (roughly wavelengths between millimeters and meters) are used to obtain images of celestial objects. Radio emissions are captured by Radio telescopes. There are several ways to characterize Radio telescopes. For the scope of this thesis, we identify two groups: single dish and multiple-dish. For observations at a certain frequency, single dish telescopes process the signal from a single receiver while multiple dish telescopes combine signals from multiple receivers. Single dish telescopes are, for exam-

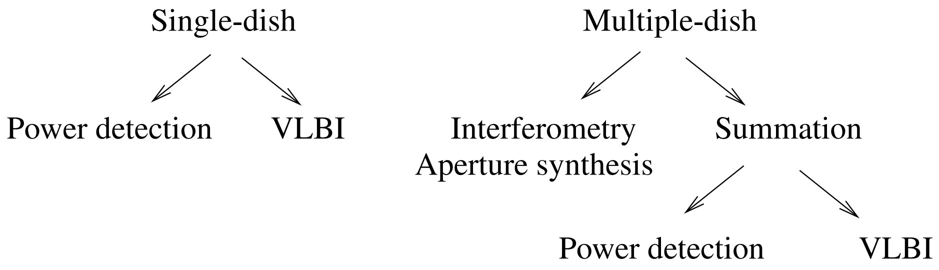


Figure 1.15: Modes of operation for Radio telescopes

ple, the telescopes in Arecibo, Puerto Rico ([31]), Effelsberg, Germany ([32]), Parkes, Australia ([33]) and Green Bank, USA ([34]). Images are constructed by measuring the power received from different directions (power detection).

The most prominent multiple-dish telescopes are the Westerbork Synthesis Radio Telescope (WSRT) in the Netherlands ([35]), the Very Large Array in the USA ([36] and [37]), the Australia Telescope Compact Array ([38]) and the Giant Metrewave Radio Telescope in India ([39]). The techniques used to construct images using the signals from the different dishes in a multiple dish receiver are: interferometry and aperture synthesis (see [40]).

A special form of interferometry is Very Large Baseline Interferometry (VLBI). In VLBI, radio telescopes which are geographically far apart, form a single telescope. The European VLBI Network (EVN, see [41]), for example, consists of the WSRT, operated as a single-dish telescope and 17 other telescopes amongst which the telescope in Effelsberg. Other similar networks are MERLIN, the Multi Element Radio Linked Interferometer Network, an array of radio telescopes distributed around Great Britain (see [42]) and the VLBA, the Very Long Baseline Array, a series of ten radio telescopes spread across the United States (see [43]).

Apart from the construction of astronomical images, radio telescopes are used for other types of astronomical observations. Two examples are the determination of the timing of bursts of electromagnetic energy received from pulsars (see [44]) and the determination of the timing of γ-ray bursts. Single-dish telescopes record the power received by the Radio telescope as a function of time. For multiple-dish telescopes, the spatial information, obtained through the individual dishes, is not relevant in this case and in general, the signals from the different dishes are summed to increase the SNR. The device which is summing the signals from different telescopes is called 'Tied Array Adder' (TAA). Summing the signals from different dishes is also common practice when a multiple-element telescope is part of a larger VLBI network. The different modes of operation for single-dish and multiple-dish telescopes are summarized in figure 1.15.

In this thesis, we will concentrate on the design of a TAA for the WSRT. The WSRT is an array of 14 single-dish antennas located in Westerbork, the Netherlands. The principle of operation of the WSRT can be found in [45], the general lay-out and mechanical aspects in [46]. A schematic overview is given in figure 1.16.

In the top-part, we see the physical layout, in the bottom-part, the corresponding functional layout. Every antenna contains a frontend. A frontend consists of a Low-Noise-Amplifier (LNA) and an RF to IF conversion stage including the necessary filters. The IF signals are transported to a central building where a backend further processes the signals (see [47]). The signals are first converted from IF to Baseband. The analog baseband signals are sampled and coarsely quantized with one or two bits resolution. The digital signals undergo different programmed delays to compensate the pathlength differences for the different antennas (see section 7.3.2). The digitized and delay-compensated signals are further processed by a real-time correlator. The results produced by the real-time correlator are stored and can be used for offline aperture synthesis by General Purpose Computers.

To support the European VLBI Network (EVN) and to enable, for example, pulsar observations, the WSRT can be used as a phased array where the signals from the different telescopes are summed by a TAA. In this second case study, we will use the Y-chart approach to analyze two different options for the realization of a TAA. In the first iteration, we will explore a completely analog TAA. In the second iteration, we analyze a digital TAA using the coarse quantization stages of an existing Back-end. In the design of the TAA using coarse quantization, the Y-chart approach is necessary to explore the different options in a structured way. There are no standard rules for calculating the effects on the sensitivity of the TAA. Specifications concerning sensitivity can only be determined after performance numbers are available. So, we explicitely use the feedback option, given by the Y-chart approach, from ‘Performance Numbers’ back to the ‘Functional Specification’ (see figure 1.11). This second case study is more or less an exercise in retrospect. The design of the digital TAA, as described in the second iteration, started already in 1995. Suggesting a completely analog solution including analog pathlength compensation would have been considered a ridiculous act at that time. Furthermore, a basic analysis of coarse quantization in TAA configuration was already available (see [48]).

However, ‘redoing’ the design of the TAA using the Y-chart approach, suggesting an analog solution and introducing coarse quantization is a useful exercise. The analysis of the effects of coarse quantization becomes an explicite stage in the design process. Since the analysis is not straightforward, it is relatively time consuming and one can anticipate on this in planning the design of new systems using coarse quantization. Furthermore, the Y-chart approach forces the designer to document the advantages and disadvantages of the analog and digital solutions which helps in understanding the background of the system specifications.

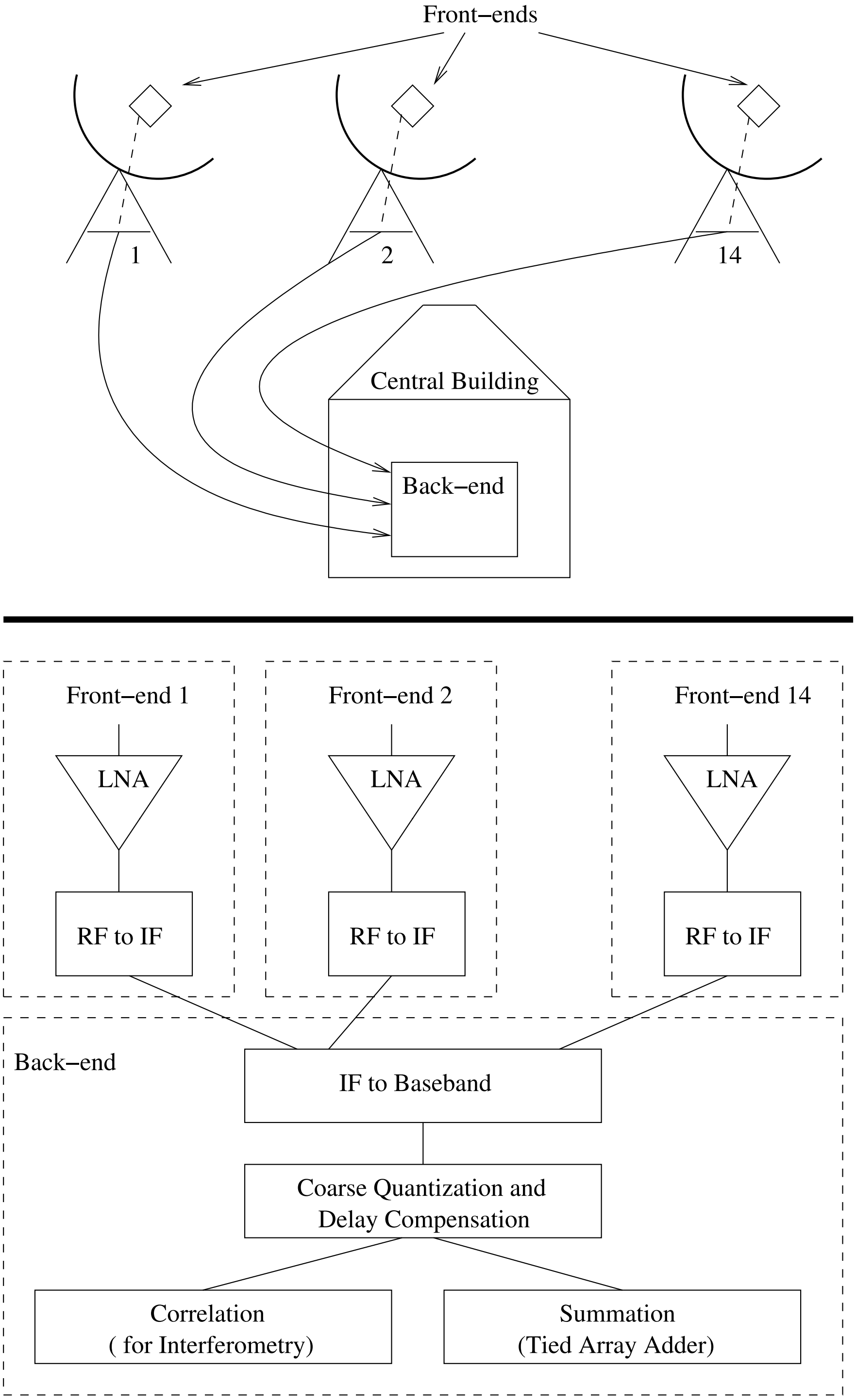


Figure 1.16: Schematic overview of the WSRT

1.7 Organization of the thesis

In this chapter, we have analyzed the evolution of the performance of digital and analog systems and subsystems in section 1.1. We have seen that the performance of digital systems is increasing more rapidly than the performance of analog systems. As a consequence, more and more functionality within mixed-signal systems is shifted from the analog domain to the digital domain. Another way to benefit from the increasing digital performance is digitally assisted analog (section 1.2). We also identified a trend that users require more and more data.

In all cases mentioned above, the performance mismatch between the analog part and the digital part of a mixed-signal system results in high performance requirements for data converters, especially for ADCs. In section 1.3 we have seen that in systems where signal characteristics have to be determined rather than the instantaneous values of a signal, coarse quantization is a way to reduce complexity of ADCs. However, the analysis of coarse quantization is not straightforward and there are no general design rules to determine its effects. For that reason we need an Analog-Digital Codesign methodology that facilitates the introduction and analysis of coarse quantization. Because the effects of coarse quantization are not known exactly during the functional specification phase, the design methodology should provide explicite feedback from the analysis results (performance numbers) to the functional specification.

In section 1.4 we have identified two basic design methodologies: the sequential design methodology (SERV paradigm) and the iterative methodology (Y-chart approach). We also developed a uniform representation in which both methodologies can be identified. From this uniform representation, the differences can be seen clearly. The sequential design methodology describes the whole system level design process while the iterative methodology concentrates on the system level design exploration. Furthermore, the sequential design methodology does not provide explicit feedback to the functional specification while the iterative methodology does. Concluding, we need to use the iterative design methodology (Y-chart approach) to explore the usefulness of coarse quantization in mixed-signal systems.

The applicability of coarse quantization will be illustrated by means of two case studies. The first case study deals with digital predistortion of power amplifiers and is introduced in section 1.6.1. The second case study concerns the design of a digital TAA and is introduced in section 1.6.2.

In chapters 2, 3, 4, 5 and 6, the first case-study is described. In chapters 7, 8 and 9, the second case-study is elaborated. In chapter 10, some conclusions and suggestions for future work will be given.

Chapter 2

Digital predistortion of Power Amplifiers

This chapter is an introduction to the Analog-Digital Codesign of a digital predistorter for Power Amplifiers (PAs) using the Y-chart approach. The first steps are: the ’Functional Specification’, the ’Architecture Specification’ and the ’Mapping’ (see figure 1.11). To support the subsequent steps, the ’Performance Analysis’ of the analog part, the digital part and the overall system, a simulator was developed. This simulator is described in chapter 3. The first iteration of the Analog-Digital Codesign of a digital predistorter for PAs was based on an existing predistortion scheme which will be called the LS predistorter. The LS predistorter is named after the Least Squares criterion that is used to determine the non-linear behavior of the PA. The design of the LS predistorter is described in chapter 4. Based on the evaluation of the LS predistorter, an alternative is developed: the Crosscorrelation predistorter. The design of the Crosscorrelation predistorter is described in chapter 5. In chapter 6, we validate the results, obtained through simulation, with measured data provided by Philips Semiconductors.

In this chapter, we will introduce the general concept of digital predistortion of Power Amplifiers.

2.1 Introduction to digital predistortion

One of the characteristics of modern, spectrally efficient transmission formats like Wideband Code Division Multiple Access (WCDMA) and Orthogonal Frequency Division Multiplexing (OFDM) is that, for the downlink application, within one transmitter the signals from multiple users are multiplexed onto a single channel. Downlink indicates the transfer of information from basestation to terminal. The resulting signals show large peak signal excursions compared to the average level.

Peak-to-average ratios of 12 dB (a factor of approximately 16 in power, a factor 4 in amplitude) occur for both WCDMA (see [49]) and OFDM (see [50]). In the uplink (from terminal to basestation), signals from a single user, on average, show smaller peak-to-average ratios.

For both the down- and uplink, there are stringent requirements concerning the power to be transmitted at different frequencies. In general, power should be concentrated in a frequency slot or channel, while the power outside this channel should be limited. Power outside the assigned channel interferes with the signals transmitted by other users using these adjacent channels and is called Adjacent Channel Interference (ACI). Spectral masks have been defined by regulatory bodies to specify allowed ACI levels. Some spectral masks are presented in section 4.2.1. Regulatory bodies that are involved in the definition of the different masks are the European Telecommunications Standards Institute (ETSI, [51]), the Federal Communications Commission (FCC, [52]) in the U.S and the International Telecommunication Union (ITU, [29]). Within the ETSI and the FCC, telecommunication standards and corresponding spectral masks for Europe and the U.S. respectively, are elaborated and defined. The ITU coordinates the global use of the electromagnetic spectrum.

The design of a transmitter should guarantee that the power-requirements within and outside the assigned frequency slot, are satisfied.

In section 1.6.1, it is indicated that the last part of a telecommunication transmitter is a Power Amplifier (PA). In traditional systems, the signal at the input of the PA satisfies the applicable spectral mask. The PA amplifies the signal in such way that the output signal still satisfies the requirements concerning the ACI levels. To achieve this, the PA should be linear. If a PA with non-linear behavior is fed with a signal with high peak-to-average ratio, severe distortion is introduced. Distortion within the band of transmission degrades signal detection at the receiver because it is part of the signal received at the other end of a communication link. This distortion cannot simply be removed by spectral filtering. For that reason, it can be regarded as noise, decreasing the SNR at the receiver. This will eventually lead to higher Bit Error Rates. Distortion outside the band of transmission (ACI) should be limited and therefore, the allowed non-linearity of the PA should be limited.

In general, PAs within telecommunication transmitters are costly because of the stringent requirements concerning non-linearity. Several techniques have been investigated to relax the linearity requirements of PAs while still satisfying the requirements concerning distortion. These techniques can roughly be categorized into: feedforward, feedback, LINC/CALLUM and predistortion. We will briefly describe these different techniques below.

A block diagram of a feedforward amplifier is given in figure 2.1.

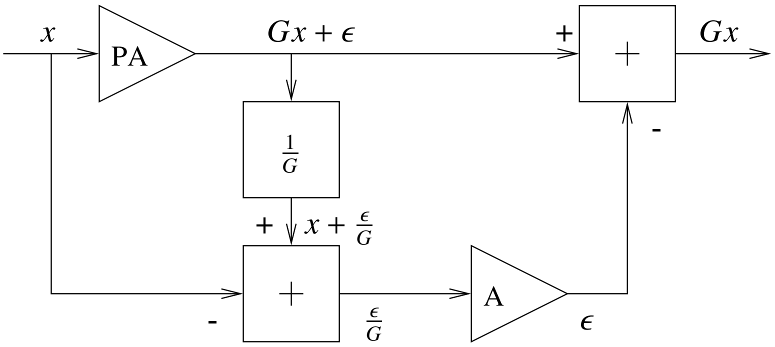


Figure 2.1: Block diagram of a feedforward amplifier

The PA is fed with the input signal x. The linear gain of the amplifier equals G and distortion ϵ is produced. A fraction of the PA output signal is tapped off and attenuated with a factor G. From the resulting signal $x + \frac{\epsilon}{G}$, the original signal x is subtracted after which the signal $\frac{\epsilon}{G}$ is amplified with a factor G by amplifier A. Note that the power of $\frac{\epsilon}{G}$ is much less then the power of x. Amplification of $\frac{\epsilon}{G}$ with a factor G is therefore much easier to realize than the amplification of the original signal x with a factor G. The last stage subtracts the error signal from the output of the PA. This solution operates in the analog domain at Radio Frequencies (RF) and suffers from phase-, delay- and amplitude-imbalances. These imbalances are temperature dependent which further complicates the design of feedforward amplifiers. Furthermore, the power combiner is a critical element in the design of feedforward amplifiers. More information concerning feedforward amplifiers can be found in [30].

An alternative to feedforward is feedback. When using feedback, the error signal is subtracted from the input and not from the output of the PA (as in feedforward PAs). This is illustrated in figure 2.2.

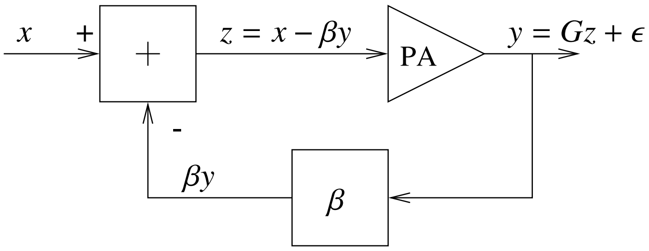


Figure 2.2: Block diagram of a feedback amplifier

A fraction (β) of the output of the PA is tapped off and subtracted from the original signal x. The resulting signal z is fed to the PA. In case no feedback is used ($\beta = 0$), the following holds:

$$y = Gx + \epsilon \tag{2.1}$$

In case feedback is used ($\beta > 0$):

$$y = G(x - \beta y) + \epsilon \tag{2.2}$$

which leads to:

$$y = \left(\frac{G}{1 + G\beta}\right)x + \left(\frac{1}{1 + G\beta}\right)\epsilon \tag{2.3}$$

It is clear that the distortion is reduced with a factor $(1 + G\beta)$. The gain of the original signal is reduced with the same factor. If the loopgain $G\beta$ is large, the gain of the original signal becomes $\frac{1}{\beta}$. The gain of the PA must be high to obtain a high resulting gain $\frac{1}{\beta}$ and a low distortion. The problem with feedback PAs, used in wideband applications, is to have sufficient loopgain $G\beta$ over the complete bandwidth. For high frequencies, the loopgain becomes too small.

Two other techniques are LINC and CALLUM. LINC stands for Linear amplification with Non-linear Components. LINC is illustrated in figure 2.3.

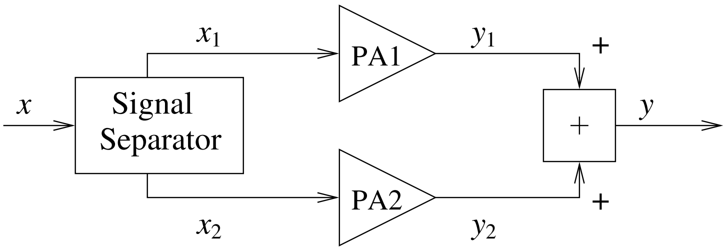


Figure 2.3: Block diagram of LINC

The general idea is to split the original signal x into two components x_1 and x_2. Both components are amplified separately by two PAs and after amplification, the resulting signals y_1 and y_2 are combined into the output signal y. There are different options when using the LINC technique. One option is to use a simple power splitter as a signal separator ($x_1 = x_2$) and use two matched PAs. These two PAs are selected in such way that the distortion generated by PA1 cancels the distortion generated by PA2. A second option is to split the original signal x, which has a fluctuating envelope, into two signals x_1 and x_2 which have a constant envelope. This way, the two individual PAs can be used more efficiently. After amplification, the signals are combined. More information concerning the LINC technique can be found in [53].

The LINC technique is further elaborated into the CALLUM technique (Combined Analogue Locked Loop Universal Modulator). The basic configuration of a CALLUM PA is given in figure 2.4.

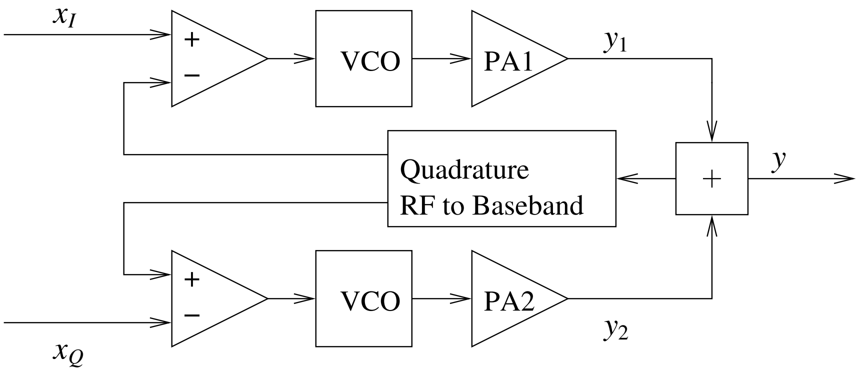


Figure 2.4: Block diagram of CALLUM

In contrast with the previous techniques, CALLUM requires a complex input signal. At the input, the in-phase signal (x_I) and the quadrature-phase (x_Q) components of the complex input signal are compared with two feedback signals. These two feedback signals are the in-phase and quadrature-phase baseband representations of the output signal. The difference between the in-phase components is used to control a constant-amplitude Voltage Controlled Oscilator (VCO). By changing the voltage at the input of the VCO, the frequency of the output signal is changed. This signal is exciting one of the PAs.

The difference between the quadrature-phase components is used to control a second VCO. This signal is amplified by a second PA. The sum of both PA outputs is the final output of the CALLUM amplifier. By means of the two control loops, the output signal is the sum of two constant-amplitude signals with different frequency characteristics. For more information concerning the CALLUM technique, we refer to [54].

The disadvantages of the LINC and CALLUM techniques are that they are sensitive to phase and amplitude imbalances and use two PAs together with a power-combining network instead of a single PA.

A general characteristic of the solutions mentioned above is that analog components are added to compensate the non-linear behavior of the PA. Both the ’compensation circuits’ and the PA are sensitive to temperature changes and aging. This can lead to instabilities. A solution which is less sensitive to temperature changes and aging is digital predistortion. The general principle of predistortion is to apply the inverse of the non-linear part of the input-output relation of the PA to the signal at the input of the PA. Predistortion followed by the PA (and its inherent distortion) should result in linear amplification. Predistortion can be applied in the analog domain (see [55]) or in the digital domain (see [56]). In digital predistortion, the digital baseband signal x is predistorted before it is converted to the analog domain, frequency translated to Radio Frequencies (RF) and amplified. (see figure 2.5).

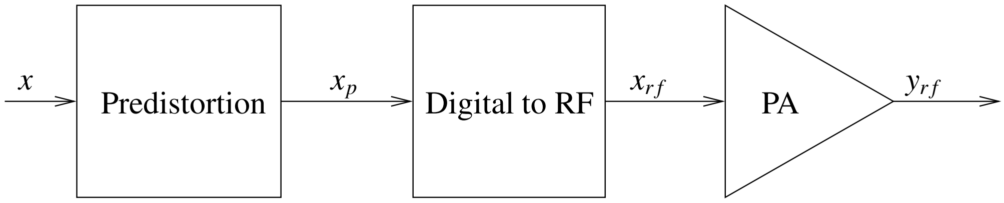


Figure 2.5: Block diagram of Digital Predistortion

Because the input-output relation of the PA changes in time due to temperature changes and aging of components in the analog part, the predistortion has to be controlled. A control mechanism constantly adapts the Predistortion block via the Predistortion Control block, see figure 2.6.

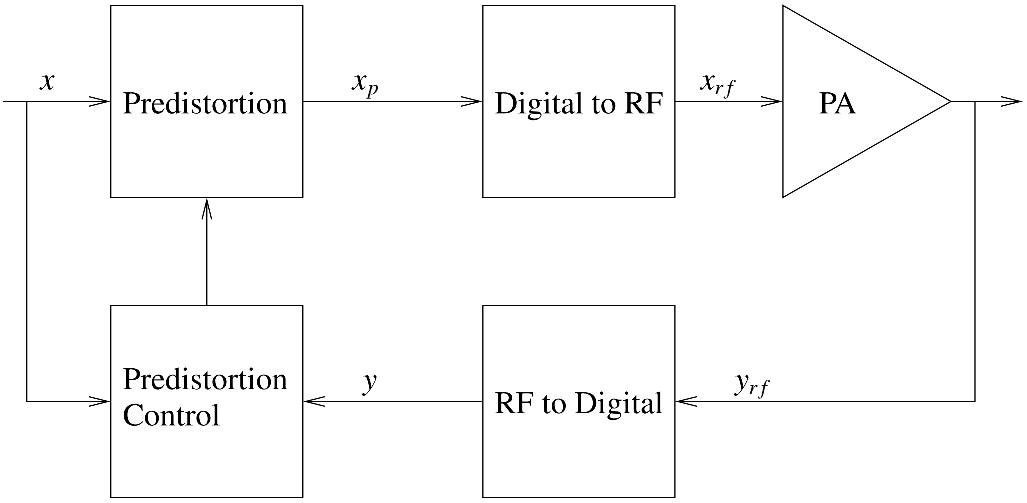


Figure 2.6: Block diagram of Digital Predistortion with feedback (Alternative 1)

A small fraction of the PA output y_{rf} is fed back. By means of a bandpass filter, the frequency components around the carrier frequency are selected and converted from RF to digital baseband, resulting in the signal y. The selection of the relevant frequency components and the effects of the conversion from RF to digital baseband will be dealt with in more detail in section 4.2.2.2.

There are two methods to control the predistortion. In the first alternative an adaptation algorithm within the Predistortion Control block compares the signal y with the input x, as indicated in figure 2.6. This solution is called Alternative 1. In the second alternative, an adaptation algorithm compares y with the output x_p of the Predistortion block, see figure 2.7.

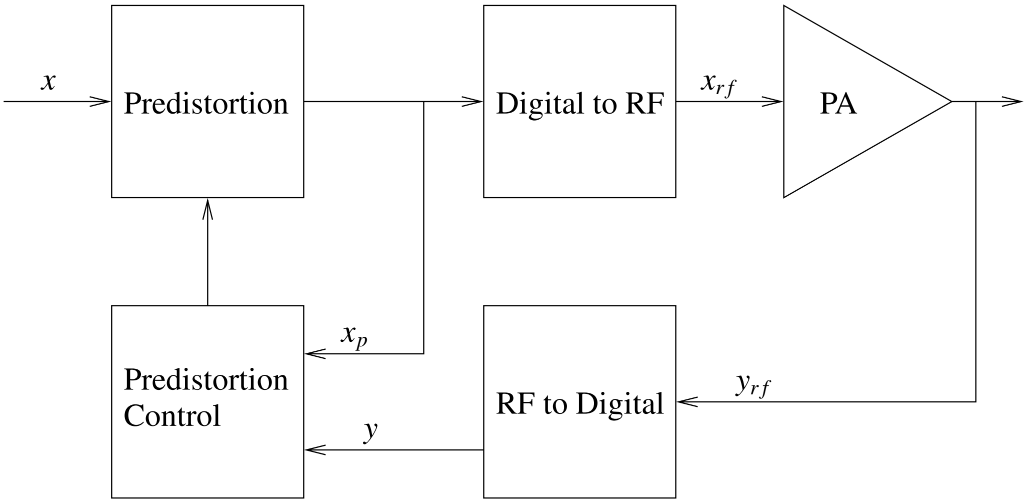


Figure 2.7: Block diagram of Digital Predistortion with feedback (Alternative 2)

This solution is called Alternative 2.

During the design of Digital Predistortion systems, stability, convergence and synchronization are important issues. Schemes which fit Alternative 1 can be found in [57], [58], [59], and [60]. In these schemes, only the combined effect of the predistortion and the PA is measured. Especially the settings of the Predistortion block at the start-up of the system are important. If these settings are not close to the ideal settings, the Predistortion Control block might not be able to find the correct values leading to unstable behavior. Predistorters according to Alternative 2 are more stable because the effects of solely the PA are measured, independent of the predistortion. For that reason, we will concentrate on Alternative 2.

When using the signals x_p and y to determine the behavior of the PA, two different approaches exist: Direct- and Indirect Learning. When using Direct Learning, the input-to-output relation of the PA at baseband frequencies is determined. The output of the PA, converted into the digital domain, is expressed as a function of the input before conversion to analog RF signals: $y = g(x_p)$. As mentioned, the basic idea behind predistortion is to apply the inverse of this relation to the digital input signal x. For that reason, the relation found (g) has to be inverted. This inverted relation (g^{-1}) is used to predistort the input signal x. For an example, see [61]. In general, finding the inverse relation g^{-1}, while knowing g, is not trivial, requiring complex digital algorithms.

The calculation of the inverse of the relation g is avoided in the Indirect Learning scheme which was introduced in [62]. An example of predistortion linearization based on the Indirect Learning scheme is given in [63]. Instead of expressing the output of the PA as a function of the input ($y = g(x_p)$), the input is expressed as a function of the output: $x_p = f(y)$. In the ideal case, $f() = g^{-1}()$. After determining the relation f, it can directly be used to predistort the signal x.

The PA (in case of Direct Learning) or the predistorter (in case of Indirect Learning) can be modeled using memory polynomials (see [64]). One can use 'normal' polynomials, which are simply the powers of different orders, or orthogonal polynomials (see [65]). This will be explained in more detail in section 4.2.2.

2.2 Outlook

In chapters 3, 4 and 5, we will focus on the Analog-Digital Codesign of a digital predistorter. We will emphasize on the design of the algorithms within the Predistortion Control block in relation to the design of mainly the ADC within the RF to Digital block. We will concentrate on Alternative 2 and analyze Direct- and Indirect Learning schemes using normal- and orthogonal polynomials.

Chapter 3

The simulator

In this chapter, we describe a simulator which was built to support the analysis of different digital predistortion algorithms. First, the structure of the simulator is given. Then the different blocks within the simulator are described.

3.1 Introduction

Before we start the analysis of different options for a digital predistorter for PAs, we first describe the general structure of a simulator.

A very important activity within the Analog-Digital Codesign of a digital predistorter for a PA using the Y-chart approach, is the ’Performance Analysis’ (see figure 1.11). Within the performance analysis, the architecture-application combination is analyzed. The results of the performance analysis are ’Performance Numbers’. Basically, two different types of performance numbers need to be generated. The first type are performance numbers which are related to the implementation of a predistortion scheme. These numbers indicate the complexity of the algorithm and they can be used to estimate the execution time of the algorithm on a specific processor. These implementation-specific performance numbers will be based on the size of the datastructures that are used and on the actions that are executed by the predistortion control algorithm. From sections 4.4.4 and 5.3.4 and from appendix C we can see that we do not need specific tools for generating these performance numbers. The second type of performance numbers are numbers which indicate the performance of the functional behavior, more precisely, the suppression of the power in the adjacent channels that is achieved. These performance numbers (suppression of the power in the adjacent channels) are not obtained that easily. There basically are two options:

- Building an executable model of digital predistortion in combination with a model of a PA. Because of the large processing power of desktop computers nowadays, it is feasible to build these models and simulate the behavior of both digital predistortion and non-linear PAs. Furthermore, models of PAs are available and can directly be incorporated into the simulator. The advantage of this option is that a simulator of digital predistortion of PAs can be built relatively fast. The infrastructure that is necessary, basically consists of a desktop computer with simulation support software. One can build a simulator in a programming language like C or one can use a computing environment like Matlab. However, because of simplifications, the PA-models never exactly reflect the behavior of real PAs. It is therefore difficult to predict exactly how a digital predistorter will perform in a final implementation of a (telecommunication) transmitter.

- Designing and realizing a hardware setup to test digital predistortion in combination with real PAs. By really designing and implementing the predistorter, including a PA, one can asses the performance in real-life situations. The step from a hardware setup to a product, using digital predistortion, is smaller than the step from a simulation model to a product. However, the design and realization of a hardware setup will be time-consuming and more expensive than building a simulator. More importantly, however, when building a hardware setup, many aspects and environmental conditions will influence the performance of the predistorter. For example, the layout of the Printed Circuit Board of the analog (RF) part and the location of cables will influence the behavior of the PA. The environmental temperature will influence the operating temperature which on its turn will influence the non-linear behavior of the PA. For that reason it is necessary to control the environmental conditions as much as possible. Furthermore, experience is necessary on how to build such an experimental setup.

For a thorough analysis of digital predistortion of PAs, acquiring performance numbers through a hardware setup would be preferable. However, at this stage of the research we are basically interested in the difference in performance of different predistortion control algorithms. From this point of view, the influence of environmental conditions is of less concern in this stage. To start with a hardware setup will require too much effort in relation to the goal of the research. The design and implementation of a simulator is more efficient. A prerequisite is of course that realistic models of PAs are available.

The simulator uses baseband-equivalent processing with baseband-equivalent Power Amplifier Models. The principle of baseband-equivalent processing is indicated in figure 3.1.

If we compare figure 3.1 with figure 2.7, we see that the Digital to RF block, the PA and the RF to Digital block are replaced by a Baseband-Equivalent Power

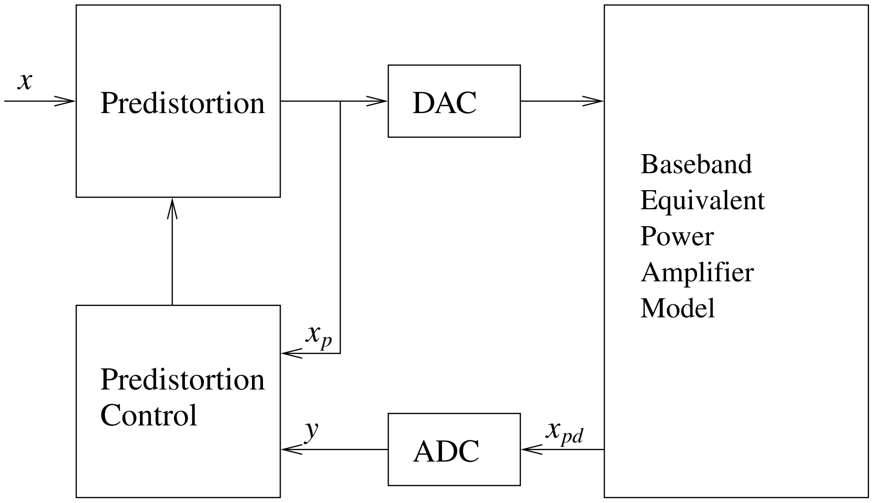


Figure 3.1: Baseband-equivalent model of digital predistortion of PAs.

Amplifier Model (PA model), a DAC and an ADC. In the PA model, the effects of time discrete-to-continuous conversion, time continuous-to-discrete conversion, the frequency conversion stages and the non-linearity of the PA are accounted for. In the DAC, the conversion from amplitude-discrete to amplitude-continuous is modeled. In the ADC, the conversion from amplitude-continuous to amplitude discrete is modeled. The advantage of this approach is that we do not have to simulate two frequency conversion stages. This is common practice in the design of simulators for mixed-signal systems containing an RF part because it reduces simulation times drastically. Furthermore, there are baseband-equivalent models available in literature ([64], [65]), based on measurements on real PAs.

The input of a Baseband-Equivalent Power Amplifier Model is a sequence of complex values, consisting of a real- and an imaginary part. The real- and imaginary part specify a vector in the complex plane. The length of this vector represents the amplitude of the RF carrier at the input of a PA in a transmitter. The angle between the vector and the positive-real axis describes the phase. The output of the Baseband-Equivalent Power Amplifier Model is a complex value describing the amplitude and phase of the RF carrier at the output of a PA. In the remainder of this thesis we will refer to Baseband-Equivalent Power Amplifier Models as PA models.

In the simulations, we neglect the effects of the DAC. We assume it has sufficient resolution and we therefore do not model its behavior in the simulator. However, we are interested in the effects of reduced resolution of the ADC in the feedback path so its behavior is modeled.

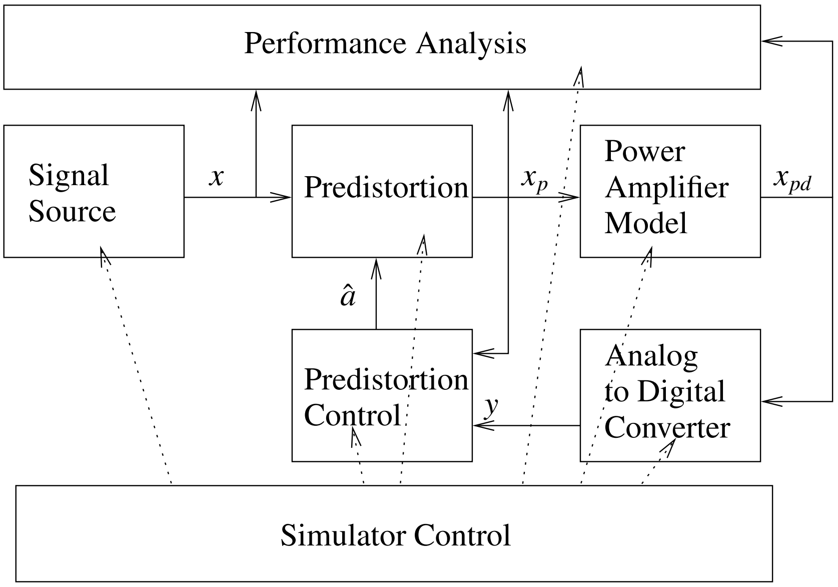


Figure 3.2: Block diagram of the simulator

3.2 Structure of the simulator

A block diagram of the simulator is presented in figure 3.2.

The simulator contains a Signal Source which produces complex samples. The complex values are predistorted by the Predistortion block, using information provided by the Predistortion Control block. The Power Amplifier Model is a Baseband-Equivalent Power Amplifer Model. The Analog-to-Digital Converter (ADC) consists of two quantizers. One quantizer for the real part of the complex output of the Baseband-Equivalent Power Amplifier Model and one quantizer for the imaginary part. Each quantizer converts the floating point numbers, used up to this point, into numbers which can be represented by a limited number of bits. The Predistortion Control block calculates an estimate of the non-linearity of the PA model and presents it to the Predistortion block. The performance of a predistortion scheme is determined by the Performance Analysis block and the simulator is controlled via a Simulator Control block. The different blocks will be explained in more detail below.

3.3 Signal Source

There are several options to model the Signal Source block. One option is to model the digital baseband processing parts of for example a UMTS transmitter, an IEEE802.11b or an IEEE802.11g transmitter. UMTS and IEEE802.11b use

Direct Sequence Spread Spectrum (DSSS) as a technique to share a frequency slot amongst multiple users. For UMTS, the access technique is also referred to as Wideband Code Division Multiple Access (WCDMA). IEEE802.11g uses Orthogonal Frequency Division Multiple Access (OFDM) for the same purpose. The two types of access-techniques (WCDMA and OFDM) generate signals with different characteristics. We chose, however, to design a single signal source which is not directly modeled according to any of the two access-techniques. The reason for this is that first, in case of a standard-specific signal source, the simulations become slow and second, for showing the effects of digital predistortion at baseband, it is not necessary to generate signals according to a specific access-technique. This is explained below.

To model a signal source according to an access-technique requires, in general, at least a random complex number generator and a modulator for each user and a filter for the combined signal of several users. For both WCDMA systems and OFDM systems, it is known (see [67] and [50]) that the addition of several users leads to high 'crest factors'. A crest factor is defined as the ratio of the peak power to the average power. These high crest factors are the main driver for the research on linearization of PAs. So, any signal source, modeled according to a specific access-technique should incorporate many users. This makes the simulations using an access-technique specific signal source, very time consuming. An alternative is to generate a dataset once, record it and use it for all simulations. Repeatedly using a small dataset has the main disadvantage that the same data is used over and over, making all simulations dependent on this specific dataset. For that reason, we chose to develop a signal source that generates a different signal for every simulation run. The generated signal should fulfill the following requirements to be comparable with a WCDMA or an OFDM signal:

1. It should have a crest factor comparable to the crest factors of WCDMA and OFDM systems.

2. The power of the signal should be concentrated within a limited frequency band called the primary channel. The power outside the primary channel should be negligible. The requirement that the power in the adjacent channels should be negligible is also important for the determination of the performance of the different predistortion systems. The power in the adjacent channels is used as a measure of the performance. The original signal, before the PA, should therefore not contain significant power in the adjacent channels.

3. The power spectral density should be flat within the primary channel, similar to the spectra within WCDMA- and OFDM systems.

The signal source we used is based on a signal source presented in [58]. The basic idea is to sum sinusoidal signals of different frequencies and equal power. In total N_{freq} sinusoidal signals are summed. The phases of these sinusoidal signals are chosen randomly. The frequencies of the signal are limited to the frequencies within the primary channels and because the sinusoidal signals have equal power, the power spectral density within the primary channel is flat. The formal description of the signal source, generating the complex signal $x[t]$, is:

$$x[t] = \frac{\sigma_x}{\sqrt{N_{freq}}} \sum_{f=1}^{N_{freq}} \cos(2\pi f t + \phi_r[f]) + i\ \cos(2\pi f t + \phi_i[f]) \tag{3.1}$$

$$N_{freq} = \frac{T}{2 \cdot M} \tag{3.2}$$

The amplitude of the in-phase and quadrature phase components is determined by σ_x. ϕ_r and ϕ_i are two arrays of length N_{freq}, where every element is independently chosen from a uniform distribution between $[-\pi, \pi\rangle$. T equals the total amount of samples in the data set to be generated, M is the oversampling factor and $t \in [0, \frac{1}{T}, \frac{2}{T}, ..., \frac{T-1}{T}]$. The oversampling factor M is defined as the ratio of the total bandwidth to be considered and the bandwidth of the generated signal. The generation of a signal is best explained by an example.

Example

Suppose we want to have 64 samples ($T = 64$) and we want to use an oversampling factor 8 ($M = 8$). From expression 3.2, it follows that we have to generate 4 sinusoidal signals ($N_{freq} = 4$). These signals have frequencies ranging from 1 to 4 Hz. The signal is sampled with intervals of $\frac{1}{T}$ seconds. Thus the sampling frequency equals 64 Hz. This is the minimum frequency required to fulfill Nyquist's sampling criterion using an oversampling factor equal to eight ($M = 8$). Note that the actual bandwidth of the generated signal or the actual bandwidth to be considered, is not mentioned in any of the definitions. Because we only use sampled signals, the actual bandwidth is not important. Only the oversampling factor M and the number of samples T are important. Further note that the dataset always contains only one cycle of the lowest frequency preventing cyclic behavior within the dataset. In figure 3.3, an example of four individual signals and the resulting sumsignal are presented. This sumsignal represents in this example, the real part of a complex signal $x[t]$ for $\sigma_x = 1$.

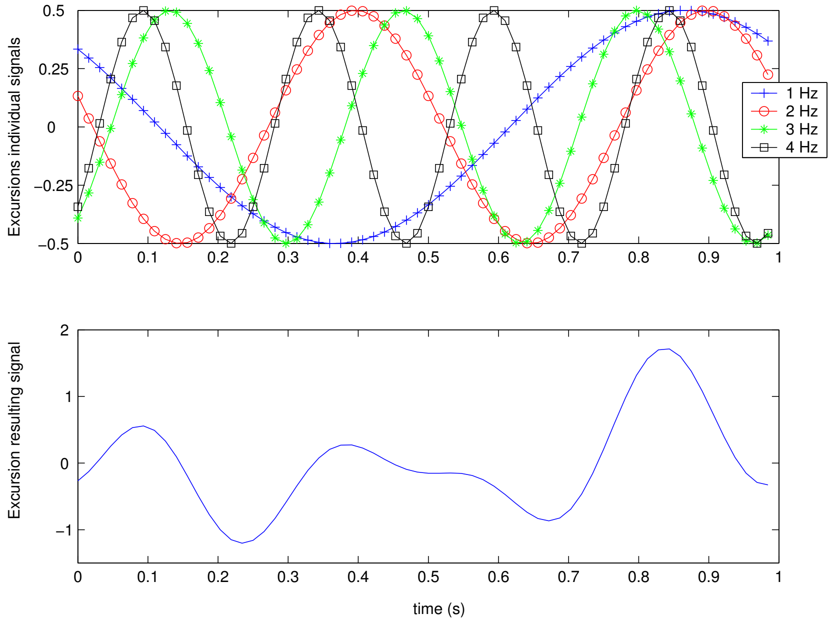


Figure 3.3: An example of a signal and its individual components

In our simulations, we use complex signals with an oversampling factor of 9.8. We started our research using datasets of 8k samples. From initial simulations, this dataset size appeared to produce stable and reproducable results. When using coarse quantization, we also used larger datasets. For different sizes (different Ts), different numbers of sinusoidal signals with different frequencies have to be generated (see expression 3.2). For a block of 8k samples, 416 sinusoidal signals are summed. A typical Power Spectral Density of such a signal is given in figure 3.4.

In this figure, we see the Normalized Power Spectral Density of the generated signal as a function of the 'normalized frequency'. Normalized frequency is defined as the ratio of the frequency of interest and the sampling frequency. In figure 3.4, we also see the Primary- and Adjacent channels. The bandwidth of the primary channel and the adjacent channels is larger than the bandwidth of the generated signal (plus 20%). This reflects the fact that, in general for the different telecommunication standards, the occupied bandwidth is less than the spacing between two carriers. The occupied bandwidth is a measure of the bandwidth containing 99 % of the total integrated power of the transmitted spectrum, centered around the carrier. For UMTS, the occupied bandwidth will be less than 5 MHz based on a chiprate of 3.84 Mcps. For IEEE802.11b and IEEE802.11g, the spacing between the outermost sub carriers equals 16.25 MHz. For two non-overlapping channels in the overlapping European channel selection mode, the carrier spacing equals 20

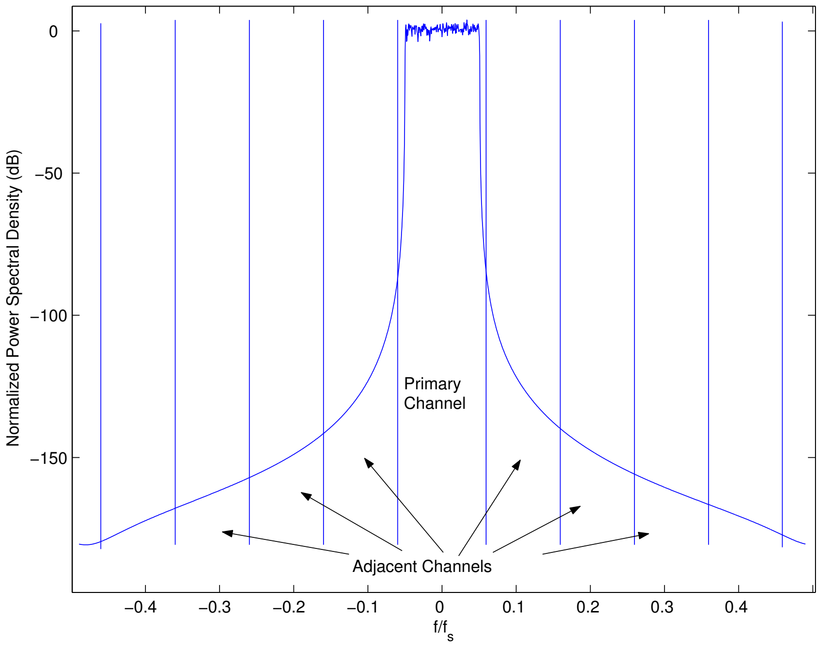


Figure 3.4: An example of the Power Spectral Density of a generated signal

MHz. Based on these figures, the use of a channel bandwidth that is 20 % larger than the occupied bandwidth is representative for the standards mentioned.

From simulations it appears that, in case the power in the primary channel and the power in the two nearest adjacent channels satisfy the different masks (see section 4.2.1), the power in the channels further than the nearest adjacent channels is always lower than the most demanding mask requires. For that reason we will refer to the two nearest adjacent channels as ’adjacent channels’.

An important characteristic of the generated signal is the crest factor. It is common practice to use the logarithmic of the ratio of the peak signal power to the average signal power as a measure of the crest factor. So for the set of samples $x_1, x_2, ..., x_T$, the crest factor is defined as:

$$CF = 10 \cdot \log\left(\frac{\max(|x_t|^2)}{\frac{1}{T}\sum_{t=1}^{T}|x_t|^2}\right) \tag{3.3}$$

To give an impression of typical crest factors, the 100 crest factors for a set of 100 generated signals are given in figure 3.5, where each signal consists of 8k samples. We see that the crest factors range from less than 8 dB to more than 10 dB. To compare our signal source with WCDMA and OFDM signal sources, we determined the statistics of the crest factor. We analyzed 500 blocks of 8k samples and

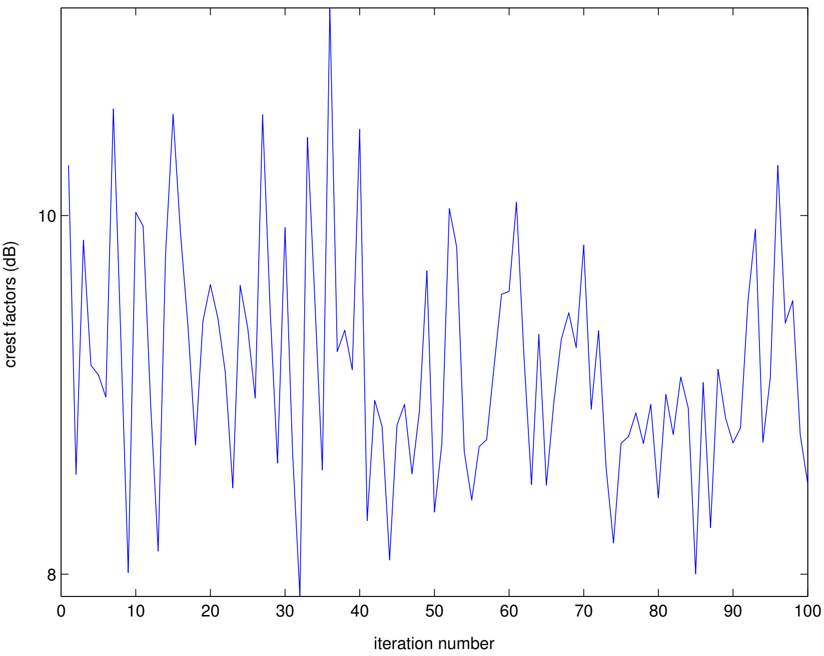


Figure 3.5: Crest factors of a generated datasets

determined the probability that the amplitude of a sample exceeds a given value (see figure 3.6).

From this figure we see that the maximum crest factor for this dataset is between 11 and 12 dB. In [67], the crest factors for single carrier UMTS are given. Depending on the number of code channels transmitted simultaneously on one carrier frequency, the maximum crest factor can range from 4.5 dB (1 code channel) to 11 dB (128 code channels). For multicarrier QAM-modulated WCDMA, some crest factors are reported in [49]. When using four carriers, the probability that the crest factor exceeds 12 dB equals 10^{-5}. From figure 3.6 it can be seen that, for our signal source, the crest factor corresponding to a cumulative probability of 10^{-5} equals approximately 10.5 dB.

To compare our signal source with an OFDM signal, we refer to [50]. The reported maximum crest factor for a cumulative probability of 10^{-5} equals 12.5 dB which exceeds our crest factor of 10.5 dB. In [50], methods are introduced that process user data in such way that crest factors are reduced. These methods are called ’peak power reduction schemes’. The peak power reduction schemes reported in [50] reduce the maximum crest factors below 10.5 dB.

Concluding, our signal source is representative for single-carrier WCDMA and for OFDM in case peak power reduction schemes are used. The crest factor

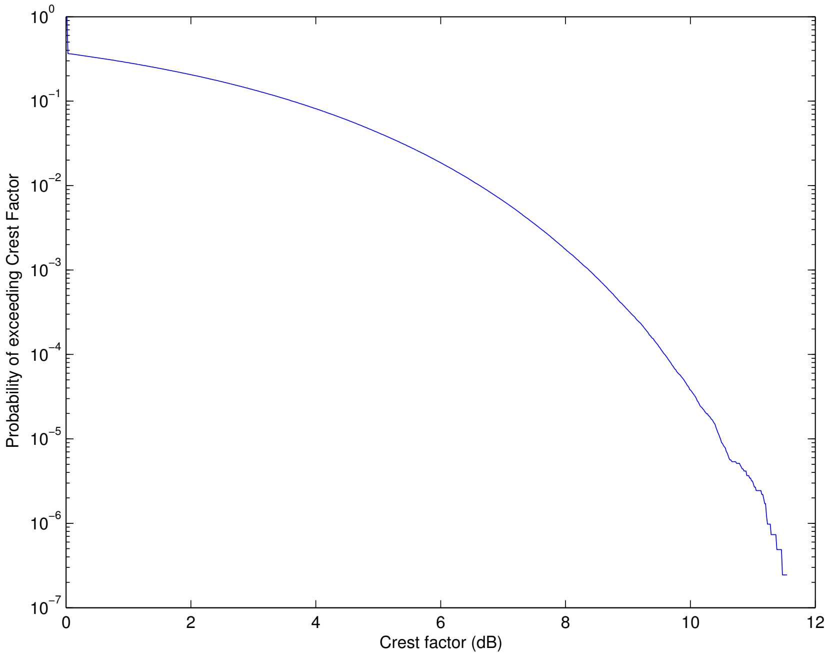


Figure 3.6: Cumulative probability of the factors

of our generated signal is slightly lower than the crest factors for multi-carrier WCDMA and OFDM without peak power reduction.

3.4 Predistortion

The Predistortion block receives an estimate of the non-linearity of the PA model from the Predistortion Control block. Depending on the learning scheme, Direct or Indirect, the estimate represents the input-to-output relation or the output-to-input relation of the PA model respectively. In the Direct Learning scheme, the Predistortion block predistorts the incoming signal with the inverse of the non-linear characteristics of the PA model, found by the Predistortion Control block. In the Indirect Learning scheme, the estimate of the non-linearity is directly used by the Predistortion block.

3.5 The Power Amplifier model

As mentioned in section 3.1, the PA models that are used are Baseband-Equivalent Power Amplifier Models. They account for the effects of continuous-time to/from discrete-time conversion, frequency conversion and the non-linearity of the PA.

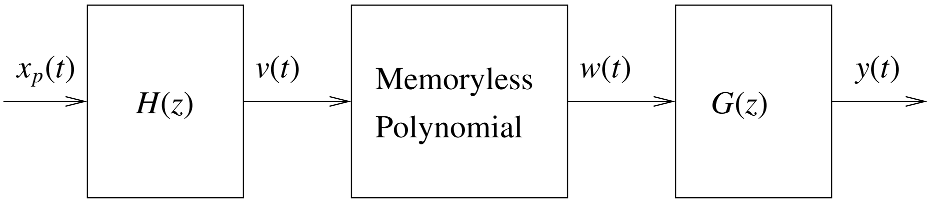


Figure 3.7: Block diagram of the PA model

We have searched through literature and selected the most recent models. Some of these models incorporate memory effects. In the discussion on the performance of different predistortion schemes, we will limit ourselves to a model presented in [64]. It combines severe non-linearity and memory effects. Below, we describe the model. Other models (together with simulation results) are given in Appendix B. In chapter 6, we use data, obtained via measurements on a real PA, provided by Philips Semiconductors, to validate our approach.

The PA model we used to develop and analyze a new Predistortion Control algorithm, is based on measurements on an actual Class AB PA. Class AB amplifiers are a compromise between Class A and Class B amplifiers. Class A amplifiers are briefly non-linear but power inefficient. Class B amplifiers are power efficient but non-linear. Class AB amplifiers combine the techniques used in Class A and Class B amplifiers and their power efficiency and linearity are average. Amplifiers, using switched-mode techniques like Class C and Class D amplifiers, cannot be used to generate signals within the high-RF frequency range. For that reason, Class AB amplifiers are often applied within telecommunication transmitters.

The model of the Class AB PA we used in our simulations, is a Wiener-Hammerstein model which is a combination of a Wiener and a Hammerstein model. A Wiener model consists of a filter followed by a memoryless polynomial (see [69], p. 28). A Hammerstein model consists of a memoryless polynomial followed by a filter. A Wiener-Hammerstein model thus consists of a filter $H(z)$, followed by a memoryless polynomial which on its turn is followed by a second filter $G(z)$ (see figure 3.7). The Wiener-Hammerstein model is alternatively named a Linear-Nonlinear-Linear cascade system ([69]).

For the specific PA model we used and which is presented in [64], the filter specifications are:

$$H(z) = \frac{1 + 0.5z^{-2}}{1 - 0.2z^{-1}}, \quad G(z) = \frac{1 - 0.1z^{-2}}{1 - 0.4z^{-1}} \tag{3.4}$$

The memoryless polynomial equals:

$$w(t) = b_1\, v(t) + b_3\, v(t) \mid v(t) \mid^2 + b_5\, v(t) \mid v(t) \mid^4 \tag{3.5}$$

where v indicates the output of filter H and w the input of filter G. The coefficients are: $b_1 = 1.0108 + 0.0858j$, $b_3 = 0.0879 - 0.1583j$, $b_5 = -1.0992 - 0.8891j$. Furthermore, in the PA model, a gain correction is applied in such way that the overall gain (in case of an input signal with a small standard deviation) equals 0 dB. This implies that the linear gain is removed from the amplifier model. This is necessary because the Predistortion Control block will control the Predistortion block in such way that the output of the PA equals the input of the Predistortion block. If we would not correct for the gain of the PA, the Predistortion block would attenuate the signal. Then a more linear part of the input range of the PA would be used leading to lower ACI levels. This, however is not an achievement of the Predistortion Control algorithm but is mainly due to the power reduction within the Predistortion block. To compare different algorithms within the Predistortion Control block, we therefore remove the linear gain from the PA output. The model is valid for a standard deviation at the input of approximately 0.13.

3.6 The Analog-to-Digital Converter

To analyze the effect of quantization in the feedback path of the digital predistorter, a model of an Analog-to-Digital Converter (ADC) is incorporated in the simulator. An ADC consists of two quantizers: one for the real part of the complex feedback signal and one for the imaginary part of the complex signal. Both quantizers are identical and we will therefore first assume that the signal x_{pd} (see figure 3.1) is a real signal. At the end of this section, we will describe the ADC in case x_{pd} is a complex signal.

The sampling rate of the ADC equals the rate of samples within our baseband-equivalent model and therefore the only task of the ADC in our simulator is quantization. The set of possible output values of a quantizer is finite and therefore the output of the quantizer can be coded into a limited number of bits. In the simulations, we use the output of the quantizer directly, not the bit-encoded values; the actual encoding is irrelevant for our purpose.

We will first give a general description of quantizers and use x_{pd} as a general name for the input. The analog output x_{pd} of the PA model is quantized into the digital signal y (see figure 3.1). We will gradually expand the complexity of the quantizer model to relate the power of the signal x_{pd} to the input range of the quantizer. In the simulations we use b-bit mid-riser quantizers. The relation between x_{pd} and $y(= Q(x_{pd}))$ in case of a mid-riser quantizer is illustrated in figure 3.8.

The main characteristic of a mid-riser quantizer is that if the input value changes its sign, the output changes its sign as well. The alternative would be to use the mid-tread quantizer (figure 3.9).

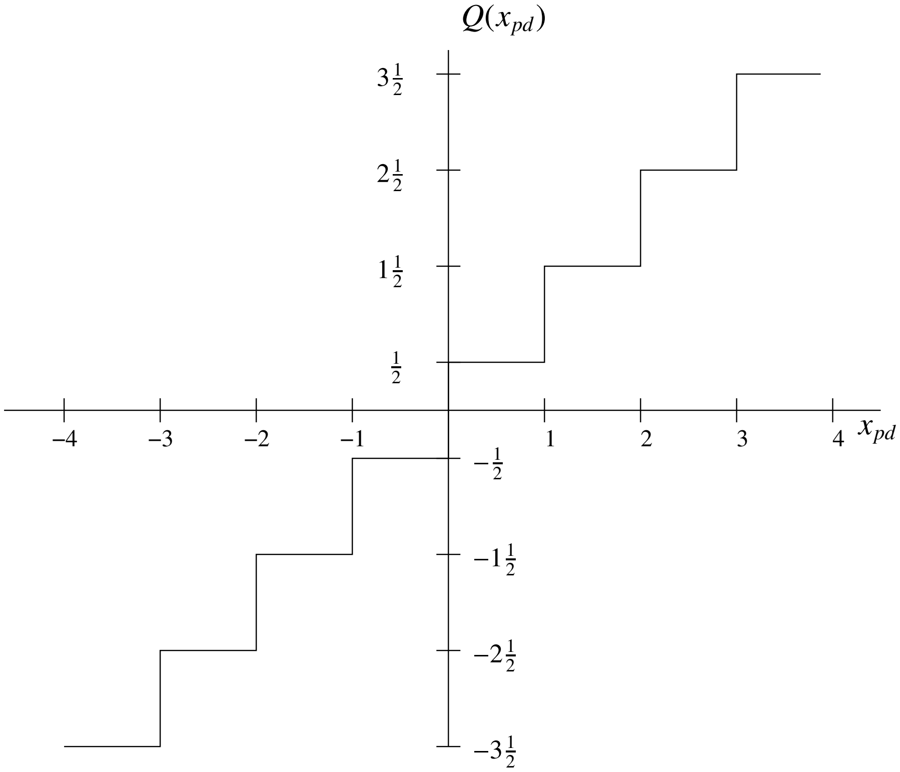


Figure 3.8: Transfer characteristic of a mid-riser quantizer for $b = 3$.

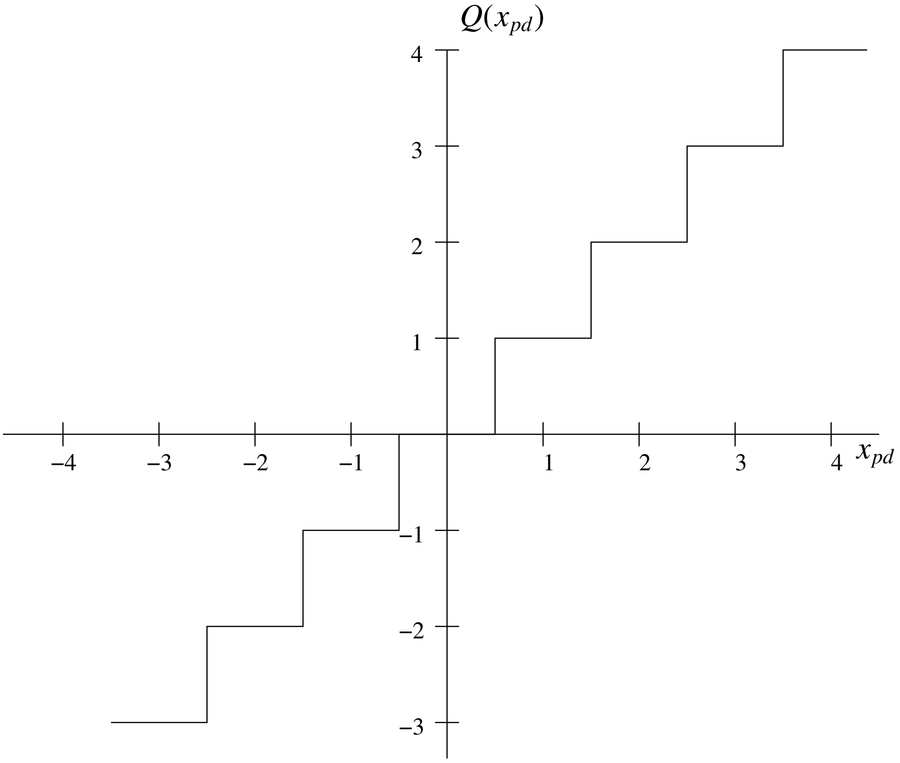


Figure 3.9: Transfer characteristic of a mid-tread quantizer for $b = 3$.

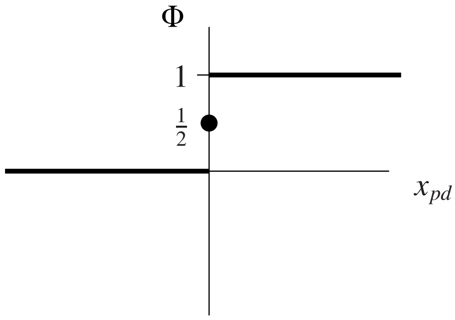


Figure 3.10: Heavisides step function Φ

The major characteristics are that the mid-tread quantizer can produce the value '0' and the number of different digital positive values that can be generated differs from the number of different negative values. One of the consequences is that, if the input is symmetrical around $x_{pd} = 0$, it is not guaranteed that the output will be symmetrical around $Q(x_{pd}) = 0$. We expect this to complicate the analysis and we therefore chose the mid-riser quantizer. For the mid-riser quantizer, the output is symmetrical around $Q(x_{pd}) = 0$ if the input is symmetrical around $x_{pd} = 0$.

As a basic building block of the quantizer, we use Heaviside's step function Φ:

$$\Phi(x_{pd}) = \begin{cases} 0 & x_{pd} < 0 \\ \frac{1}{2} & x_{pd} = 0 \\ 1 & x_{pd} > 0 \end{cases} \tag{3.6}$$

Heavisides step function is illustrated in figure 3.10.
A b-bit mid-riser quantizer is then generally described by:

$$Q(x_{pd}) = \left(\sum_{i=-2^{b-1}+1}^{2^{b-1}-1} \Phi(x_{pd} - i) \right) - 2^{b-1} + \frac{1}{2} \tag{3.7}$$

where x_{pd} is a real (not complex) signal. Before we can use this quantizer, we have to pre-process the input signal to relate the power of the input signal x_{pd} to the input range of the quantizer. The pre-processing is illustrated in figure 3.11 and will be explained below.

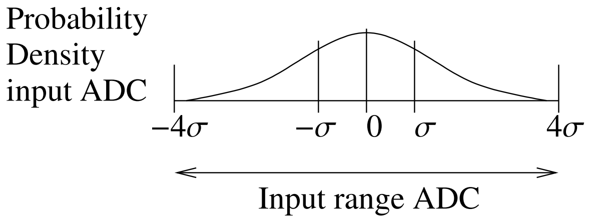


Figure 3.12: ADC input range and the probability density at the ADC input

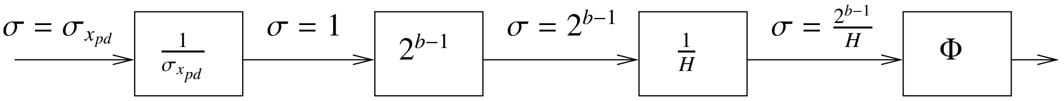

$\sigma = \sigma_{x_{pd}}$
$\frac{1}{\sigma_{x_{pd}}}$
$\sigma = 1$
2^{b-1}
$\sigma = 2^{b-1}$
$\frac{1}{H}$
$\sigma = \frac{2^{b-1}}{H}$
Φ

Figure 3.11: Pre-processing of the input of the quantizer

First we normalize the standard deviation of the signal x_{pd} to one. This is realized by multiplication with $\frac{1}{\sigma_{x_{pd}}}$, where $\sigma_{x_{pd}}$ represents the standard deviation of the feedback signal x_{pd}. The maximum input range of the quantizer, equals $[-2^{b-1}, ..., 2^{b-1}]$. Since we want to use the complete input range of the quantizer, we amplify the input signal with the factor 2^{b-1}. We assume that the characteristics of x_{pd} are similar to the characteristics of a Gaussian noise signal. Since the standard deviation of x_{pd} after multiplication with $\frac{2^{b-1}}{\sigma_{x_{pd}}}$ equals the maximum input range of the quantizer, quantization is not done correctly for a significant fraction of the input signal. For that reason we reduce the power of the signal. If we reduce the power of the signal with, for example, a factor 4, the quantizer can deal with amplitudes up to the $+4\sigma_{x_{pd}}$ and $-4\sigma_{x_{pd}}$ levels. Amplitudes higher than the $+4\sigma_{x_{pd}}$ and lower than the $-4\sigma_{x_{pd}}$ levels will be clipped. The power-reduction factor is called the 'Headroom factor' H. In our simulations, we will also use a Headroom factor of 4 for both the real and imaginary parts of the signal. We have already seen that the probability that crest factors beyond 12 dB (a factor 4 in amplitude) appear, is below 10^{-7} (see figure 3.6), so by choosing $H = 4$, clipping effects are negligible.

The relation between the probability density of the signal at the input of the ADC and the input range of the ADC is illustrated in figure 3.12.

In a digital predistorter, x_{pd} is a complex signal so we have to use separate quantizers for both the real and imaginary part of the signal. The quantization is then described as:

$$Q_{re}(x_{pd}) = \frac{1}{g_{re}} \cdot \left(\left(\sum_{i=-2^{b-1}+1}^{2^{b-1}-1} \Phi(g_{re}\mathrm{Re}(x_{pd}) - i) \right) - 2^{b-1} + \frac{1}{2} \right) \quad (3.8)$$

$$Q_{im}(x_{pd}) = \frac{1}{g_{im}} \cdot \left(\left(\sum_{i=-2^{b-1}+1}^{2^{b-1}-1} \Phi(g_{im}\mathrm{Im}(x_{pd}) - i) \right) - 2^{b-1} + \frac{1}{2} \right) \quad (3.9)$$

where $\mathrm{Re}(x_{pd})$ and $\mathrm{Im}(x_{pd})$ give the real and imaginary part of the signal x_{pd} respectively and g_{re} and g_{im} determine the granularity of the quantization process of the real and imaginary parts (as illustrated in figure 3.11):

$$g_{re} = \frac{2^{b-1}}{H\,\sigma_{\mathrm{Re}(x_{pd})}} \quad (3.10)$$

$$g_{im} = \frac{2^{b-1}}{H\,\sigma_{\mathrm{Im}(x_{pd})}} \quad (3.11)$$

3.7 Predistortion Control

In the Predistortion Control block (see figure 3.2), two different algorithms are implemented. These algorithms estimate the non-linearity and quantify the memory effects of the PA. The overall effects of the non-linearity and the memory effects will be called 'non-linear behavior'. The algorithms are described in section 4.2.2 and in section 5.1.2.

As will be explained in section 3.8, the simulator can run for several cycles. In each cycle (for example cycle i), an estimate $\hat{\mathbf{a}}_i$ of the non-linear behavior of the PA model is obtained. This estimate is used to predistort the signal in the next cycle (cycle $i + 1$). From simulations it appears that the performance of the predistorter fluctuates considerably when the estimate is used directly without taking previous estimates $\hat{\mathbf{a}}_{i-1}$, $\hat{\mathbf{a}}_{i-2}$, etc. into account. The fluctuations are caused by the fact that the number of samples within one cycle is finite, in combination with the randomness of the input signal x. Suppose, for example, that in the 8k samples within cycle i, extreme amplitudes are not present. For these PA input amplitudes, no information concerning the corresponding PA output amplitudes is available. Hence, the distortion for these amplitudes is not known. So, the non-linear behavior for large amplitudes of x cannot be determined in cycle i. When these amplitudes occur in cycle $i+1$, this cannot be corrected, leading to relatively large Adjacent Channel Interference (ACI) levels in cycle $i+1$. For that reason, we do not use the estimated values $\hat{\mathbf{a}}_i$ directly to determine the predistortion in cycle $i + 1$ but we average $\hat{\mathbf{a}}_i$ with the value $\breve{\mathbf{a}}_{i-1}$, previously used by the predistorter.

To average these values, we use a first-order Infinite Impulse Response filter (IIR-filter):

$$\breve{\mathbf{a}}_i = w \cdot \hat{\mathbf{a}}_i + (1 - w) \cdot \breve{\mathbf{a}}_{i-1} \tag{3.12}$$

where w can be programmed and ranges between 0 and 1. We use a first-order IIR-filter because of its simplicity.

Reducing the value of w, gives newly obtained estimates of the non-linearity of the PA less importance. Alternatively, one could say that the memory of the system becomes 'longer' and predistorter settings are based on more information from the past. If the behavior of the PA is stable, it is preferable to use small values of w because then, estimates are based on more data and therefore become more accurate, leading to larger suppression of ACI levels. However, if the behavior of the PA changes in time, the system will respond slowly. If the behavior changes relatively fast, the predistorter settings should be based mostly on recent information hence w should be closer to one. Thus, the value of w is a compromise between accuracy of the predistorter settings and the rate of change of the PA behavior. A value $w = \frac{1}{3}$ is a compromise that gives good suppression of ACI levels and acceptable response times to changes in the behavior of the PA. In sections 4.4.4 and 5.3.4, we will deal with system level timing aspects in more detail.

3.8 Simulator Control

The Simulator Control block synchronizes the execution of the different blocks of the simulator and contains the interface to the user. The user can set several parameters. The most important parameters are: the number of samples in a block (T), the oversampling factor (M) and the learning scheme (Direct or Indirect). Some typical values are: $T = 8\text{k}$ and $M = 9.8$.

Every simulation run consists of cycles. The number of cycles (N_{cycles}) can be specified. Every cycle consists of several steps. For all cycles, except the first cycle, the steps are described below. In this description, first the block which has to execute the step is given (see also figure 3.2) and second, the action. The cycle number is denoted as i:

1. Signal Source: Generate a signal of T complex samples.

2. Predistortion: Predistort the signal of cycle i using the estimated non-linear behavior of the power amplifier, represented by $\breve{\mathbf{a}}_{i-1}$. These coefficients have been generated in cycle $i - 1$. Dependent on the learning scheme, Direct or Indirect, use an inversion algorithm or use the measured non-linear behavior directly.

3. Power Amplifier Model: Distort the signal of cycle i.

4. Analog-to-Digital Converter. Quantize the signal of cycle i.

5. Performance Analysis: Calculate and store the ACI levels.

6. Predistortion Control: Calculate a new estimate $\hat{\mathbf{a}}_i$ of the relation from input to output of the PA model (Direct Learning) or from output to input (Indirect Learning). This estimate is based on the data of cycle i and will be used to calculate the new predistortion values $\breve{\mathbf{a}}$ for cycle $i + 1$. Then proceed with step 1.

For the first cycle, no predistortion is used (the gain of the Predistortion block equals one for every sample). The steps are repeated N_{cycles} times. After the last cycle, the overall performance is calculated.

3.9 Performance Analysis

Within the Performance Analysis block, the achievements of the predistortion mechanism are determined. As a measure of performance, we use the power in the two channels adjacent to the primary channel. For every block of T samples, the power in the adjacent channels and the primary channel at the output of the PA model, is determined. This is done in case predistortion is applied and in case no predistortion is applied. In simulations where several cycles are executed, the average performance is determined. In this case, the lower and upper adjacent channels are treated similarly but separately. The power in the adjacent channels is summed over the cycles and divided by the number of cycles. For the lower adjacent channel, the average power is thus calculated by:

$$P_{low} = \frac{1}{N_{cycles}} \sum_{i=1}^{N_{cycles}} P_{low,cycle(i)} \tag{3.13}$$

where N_{cycles} equals the number of cycles over which the average performance is calculated, $P_{low,cycle(i)}$ equals the average power in the lower adjacent channel of cycle i and P_{low} equals the average power in the lower adjacent channel over N_{cycles} number of cycles. For the upper adjacent channel, a similar value P_{up} is calculated and for the primary channel, P_{prim} is determined. Note that, in general, $P_{up} \neq P_{low}$ because of the complex signal source. The performance of a predistorter is determined by the worst case suppression, thus by the maximum power level in one of the adjacent channels (P_{adj}):

$$P_{adj} = \max(P_{low}, P_{up}) \tag{3.14}$$

This power level is related to the power in the primary channel and expressed in dB:

$$P_{dB} = 10 \cdot \log \left(\frac{P_{adj}}{P_{prim}} \right) \tag{3.15}$$

The value P_{dB} is used as a measure of the performance of a predistorter.

When running simulations over several cycles, the estimated non-linear behavior $\hat{\mathbf{a}}_i$ is generally not directly used in the Predistortion block. As described above, a weighted sum with previous estimates $\hat{\mathbf{a}}_{i-1}$, $\hat{\mathbf{a}}_{i-2}$, etc. is used instead. The effect is that the total system needs a few cycles to stabilize after starting up. Within the Performance Analysis block, a programmable number of cycles, starting with the first cycle, is ignored. This means that the performance in these cycles is not used in the calculation of the average performance. If, for example, the weight w of the estimated coefficients $\hat{\mathbf{a}}$ equals $\frac{1}{3}$, we ignore the first 9 cycles in our simulations.

Besides the performance of a predistorter, the Performance Analysis block also determines the average variance of the signal at the input and at the output of the Predistortion-PA model combination, again with- and without predistortion. These values are used to determine the 'compression' of the signal by the PA model. For high-power input signals, the gain of a PA model is generally less than wanted and compression is a measure of the gain-loss caused by the non-linear behavior of the PA model. The average variance over N_{cycles} cycles is calculated as:

$$P_{in} = \frac{1}{N_{cycles} \cdot T} \sum_{n=1}^{N_{cycles}} \sum_{t=1}^{T} |x(n,t)|^2 \tag{3.16}$$

$$P_{out} = \frac{1}{N_{cycles} \cdot T} \sum_{n=1}^{N_{cycles}} \sum_{t=1}^{T} |x_{pd}(n,t)|^2 \tag{3.17}$$

where $x(n,t)$ is sample t in cycle n of the input x of the predistorter and $x_{pd}(n,t)$ is sample t in cycle n of the output x_{pd} of the PA model. The compression (C_{dB}) is then defined as:

$$C_{dB} = 10 \cdot \log \left(\frac{P_{out}}{P_{in}} \right) \tag{3.18}$$

The small signal linear gain of the amplifier was set to 1 (see section 3.5), so for the linear amplifier, the compression equals 0 dB.

3.10 Outlook

In chapters 4 and 5 we will use the simulator, described above, to analyze different predistorters. The most important aspect is that the different predistorters use the same signal source and the same PA models which enables a fair comparison between different alternatives.

Chapter 4

The LS predistorter

The basic idea behind the Analog-Digital Codesign of a digital predistorter for Power Amplifiers is to reduce the complexity of a telecommunication transmitter, by changing the digital Predistortion Control block and by introducing coarse quantization. The words ’reduce’ and ’change’ in this phrase indicate that we use an existing solution as a reference. As a reference solution, we chose the digital baseband predistorter using memory polynomials as was presented in [64] and which will, in this thesis, be called ’LS predistorter’. ‘LS’ stands for Least Squares and refers to the Least Squares criterion used to estimate the distortion characteristics of a PA. We selected the LS predistorter because it currently is the most recent predistorter dealing with memory effects. To guide the design of a digital predistorter, we use the Y-chart approach (see figure 1.11). The architecture specification, the functional specification and performance analysis of the first iteration within the Y-chart approach are described below.

4.1 Architecture specification

Architecture specification is the first step in the first iteration of the Analog-Digital Codesign of a digital predistorter for PAs in telecommunication transmitters. In this step, we design an architecture on which the application can be mapped. In the architecture, (processing) blocks and the interconnection structure are identified. The aim of the architecture specification phase is to define an architecture in such way that the functionality of the system can be implemented. A general architecture which is expected to fulfill this requirement is shown in figure 4.1.

The general architecture comprises a digital processing part, two data converters, an analog processing part and a PA. The data converters are a DAC in the forward path and an ADC in the feedback path. To enable the analysis of coarse quantisation in the feedback path, we actually do not generate a single architec-

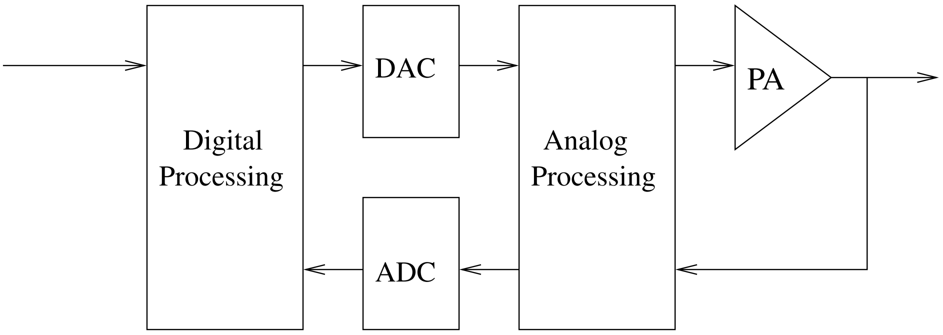


Figure 4.1: Architecture specification of the digital predistorter

ture but we create a set of architectures. Each architecture distinguishes itself by a different resolution of the ADC in the feedback path. These architectures will be used in the performance analysis phase to analyze different architecture-application combinations.

4.2 Functional specification

The second step in the first iteration of the Analog-Digital Codesign of a digital predistorter for PAs in telecommunication transmitters is the functional specification. Functional specification is alternatively named: application design. The result of this phase is a set of requirements concerning the allowed ACI levels of a PA and a specification of the algorithm to control the predistorter.

4.2.1 ACI-level requirements

One of the most important specification of a telecommunication standard for the design of the PA is the transmission mask. As an example, the masks for GSM base stations (figure 4.2), UMTS base stations (figure 4.3), IEEE802.11b and IEEE802.11g (figure 4.4) are given. A transmission mask defines the power a PA is allowed to transmit at different frequencies. In these masks, the frequency of the carrier corresponds to 0 Hz and the x-axis indicates frequency separation from the carrier. This frequency separation is alternatively called 'offset'.
The mask for GSM is derived from the output RF specifications in [10] and the mask for UMTS is derived from the spectrum emission mask given in [11]. Both reflect the most demanding requirements. The masks for IEEE802.11b and IEEE802.11g are taken directly from [12] and [13] respectively. All masks are symmetrical around the carrier. For example, the requirements at -200 kHz offset are equal to the requirements at +200 kHz offset. The signal at the output of a

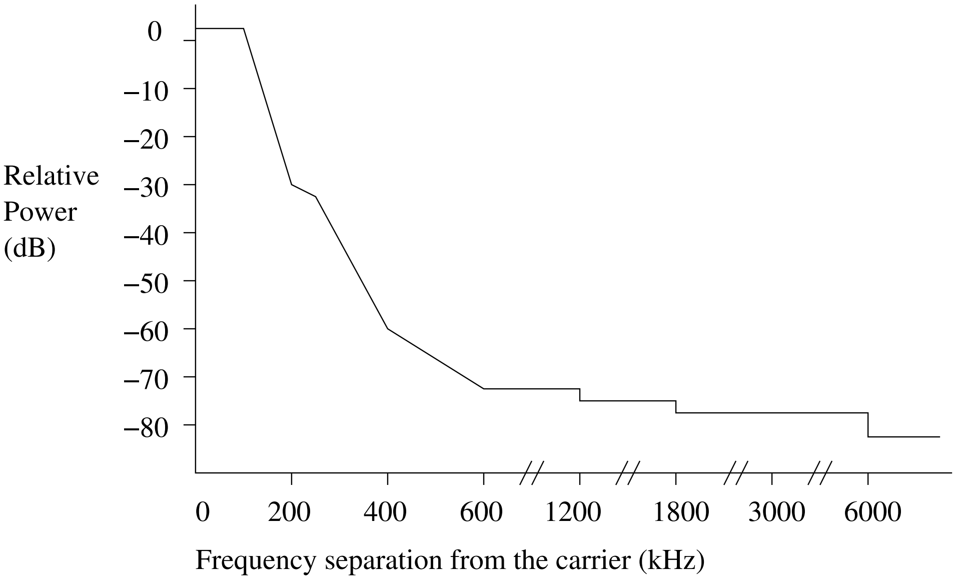


Figure 4.2: Transmission mask for GSM base station

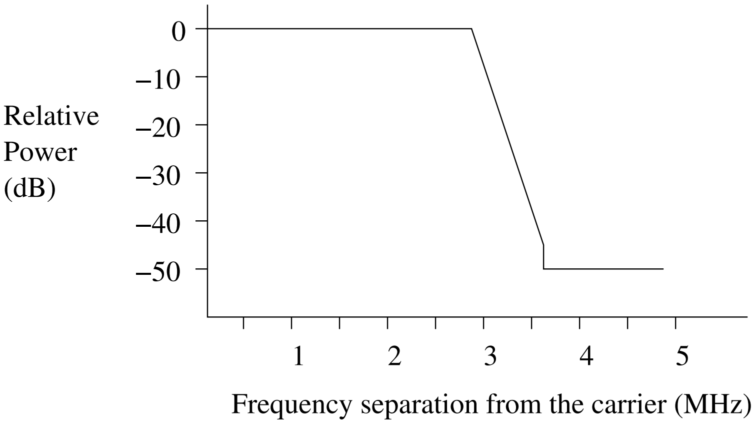


Figure 4.3: Transmission mask for UMTS base station

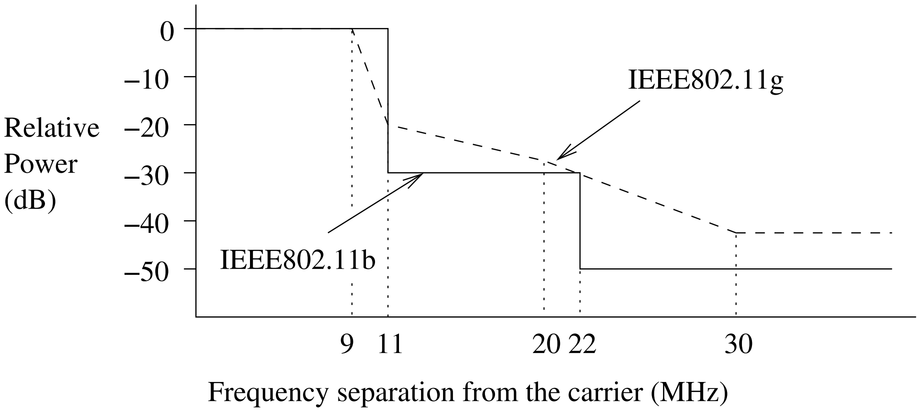


Figure 4.4: Transmission masks for IEEE802.11b and IEEE802.11g

PA has to satisfy the transmission masks; at any frequency, it should not transmit more power than indicated by the mask. In the remainder of this thesis we will not concentrate on a specific mask but look at algorithms to reduce the Adjacent Channel Interference levels in general. Whenever appropriate, references to mainly the masks for UMTS, IEEE802.11b and IEEE802.11g will be given.

4.2.2 Algorithm specification

The second task within the functional specification is the algorithm specification. The algorithm determines the non-linear behavior of the PA. We will first give an introduction to memory polynomials. Memory polynomials are used within the LS predistorter to estimate the non-linear behavior of the PA. After that, the LS predistorter algorithm is described. In this description, we assume that no ADCs are present in the feedback path, so the signal at the output of the PA model (x_{pd}) equals the signal presented to the Predistortion Control block (y), see figure 3.1.

4.2.2.1 Introduction to memory polynomials

In section 3.5, we have described a specific model of a PA, the Wiener-Hammerstein model. It determines the relation from input to output of the PA. It is used to 'generate' distortion. Within the Predistortion Control block, we have to determine the non-linear behavior of the PA model, using the signals at the input and output of the PA model. One option is to make use of the knowledge that the PA model is a Wiener-Hammerstein model and that we try to determine its parameters. However, this model is very specific. When other PA-models are used to generate the distortion, assuming a Wiener-Hammerstein model within the Predistortion Control block, might not lead to satisfactory results. For that reason, we want to use a more general model. Such a model, presented in [61], is described below.

If the output y at a certain moment in time t only depends on the input x_p at that specific moment in time, the distortion is generally described by:

$$y(t) = D\big(|x_p(t)|\big) \cdot x_p(t), \text{ where } D : \mathbb{R}_+ \to \mathbb{C} \tag{4.1}$$

The function D will be described in more detail below. For many PAs, the output at a certain moment in time does not only depend on the input at that moment but also on the input at previous moments (history). Then the PA suffers from so-called memory effects. In the digital domain, this means that a sample of y at time t does not only depend on the value of x_p at time t but also on previous samples:

$$\begin{aligned} y(t) =& D_0\big(|x_p(t)|\big) \cdot x_p(t)+ \\ & D_1\big(|x_p(t-1)|\big) \cdot x_p(t-1)+ \\ & D_2\big(|x_p(t-2)|\big) \cdot x_p(t-2) + ... \end{aligned} \tag{4.2}$$

In PA models, the functions D_i are very often approximated by polynomials:

$$D_i(x) = a_{0i} + a_{1i}x + a_{2i}x^2 + a_{3i}x^3 + ... \tag{4.3}$$

In this expression, all orders are given up to infinity. Normally, only a limited number of expressions is used. A general description of a PA model combines expressions 4.2 and 4.3, limits the order of the polynomials from 1 to K and the memory depth from 0 to $\tau_{max} - 1$, which leads to:

$$y(t) = \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} a_{k\tau} x_p(t-\tau) \mid x_p(t-\tau) \mid^{k-1} \tag{4.4}$$

The relation is said to be described by a memory polynomial. The order of the polynomial is limited by K and the number of samples from the past taken into account is determined by τ_{max}. τ_{max} is called the memory length.

The non-linearity model, presented in expression 4.4 is related to a Volterra model. This is explained below.

4.2.2.2 Relation to the Volterra model

A Volterra model of a weak non-linearity is presented in [68], section 4.2. A discrete version of the Volterra model is presented in [69]. This discrete Volterra model is:

$$w(t) = h_0 + \sum_{k=1}^{\infty} \overline{h}_k[s(t)] \tag{4.5}$$

where

$$\overline{h}_k[s(t)] = \sum_{\tau_1=-\infty}^{\infty} ... \sum_{\tau_k=-\infty}^{\infty} h_k(\tau_1, ..., \tau_k) \cdot s(t-\tau_1)\cdot, ..., \cdot s(t-\tau_k) \tag{4.6}$$

The multidimensional functions $h_k(\tau_1, ..., \tau_k)$, called Volterra kernels, characterize the non-linear behavior. They can be used to model a non-linear PA. Suppose the PA (not the baseband equivalent PA model) is modeled as a discrete Volterra model. Refering to figure 2.7, we rewrite expression 4.6:

$$y_{rf}(t) = h_0 + \sum_{k=1}^{\infty} \overline{h}_k[x_{rf}(t)] \tag{4.7}$$

where

$$\overline{h}_k[x_{rf}(t)] = \sum_{\tau_1=-\infty}^{\infty} ... \sum_{\tau_k=-\infty}^{\infty} h_k(\tau_1, ..., \tau_k) \cdot x_{rf}(t-\tau_1)\cdot, ..., \cdot x_{rf}(t-\tau_k) \tag{4.8}$$

When using this model, an infinite number of kernels have to be determined. We will first simplify expression 4.8. In expression 4.8, future amplitudes of x_{rf} are taken into consideration. Because PAs are causal systems, the following holds: $h_k(\tau_1, ..., \tau_k) = 0, if any \tau_i < 0, i = 1, ..., k.$

Furthermore we simplify the expression by assuming that $h_k(\tau_1, ..., \tau_k) = 0$, if $\exists\ i, j$ where $\tau_i \neq \tau_j$, for $i, j = 1, ..., k$. The last simplifications are: the limitation of the summation over τ and k to $\tau_{max} - 1$ and K respectively, and the assumption that h_0 equals zero. The resulting expression becomes:

$$y_{rf}(t) = \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} h_k(\tau) \cdot x_{rf}(t-\tau)^k \tag{4.9}$$

We want to obtain an expression like 4.9 in case the input and output signals are the complex envelopes of the actual signals and the non-linearity is characterized by equivalent low pass Volterra kernels. In [70], p. 69, it is suggested to write x_{rf} as:

$$x_{rf}(t) = \text{Re}[x_p(t) \cdot e^{i2\pi f_c t}] \tag{4.10}$$

where x_p is the predistorted digital baseband signal (see figure 2.7) and f_c is the carrier frequency. Expression 4.10 can be used to rewrite expression 4.9:

$$y_{rf}(t) = \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} h_k(\tau) \cdot \left(\mathrm{Re}\left[x_p(t-\tau) \cdot e^{i2\pi f_c(t-\tau)} \right] \right)^k \tag{4.11}$$

For predistortion of PAs, only the frequency components of y_{rf} around the carrier frequency f_c are of interest. These frequencies form the so-called 'first' or 'principal zone'. $y_1(t)$ is defined as that part of the signal y_{rf} which has its frequency components within the principal zone. In [71], p. 18, it is shown that $y_1(t)$ can be expressed as:

$$\begin{aligned} y_1(t) = {} & \mathrm{Re}\left[\left\{ \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} h'_k(\tau) \cdot \left(\frac{|x_p(t-\tau)|}{2} \right)^k \right\} \cdot e^{i\angle x_p(t-\tau)} \cdot e^{i2\pi f_c t} \right] \\ = {} & \mathrm{Re}\left[\sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} \frac{h'_k(\tau)}{2^k} \cdot |x_p(t-\tau)|^{k-1} \cdot |x_p(t-\tau)| \cdot e^{i\angle x_p(t-\tau)} \cdot e^{i2\pi f_c t} \right] \\ = {} & \mathrm{Re}\left[\sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} \tilde{h}_k(\tau) \cdot x_p(t-\tau) \cdot |x_p(t-\tau)|^{k-1} \cdot e^{i2\pi f_c t} \right] \end{aligned} \tag{4.12}$$

where $h'_k(\tau)$ and $\tilde{h}_k(\tau)$ are functions of $h_k(\tau)$ and the carrier frequency f_c. Referring to figure 2.7, the expression for the digital baseband signal $y(t)$ becomes:

$$y(t) = \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} \tilde{h}_k(\tau) \cdot x_p(t-\tau) \mid x_p(t-\tau) \mid^{k-1} \tag{4.13}$$

This expression equals expression 4.4 where the low pass Volterra kernels $\tilde{h}_k(\tau)$ have a similar function as the polynomial coefficients $a_{k\tau}$.

4.2.2.3 The LS predistorter algorithm

In the Predistortion Control block (see figure 3.2), the predistorted signal x_p and the PA model output y are available and are used to describe the behavior of the PA model by means of memory polynomials. The memory polynomial, given in expression 4.4, can be used to determine the values $a_{k\tau}$. These values determine the behavior from input x_p to output y. In section 2.1, we have defined this approach as the Direct Learning approach. The Predistortion block has to invert this relation to correctly predistort the incoming signal x. In [62], it was suggested to describe the input x_p as a function of the output y. This can be done by means of a memory polynomial as well (see [61] and [64]):

$$x_p(t) = \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} a_{k\tau} y(t-\tau) \mid y(t-\tau) \mid^{k-1} \tag{4.14}$$

This approach is called 'Indirect Learning'. Expression 4.14 can be obtained from expression 4.4 by exchanging x_p and y. If the Predistortion Control block uses the Indirect Learning approach to describe the relation between x_p and y, the polynomial coefficients $a_{k\tau}$ can directly be used by the Predistortion block to predistort the incoming signal x. This is an advantage because no inversion procedure is needed. It is important to note that the values $a_{k\tau}$ in expression 4.14 differ from the values $a_{k\tau}$ in expression 4.4. In expression 4.4 (Direct Learning) they describe the relation from input to output, in expression 4.14 (Indirect Learning) they describe the relation from output to input. We will develop the LS predistorter using the Indirect Learning approach. Furtheron we will generalize the predistorter to include Direct Learning as well.

The problem we are confronted with is to find the polynomial coefficients $a_{k\tau}$ given x_p, y and a set of equations described by expression 4.14. The set of equations can be solved using linear algebra. For that purpose we rewrite expression 4.14. For the sake of convenience we redefine $\varsigma = \tau_{max} - 1$.

$$\begin{array}{rcllll} x_p(t) & = & a_{10}y(t) & + & ... & + \quad a_{K0}y(t)|y(t)|^{K-1} \\ & + & a_{11}y(t-1) & + & ... & + \quad a_{K1}y(t-1)|y(t-1)|^{K-1} \\ & + & ... & & & \\ & + & a_{1\varsigma}y(t-\varsigma) & + & ... & + \quad a_{K\varsigma}y(t-\varsigma)|y(t-\varsigma)|^{K-1} \end{array} \tag{4.15}$$

To simplify this expression, we define:

$$y_{k\tau}(t) = y(t-\tau) \mid y(t-\tau) \mid^{k-1} \tag{4.16}$$

Expression 4.15 can then be rewritten into:

$$\begin{array}{rcllll} x_p(t) & = & a_{10}y_{10}(t) & + & ... & + \quad a_{K0}y_{K0}(t) \\ & + & a_{11}y_{11}(t) & + & ... & + \quad a_{K1}y_{K1}(t) \\ & + & ... & & & \\ & + & a_{1\varsigma}y_{1\varsigma}(t) & + & ... & + \quad a_{K\varsigma}y_{K\varsigma}(t) \end{array} \tag{4.17}$$

In vector notation this can be written as:

$$x_p(t) = [y_{10}(t), ..., y_{K0}(t), ..., y_{1\varsigma}(t), ..., y_{K\varsigma}(t)] \begin{bmatrix} a_{10} \\ ... \\ a_{K0} \\ ... \\ a_{1\varsigma} \\ ... \\ a_{K\varsigma} \end{bmatrix} \tag{4.18}$$

In this expression, $x_p(t)$ and $y_{10}(t)$ to $y_{K\varsigma}(t)$ are given and a_{10} to $a_{K\varsigma}$ are unknown. Expression 4.18 cannot be solved because the number of unknown values $(K \cdot \tau_{max})$ exceeds the number of equations, which equals one. We therefore need at least $K \cdot \tau_{max}$ equations similar to equation 4.18. This can be realized by using more samples, say T samples ranging from 1 to T. We assume that the behavior of the PA does not change during the time T. The vector **a** is defined as:

$$\mathbf{a} = [a_{10}, ..., a_{K0}, ..., a_{1\varsigma}, ..., a_{K\varsigma}]^T \tag{4.19}$$

We obtain the following T equations:

$$\begin{array}{lcl} x_p(1) & = & [y_{10}(1) \quad y_{20}(1) \quad ... \quad y_{K\varsigma}(1)]\mathbf{a} \\ x_p(2) & = & [y_{10}(2) \quad y_{20}(2) \quad ... \quad y_{K\varsigma}(2)]\mathbf{a} \\ ... & & \\ x_p(T) & = & [y_{10}(T) \quad y_{20}(T) \quad ... \quad y_{K\varsigma}(T)]\mathbf{a} \end{array} \tag{4.20}$$

We define the columns of this expression as separate column vectors $\mathbf{x}_p$ and $\mathbf{y}_{k\tau}$ and combine the column vectors $\mathbf{y}_{k\tau}$ into a matrix $\mathbf{Y}$:

$$\mathbf{x}_p = [x_p(1), ..., x_p(T)]^T \tag{4.21}$$

$$\mathbf{y}_{k\tau} = [y_{k\tau}(1), ..., y_{k\tau}(T)]^T \tag{4.22}$$

$$\mathbf{Y} = [\mathbf{y}_{10}\mathbf{y}_{20}, ..., \mathbf{y}_{K\varsigma}] \tag{4.23}$$

Expression 4.20 can than be rewritten in a very compact way:

$$\mathbf{x}_p = \mathbf{Ya} \tag{4.24}$$

If the inverse behavior of the PA model (inverse because of the Indirect Learning approach) exactly matches a memory polynomial of order K and memory depth τ_{max}, we need exactly $K \cdot \tau_{max}$ equations to determine the $K \cdot \tau_{max}$ unknown values. This means that the number of samples T equals the number of unknown values $K \cdot \tau_{max}$. As a consequence, the matrix $\mathbf{Y}$ is square containing $(K \cdot \tau_{max})^2$ elements.

Expression 4.24 is then easily solved by using the inverse of matrix $\mathbf{Y}$ (under the condition that the inverse exists):

$$\hat{\mathbf{a}} = \mathbf{Y}^{-1}\mathbf{x}_p \tag{4.25}$$

Because of the assumption that the inverse behavior of the PA model exactly matches a memory polynomial, the estimated polynomial coefficients exactly equal the real polynomial coefficients: $\hat{\mathbf{a}} = \mathbf{a}$. In practice however, the inverse behavior of the PA model does not exactly match a memory polynomial. Thus, the non-linear behavior, determined by the polynomial coefficients $\hat{\mathbf{a}}$, is not exactly equal to the inverse behavior of the PA model. If we determine, in that case, an estimate of the original signal $\mathbf{x}_p$, using estimated polynomial coefficients $\hat{\mathbf{a}}$ and the output of the PA gathered in $\mathbf{Y}$, we obtain

$$\hat{\mathbf{x}}_p = \mathbf{Y}\hat{\mathbf{a}} \tag{4.26}$$

where $\hat{\mathbf{x}}_p - \mathbf{x}_p \neq \mathbf{0}$. Since it is not possible to find the exact solution, we have to find the best approximation. To determine which approximation is the best, we have to define a criterion. The criterion used for the LS predistorter is the Least Squares criterion. The Square Error is defined as:

$$SE = \sum_{t=1}^{T} \left|\hat{x}_p(t) - x_p(t)\right|^2 \tag{4.27}$$

If expression 4.25 is used to calculate the estimated polynomial coefficients $\hat{\mathbf{a}}$ and if expression 4.26 is used to calculate an estimate of the original signal $\hat{\mathbf{x}}_\mathbf{p}$, it is shown in [72], p. 378, that the Square Error is minimized. A prerequisite is that the matrix $\mathbf{Y}$ is square and its inverse exists. A better estimate of the non-linear behavior of the PA is obtained if more samples than the minimum number of samples required, are used. This means: $T > K \cdot \tau_{max}$. As a consequence, the matrix $\mathbf{Y}$ becomes rectangular with size $T \times K \cdot \tau_{max}$. A rectangular matrix does not have an inverse. However, a pseudo inverse can be used. This pseudo inverse is defined as $(\mathbf{Y}^H\mathbf{Y})^{-1}\mathbf{Y}^H$ where ^{H} indicates the Hermitian operation [1]. In [72], p. 378, it is shown that in case of a rectangular matrix $\mathbf{Y}$, the Least Squares solution equals:

$$\hat{\mathbf{a}} \;=\; (\mathbf{Y}^H\mathbf{Y})^{-1}\mathbf{Y}^H\mathbf{x}_p \tag{4.28}$$

[1] If matrix $\mathbf{B}$, consisting of elements $b^r_{ij} + i \cdot b^i_{ij}$ is the Hermitian matrix of $\mathbf{A}$, consisting of the elements $a^r_{ij} + i \cdot a^i_{ij}$, then $b^r_{ij} = a^r_{ji}$ and $b^i_{ij} = -a^i_{ji}$

4.2.2.4 Extension of the LS predistorter

Besides the Indirect Learning scheme, the Direct Learning scheme can be used as well. In the Direct Learning scheme, the output y of the PA model is written as a function of the input x_p of the PA model, as indicated by expression 4.4. It is however more convenient to use one equation which can be used for both the Direct- and Indirect Learning schemes, using general signal names x_1 and x_2:

$$x_2(t) = \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} a_{k\tau} x_1(t-\tau) \mid x_1(t-\tau) \mid^{k-1} \tag{4.29}$$

The Indirect Learning scheme is used when $x_1 = y$ and $x_2 = x_p$. The Direct Learning scheme is used when $x_1 = x_p$ and $x_2 = y$.

In expression 4.29, powers of x_1, $x_1(t-\tau)|x_1(t-\tau)|^{k-1}$, are used. We define a 'basis function' ϕ_k as: $\phi_k(x_1(t-\tau)) = x_1(t-\tau)|x_1(t-\tau)|^{k-1}$. For every value k, ϕ_k is a polynomial consisting of a single product term. These single-term polynomials will be called 'normal polynomials'.

In the presence of quantization noise and when using finite precision digital processing, the inversion of the matrix $\mathbf{Y}^H\mathbf{Y}$ (expression 4.28) is prone to numerical instabilities due to the use of normal polynomials. In [65], an alternative for the use of normal polynomials is proposed: orthogonal polynomials. A system of polynomials $\psi_k(x)$ is defined to be orthogonal, with respect to a weight function $w(x)$, on the interval $a \leq x \leq b$, if:

$$\int_a^b w(x)\psi_n(x)\psi_m(x)dx = 0 \tag{4.30}$$

where $n \neq m,\ n, m = 0, 1, 2...$ and $w(x) \geq 0$.

In our application, we want the polynomials to be orthogonal with respect to the probability density of the absolute value of the input signal, so $w(x)$ equals the probability density of $|x|$. The orthogonal polynomials presented in [65] are based on a uniform distribution of the amplitude of x, where $0 \leq |x| < 1$. In our case, using the signal source as described in section 3.3, the signal x consists of a real- and an imaginary component which behave approximately as Gaussian noise sources. The amplitude of the resulting signal then approximately has a Rayleigh distribution (see [14], pg. 161). We will, however, use the orthogonal polynomials as they were presented in [65] because, in that publication, it was shown to yield improvements for a Rayleigh distribution as well.

In our simulations we will compare the use of normal polynomials with orthogonal polynomials. An orthogonal polynomial basis function of order k is denoted as ψ_k. In most applications only odd polynomial basis functions up to the

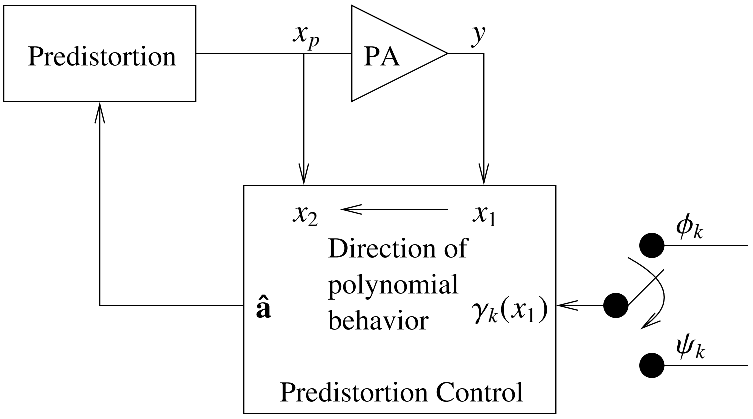


Figure 4.5: Schematic representation of Indirect Learning schemes

fifth order are used. When searching through literature, this appears to be common practice in digital predistortion design. Using as few polynomial degrees as possible reduces the complexity of the final digital predistorter. Furthermore, adding even polynomials and adding polynomials with a degree higher than five, does not imply a fundamental change of the algorithm. We will therefore only give the odd basis functions up to the fifth degree (for other basis functions see [65] and chapter 6):

$$\begin{aligned} \psi_1(x) =&\ x \\ \psi_3(x) =&\ 15 \mid x \mid^2 x - 20 \mid x \mid x + 6x \\ \psi_5(x) =&\ 210 \mid x \mid^4 x - 504 \mid x \mid^3 x + 420 \mid x \mid^2 x \\ &- 140 \mid x \mid x + 15x \end{aligned} \tag{4.31}$$

If we generally define a basisfunction by $\gamma_k(x)$, four different LS predistorters can be identified. The first two predistorters use the Indirect Learning scheme and are illustrated schematically in figure 4.5.

For the Indirect Learning schemes, the relation from output to input is described by a polynomial. There are two choices for polynomial basis function: normal polynomials (ϕ_k) and orthogonal polynomials (ψ_k). The estimate $\hat{\mathbf{a}}$ is directly used to program the predistorter. A similar schematic representation for the Direct Learning scheme is given in figure 4.6.

For the Direct Learning scheme, the relation from input to output is described by a polynomial. Again there are two choices for polynomial basis function: normal polynomials (ϕ_k) and orthogonal polynomials (ψ_k). The estimate $\hat{\mathbf{a}}$ is inverted before it is being used for predistortion. All four different predistorters can be defined within a single framework:

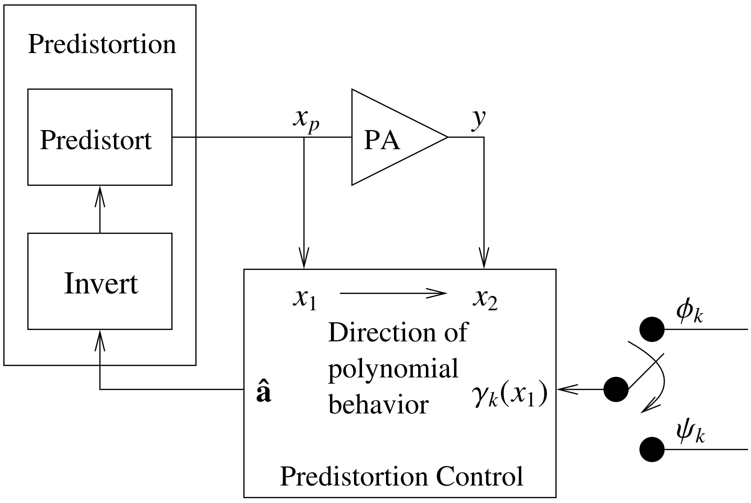


Figure 4.6: Schematic representation of Direct Learning schemes

1. Indirect Learning using normal polynomials ϕ_k: $x_1 = y$, $x_2 = x_p$, $\gamma_k = \phi_k$.
2. Indirect Learning using orthogonal polynomials ψ_k: $x_1 = y$, $x_2 = x_p$, $\gamma_k = \psi_k$.
3. Direct Learning using normal polynomials ϕ_k: $x_1 = x_p$, $x_2 = y$, $\gamma_k = \phi_k$.
4. Direct Learning using orthogonal polynomials ψ_k: $x_1 = x_p$, $x_2 = y$, $\gamma_k = \psi_k$.

Defining $\gamma_{k\tau}(t) = \gamma_k(x_1(t-\tau))$ for $t \in \mathbb{Z}$ and using the definitions of x_1 and x_2, a generalized version of expression 4.29 equals:

$$x_2(t) = \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} a_{k\tau}\gamma_{k\tau}(t) \tag{4.32}$$

A general description of the LS predistorter then becomes:

$$\begin{aligned}
\mathbf{x}_2 &= [x_2(1), ..., x_2(T)]^T && (4.33)\\
\mathbf{y}_{k\tau} &= [\gamma_{k\tau}(1), ..., \gamma_{k\tau}(T)]^T && (4.34)\\
\mathbf{\Gamma} &= [\mathbf{y}_{10}, ..., \mathbf{y}_{K0}, ..., \mathbf{y}_{1\tau_{max}-1}, ..., \mathbf{y}_{K\tau_{max}-1}] && (4.35)\\
\mathbf{x}_2 &= \mathbf{\Gamma a} && (4.36)
\end{aligned}$$

The estimates $\hat{\mathbf{a}}$ of the polynomial coefficients $\mathbf{a}$ are obtained via:

$$\hat{\mathbf{a}} = (\mathbf{\Gamma}^H\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^H\mathbf{x}_2 \tag{4.37}$$

where $\hat{\mathbf{a}}$ is the LS solution of the equation $\mathbf{x}_2 = \mathbf{\Gamma a}$. The expressions, elaborated for the four options, are given in appendix A.

4.2.2.5 The inverse of the PA memory polynomial

In the Indirect Learning scheme, the estimates of the polynomial coefficients ($\hat{\mathbf{a}}$) can directly be used in the Predistortion block; the incoming data x is predistorted using $\hat{\mathbf{a}}$. In the Direct Learning scheme, however, the inverse of the memory polynomial has to be determined. A procedure to approximate the inverse of a memory polynomial in case of a Direct Learning scheme, was presented in [61]. It is based on the general memory polynomial, used in the Direct Learning scheme with a memory length equal to two. The estimated behavior of the PA model is then described by expression 4.4 using the estimated polynomial coefficients $\hat{a}_{k\tau}$ where τ_{max} equals 2. Expression 4.4 can then be rewritten into another format:

$$y(t) = \beta_0(|x_p(t)|)x_p(t) + \beta_1(|x_p(t-1)|)x_p(t-1) \tag{4.38}$$

where

$$\begin{aligned} \beta_\tau(x) &= \sum_{k=1}^{K} \tilde{a}_{k\tau} x^{k-1} && (4.39) \\ \tilde{a}_{k\tau} &= \hat{a}_{k\tau} && (4.40) \end{aligned}$$

The purpose of the inversion algorithm is to determine the predistorted value $x_p(t)$ based on $y(t)$, $x_p(t-1)$ and $\hat{a}_{k\tau}$. Expression 4.38 is therefore rewritten as:

$$x_p(t) = \frac{1}{\beta_0(|\ x_p(t)\ |)}\left(y(t) - \beta_1(|\ x_p(t-1)\ |)x_p(t-1)\right) \tag{4.41}$$

The problem with this expression is that $|\ x_p(t)\ |$ needs to be known in order to calculate $x_p(t)$. So, before computing its value, $x_p(t)$ is already used. In [61], it is suggested to get around this problem by using $y(t)$ as an initial value ($x_p^{initial} = y(t)$). The algorithm then calculates a new estimate of $x_p(t)$, uses this as the next initial value, calculates a new estimate etc. This process is repeated until a stable estimate of $x_p(t)$ is obtained. In [61] it is indicated that a stable value is obtained after one or two iterations. However, from simulations we found that the algorithm only gives satisfactory results if the PA model satisfies two conditions.

First, the memory effects should not be severe which means that the β_1 values should be significantly smaller than the β_0 values. If this condition is not satisfied,

β_0 becomes small, destabilizing the algorithm. If, for example, the PA model only delays the signal with one sample, $\hat{a}_{k0} = 0, \quad \forall k$, which implies that $\beta_0(x) = 0$. This results in a 'division by zero' error in expression 4.41.

Second, the distortion should be weak; the first order parameter $\hat{a}_{10}$ should be close to 1. If the distortion of the PA model is severe ($\hat{a}_{10} << 1$), the difference between $x_p^{initial}$ (the initially assumed value of x_p) and the real value of x_p might be significant. This destabilizes the algorithm as well. To cope with amplifiers with severe distortion and memory effects, we modified the algorithm in two ways. First, in case of severe memory effects where $\hat{a}_{11} > \hat{a}_{10}$, we define $\hat{a}_{k2} = \hat{a}_{k0}$ and use an additional delay in the predistorter:

$$\begin{aligned} \tilde{a}_{k\tau} = & \quad \hat{a}_{k\tau} \text{ if } \hat{a}_{11} \leq \hat{a}_{10} \\ = & \quad \hat{a}_{k\tau+1} \text{ if } \hat{a}_{11} > \hat{a}_{10} \end{aligned} \tag{4.42}$$

This measure prevents β_0 from becoming too small. Second, instead of using $x_p^{initial}(t) = y(t)$ we used as initial value for $t > 1$:

$$x_p^{initial}(t) = \frac{1}{\tilde{a}_{10}} \left(y(t) - \beta_1(| \, x_p(t-1) \, |) x_p(t-1) \right) \tag{4.43}$$

The part between the large brackets takes into consideration the memory effects from the beginning. The effects of the first order component differing from 1 are canceled by division with $\tilde{a}_{10}$. For $t = 1$, we used: $x_p^{initial} = y(t)$. In all simulations where a Direct Learning scheme is involved, the inverse of the PA model is determined using this algorithm.

4.2.3 Implementation aspects

The complexity of the algorithm to estimate the polynomial coefficients a, is at least $O(T)$: the number of complex multiplications scales linearly with the number of samples T used to update the predistorter polynomial coefficients. To be more precise, in general $K\tau_{max}$ is small compared to T. Then, the complexity is basically determined by the following two operations: $\mathbf{\Gamma}^H\mathbf{\Gamma}$ and $\mathbf{\Gamma}^H\mathbf{x}_2$, where $\mathbf{\Gamma}$ is a $T \times K\tau_{max}$ matrix and $\mathbf{x}_2$ is a T-element column vector. The number of complex multiplications within $\mathbf{\Gamma}^H\mathbf{\Gamma}$ equals $(K\tau_{max})^2 \cdot T$ and within $\mathbf{\Gamma}^H\mathbf{x}_2$, the number of complex multiplications equals $K\tau_{max} \cdot T$. The matrix inversion $((\mathbf{\Gamma}^H\mathbf{\Gamma})^{-1})$ requires $(K\tau_{max})^3$ complex multiplications (see [73], pg. 48). The total number of complex multiplications therefore equals $K\tau_{max}(K\tau_{max} + 1) \cdot T + (K\tau_{max})^3 \approx K\tau_{max}(K\tau_{max} + 1) \cdot T$.

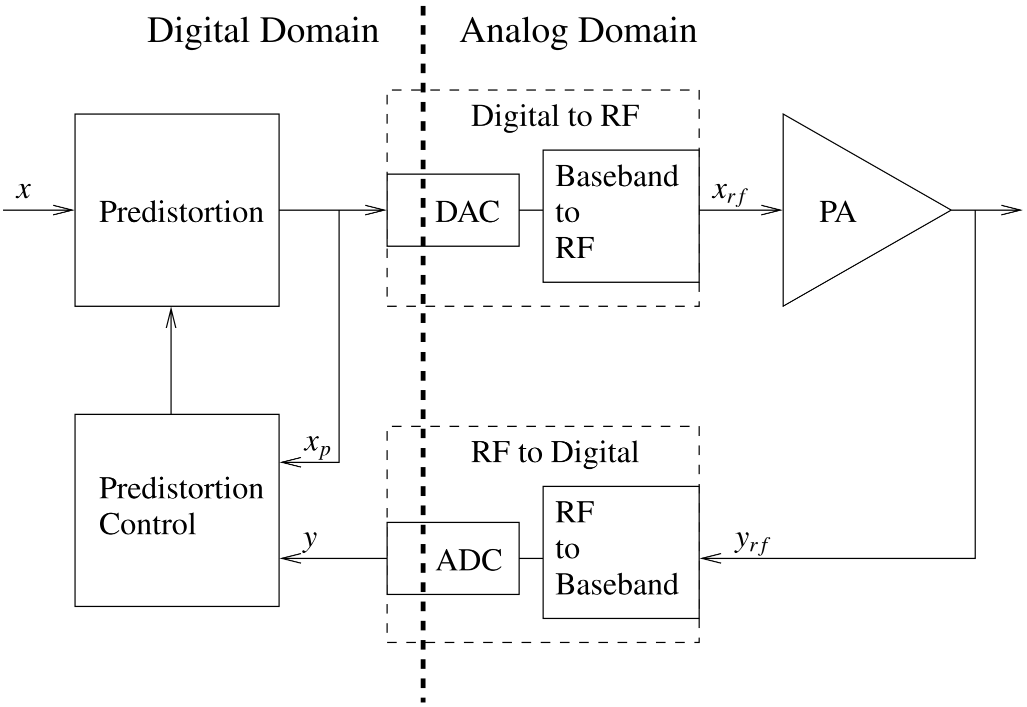


Figure 4.7: Mapping and architecture specification of the predistorter

4.3 Mapping

After the architecture specification and the functional specification of the first iteration of the Analog-Digital Codesign of a digital predistorter for PAs in telecommuncation transmitters, the mapping is the third phase. In the mapping phase, the application (algorithm) is mapped onto the set of architectures.

Within a telecommunication transmitter, the functionality involved in the digital predistortion at baseband is generally mapped onto the functional blocks as illustrated in figure 4.7.

The Predistortion and Predistortion Control blocks are mapped to the digital part of the system. The advantages, compared to analog predistortion and analog predistortion control, are that a greater degree of precision can be achieved when computing the predistortion coefficients (see [56]) because digital signal processing functions can be used of which analog equivalents are not available. Furthermore, instability due to temperature changes and ageing is of less concern and there are many options for the implementation of the digital signal processing part.

The first stage of the Digital to RF functionality consists of a Digital-to-Analog Converter (DAC) and the necessary analog band-limiting filters. The second stage consists of an analog frequency converter that converts the analog baseband signal into an analog RF signal. This RF signal is the input of the PA.

To monitor the behavior of the PA continuously, a feedback path is present. The purpose of the RF to Baseband functionality within the RF to Digital block is to convert the RF output of the PA to an analog baseband signal. The last stage of this functionality is the ADC.

According to this mapping, the design of the Predistortion and Predistortion Control blocks is part of the digital design process while the design of the Digital to RF and RF to Digital blocks is part of the analog design process. The design of the ADC and the DAC is included in the analog design because of the reasons mentioned in chapter 2.1. The design of the ADC is more demanding than the design of the DAC. Therefore, in the remainder of this thesis, the focus will be on the Analog-Digital Codesign of the Predistortion Control and the RF to Digital blocks.

Note that we exclude the use of digital IF. When using digital IF in for example the RF to Digital block, the RF signal is first converted to an intermediate frequency in the analog domain. A high frequency ADC then converts the analog signal into a digital-IF signal. In the digital domain, this signal is frequency converted from IF to baseband. Because we want to investigate the effects of quantization in the feedback path using a baseband-equivalent model, we do not analyze transmitters using digital-IF.

4.4 Performance analysis

4.4.1 Performance in case of high resolution quantization

The performance of the LS predistorter has been analyzed using the simulator described in section 3. Some general settings are summarized in table 4.1:

Parameter	Value	Function
T	8k	Samples within 1 cycle
N_{cycles}	40	Number of cycles
w	$\frac{1}{3}$	weight of $\hat{\mathbf{a}}_i$
M	9.8	Oversampling factor

Table 4.1: General simulator settings

We start the simulations without using an ADC in the feedback path. The output, obtained from the baseband equivalent PA model, is passed without quantization

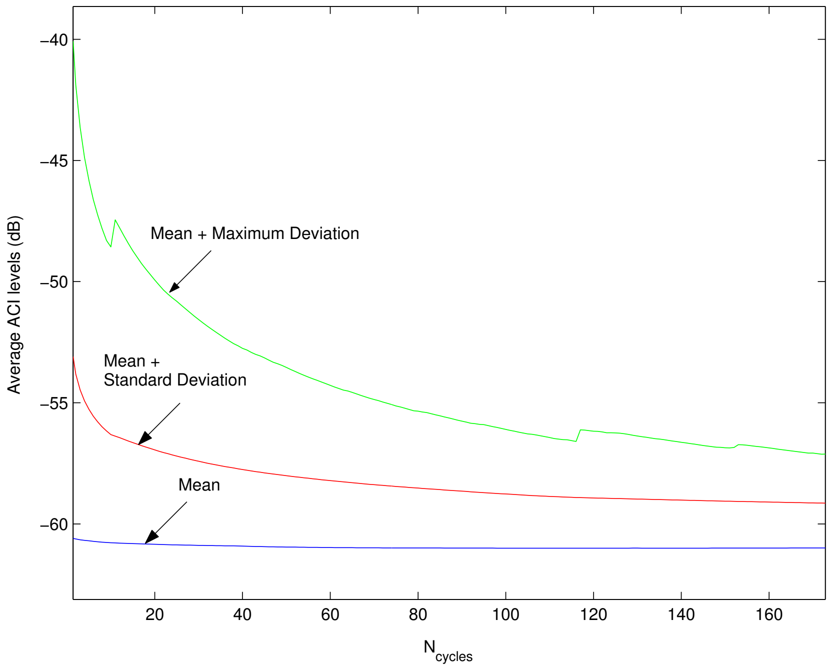


Figure 4.8: Accuracy of the ACI-level estimates as function of the number iterations

to the Predistortion Control block. For both polynomial types (normal and orthogonal) we only use odd polynomials up to the fifth degree.

As mentioned in table 4.1, the number of cycles on which the average performance of the predistorter is based equals 40. This number was chosen after simulations for different number of cycles (different values of N_{cycles}). In these simulations, we used the LS predistorter, Indirect Learning and normal polynomials. We simulated 1200 cycles and divided them into blocks of N_{cycles} cycles. Within one block, we calculate the average Adjacent Channel Interference (ACI) levels. So, for each value of N_{cycles}, we have a set of $1200/N_{cycles}$ average ACI levels and we calculated the mean, the standard deviation and the maximum deviation of these average ACI levels. For different values of N_{cycles}, these three values are given in figure 4.8.

We see that the standard- and maximum deviation reduce if the number of cycles N_{cycles}, over which the performance level is calculated, increases. This is to be expected; more data leads, in general, to a more accurate estimate. From that point of view N_{cycles} should be as large as possible. The simulation time however, increases linearly with the number of cycles N_{cycles}. A value of 40 for the number of cycles is a trade-off between accuracy and simulation time. For $N_{cycles} = 40$, the standard deviation on the estimate of the power in the adjacent channels is around

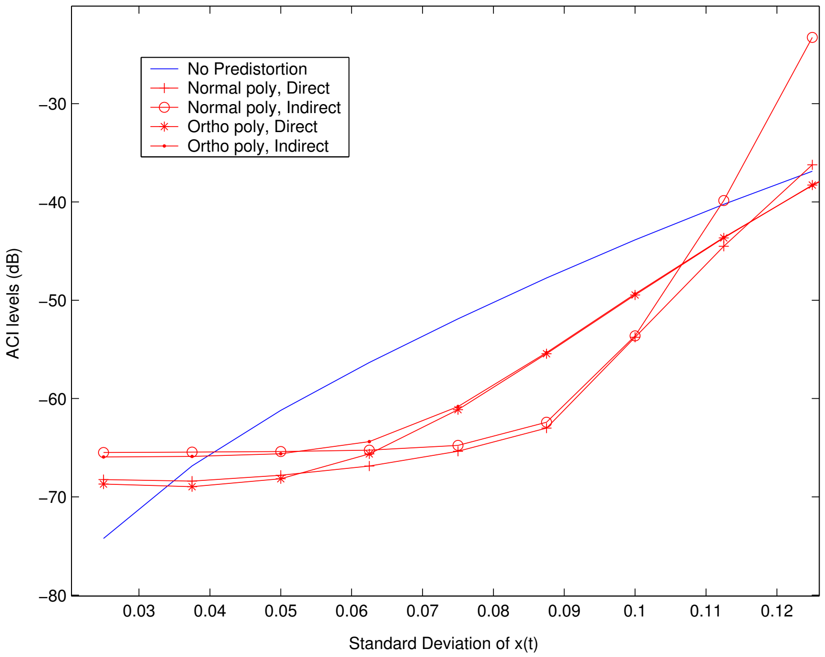


Figure 4.9: Performance LS predistorter

3 dB.

Using 40 cycles for every simulation run, we determined the performance (ACI levels) of the LS predistorter as a function of the standard deviation of the predistorter input signal x. For the simulations we used the Wiener-Hammerstein PA model given in section 3.5. The results are presented in figure 4.9.

The performance is presented for all four schemes: Direct Learning with normal polynomials, Indirect Learning with normal polynomials, Direct Learning with orthogonal polynomials and Indirect Learning with orthogonal polynomials. The range of the standard deviation of $x(t)$ is determined by the actual PA model (see section 3.5). The actual power of the input signal is not relevant, only the performance of a scheme in relation to the performance of other schemes is important. We see that for small standard deviations at the input (< 0.035), the distortion in case no predistortion is used is small (below -70 dB) and the LS predistorter does not give an improvement. On the contrary, the LS predistorter increases the power levels in the adjacent channels as can be seen from figure 4.9. For standard deviations larger than 0.035, the distortion increases and the LS predistorter gives a significant suppression of the ACI levels. The maximum suppression appears for a standard deviation at the input of approximately 0.09 when using normal polynomials. Simulation results are available for an input standard deviation of 0.0875. For this input level, the ACI levels are reduced from -47 dB

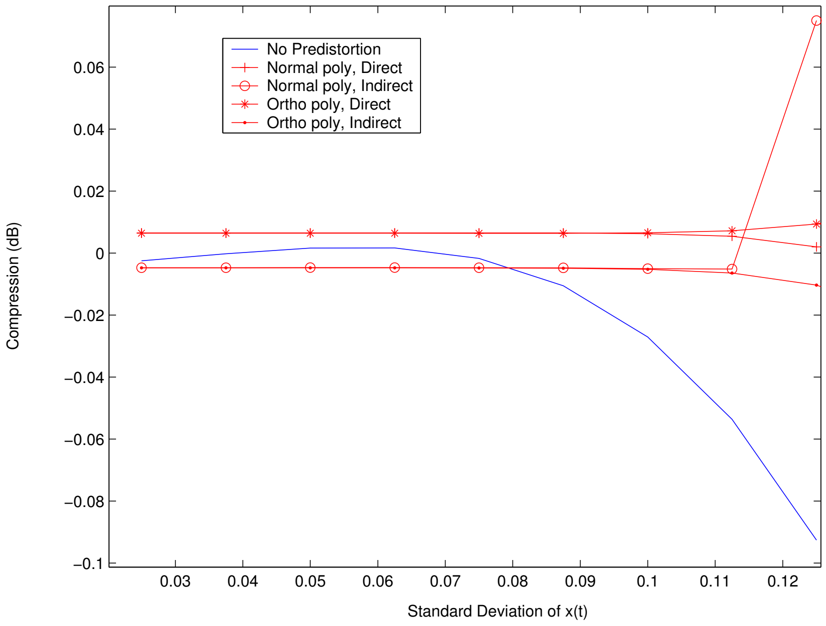


Figure 4.10: Compression LS predistorter

to -63 dB. For standard deviations roughly between 0.06 and 0.1, the LS predistorter using normal polynomials performs better than the LS predistorter using orthogonal polynomials. The reason for this is that odd normal polynomials up to the fifth degree better fit the PA model than odd orthogonal polynomials up to the fifth degree. This is because the non-linearity of the PA model is also described by odd normal polynomials up to the fifth degree (see section 3.5).

For large standard deviations (> 0.11), the suppression of ACI levels is negligible. When using normal polynomials in the Indirect Learning scheme, the predistorter even increases the ACI levels.

Besides the ACI levels, the compression of the Predistortion-PA model combination is an important measure of the performance. In the ideal case, the compression as defined in expression 3.18, equals 0 dB (the gain of the combination of the Predistortion block and the baseband-equivalent model of the PA, after correction with the linear gain G, equals one). The compression of the signals with and without predistortion is presented in figure 4.10.

From the simulations, we see that the compression in case no predistortion is used, is increasing for an increasing standard deviation. Note that more negative values indicate a larger compression. When using predistortion, the compression is reduced for all four schemes which implies that the Predistortion-PA model combination delivers more power than in case no predistortion is used, although the difference is very small. When using normal polynomials in the Indirect Learning scheme, the predistorter even increases the power level. In this case, the ACI levels are also increased (see figure 4.10) which indicates that the predistorter actually cannot deal with the increased power level correctly.

In general, the design of the predistorter-PA combination will be dimensioned in such way that, in all cases, the transmission mask of the applicable standard is always satisfied. If, for example, the maximum allowed ACI levels equals -60 dB, the maximum allowed standard deviation at the input of the predistorter equals approximately 0.08 in case predistortion is not active. For any predistortion scheme, if active, the maximum allowed standard deviation at the input of the predistorter is 0.1. So, the predistorter-PA combination can deliver approximately 2 dB more power than the PA without predistortion.

We conclude that the Predistortion-PA model combination behaves more linearly and leads to a lower compression than the PA without predistortion for a restricted range of the standard deviation of the input signal.

4.4.2 Effects of quantization

In [74], an analysis of quantization in digital predistortion is given. This analysis mainly deals with the effects of quantization in the digital part of the system. These effects are called 'limited wordlength effects'. Because of the continuing miniaturization in especially the digital domain, limiting the wordlength in the digital signal processing part of the predistortion system is not of utmost importance. However, for the ADC in the feedback path, wordlength (resolution) is important. Because of the increasing bandwidth and/or resolution requirements of recent communication standards, the specifications for the ADCs are demanding leading to relatively high costs and power consumption. Costs and power consumption can, in general, be reduced when ADCs with a lower resolution (less bits) are used (see figure 1.5). Using a lower resolution leads to an increase of quantization errors. For that reason, an ADC is included in the simulation model enabling the simulation of the effects of quantization errors.

Determining the absolute complexity of an ADC is not our main interest. We will concentrate on the relative complexity; will the complexity increase or decrease by changing some of the specifications. In general, by reducing the resolution (the number of bits) of an ADC, the complexity is reduced. If we look for example at a b-bit full-flash ADC with a voltage input (see [5], section 4.4), the input voltage is compared with $2^b - 1$ reference voltages. This means that all these voltages have to be generated and more importantly, $2^b - 1$ comparators have to be used for the comparison. The $2^b - 1$ comparator outputs have to be coded into b bits by digital logic. To code the outputs into b bits, $b(2^{b-1} - 1)$ comparisons have to be performed. Reducing the number of bits reduces the number of reference voltages to be produced, reduces the number of comparators and reduces the complexity of the digital coding part.

In [5], section 2.4, the accuracy or linearity of an ADC is specified through the Integral Non Linearity (INL) and Differential Non Linearity (DNL) specifications. The INL is the deviation of a reference voltage from the ideal value. The DNL describes the deviation from ideal for the difference between two adjacent reference voltages. Both INL and DNL are specified over the full range of the ADC input. It is common practice to specify INL and DNL in least significant bits (LSB). One LSB corresponds with the ideal difference between two adjacent reference voltages.

The linearity requirement or more precisely, the allowed non-linearity of the ADC in the RF to Digital block in the feedback path of a Digital Predistortion system, is determined by the non-linearity of the PA; the non-linearity of the ADC should be negligible compared to the non-linearity of the PA. When specifying the INL and DNL of the ADC in LSBs, one should be aware that the specifications change when changing the resolution of the ADC. Every time the number of bits is reduced by 1, the specification (in LSBs) should be divided by a factor 2. This reflects that the number of comparators and reference voltages is reduced when the resolution of the ADC is reduced but the accuracy of the remaining reference voltages and comparators, remains equal.

The effect of reducing the resolution of the ADC is that more quantization noise is produced. Suppose we have an ideal ADC (INL = 0, DNL = 0) with b bits and input range $[-A..A]$. An estimate of the power of the quantization noise then equals: $\frac{1}{12}(\frac{A}{2^{b-1}})^2$ (see [5], pg. 11). The amplitude peak-to-average ratio of the input signal is defined as k and the peak is the largest signal excursion at the input which still fits within the input range of the ADC. This implies that the power of the input signal equals $\frac{A^2}{k^2}$. The SNR, expressed in decibels, at the output then equals:

$$\begin{aligned} \text{SNR} = &\quad 10 \cdot \log\left(\frac{A^2/k^2}{\frac{1}{12}A^2/(2^{b-1})^2}\right) \\ = &\quad 6.02b + 4.77 - 20\log k \end{aligned} \tag{4.44}$$

For a sinusoidal input signal, covering the entire input range of the ADC, the amplitude peak-to-average ratio k equals $\sqrt{2}$ (see [5], p. 12).

Reducing the resolution of the ADC with one bit reduces the SNR at the output of the ADC with 6 dB.

As illustrated in section 1.3, figure 1.7, the quantization noise will not be independent of the original signal if the number of bits becomes small; there is correlation between the original signal and the quantization noise. A way to reduce this correlation is to add a so-called dither signal to the original signal (see

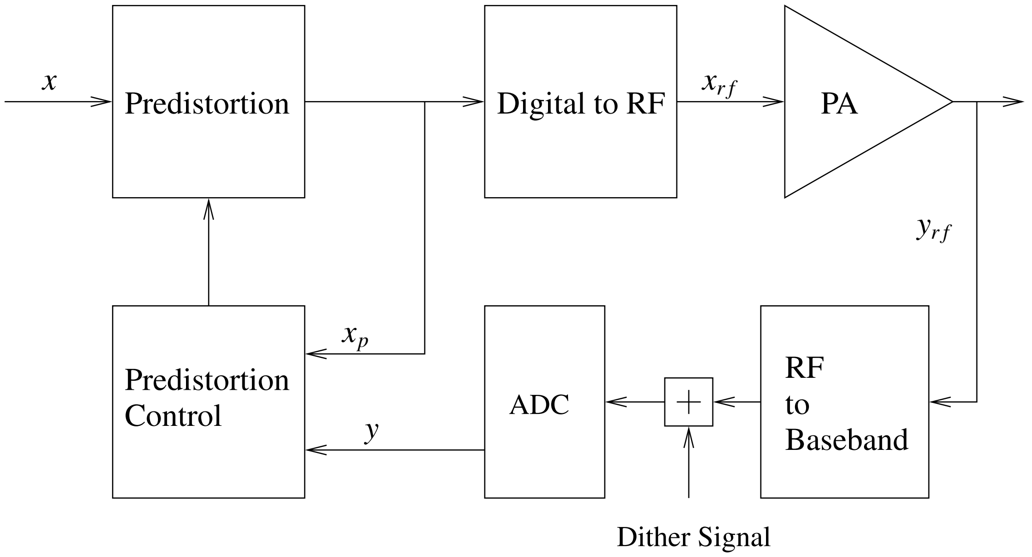


Figure 4.11: Addition of a dither signal in LS Predistortion

for example [75], [90] and [91]). For the LS predistorter, this means that a dither signal is added before the ADC, see figure 4.11.

For the analysis of the effects of quantization on the performance of the LS predistorter, the dither signal is a complex signal where the real and imaginary parts are independent. Both the real and imaginary parts have a uniform distribution and the range equals d LSB. The bandwidth equals the bandwidth of the original signal. A uniformly distributed dither signal with a range of d LSB roughly adds noise with power d^2 times the power of the quantization noise. Thus the SNR of the signal after adding the dither signal and after quantization becomes for d close to 1:

$$\begin{aligned} \text{SNR} = & \quad 10 \cdot \log \left(\frac{A^2/k^2}{\left(\frac{1}{12}A^2/(2^{b-1})^2\right) \cdot (1+d^2)} \right) \\ \approx & \quad 6.02b + 4.77 - 20 \log k - 3.01d \end{aligned} \tag{4.45}$$

4.4.3 Performance in case of coarse quantization

To analyze the effects of quantization in the feedback path on the digital signal processing algorithm, the standard simulator settings, in table 4.1, were used. Furthermore, we use a b-bits ADC in the feedback path. The Headroom factor H of the ADC equals 4.

As standard deviation for the input signal $x(t)$, we used the value where the best performance was obtained. This is an input standard deviation of 0.0875 (see

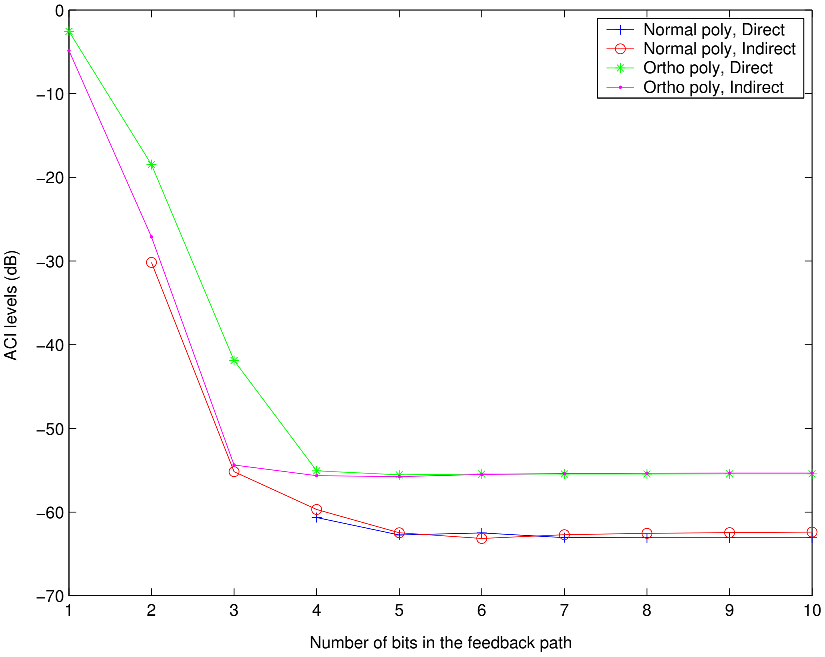


Figure 4.12: Quantization effects on the LS predistorter

figure 4.9). In figure 4.12, the performance as a function of the resolution of the ADC in the feedback path, is given. Note that we did not yet inject a dither signal.

The performance of the LS predistorter is stable for a resolution of 4 bits or more. For 2 and 3 bits, the performance decreases drastically. To understand this behavior, we investigated the effects of quantization more deeply.

If we consider quantization in the feedback path of the predistortion system, the PA output converted to baseband y (see figure 4.11) contains quantization noise and the power of this quantization noise depends on the resolution of the ADC in the feedback path. Modeling the effects of the noise on digital predistortion differs for the two learning schemes. We start with an analysis of the Direct learning scheme. In the Direct Learning scheme, the estimates of the coefficients of the memory polynomial are the elements of the solution of equation 4.36. We repeat this equation:

$$\mathbf{x}_2 = \mathbf{\Gamma} \cdot \mathbf{a} \tag{4.36}$$

Because of the Direct Learning scheme, x_2 corresponds with the feedback signal y and the elements of $\mathbf{\Gamma}$ are determined by x_p. If y is quantized, equation 4.36 becomes:

$$\mathbf{x}_2 + \mathbf{n} = \mathbf{\Gamma} \cdot \mathbf{a} \Rightarrow \mathbf{x}_2 = \mathbf{\Gamma} \cdot \mathbf{a} - \mathbf{n} \tag{4.46}$$

where $\mathbf{n} = [n(1), ..., n(T)]^T$ is a vector representing the quantization noise. We assume that the average or expected value of the elements of the noise vector $\mathbf{n}$ equals zero: $\mathrm{E}(\mathbf{n}) = \mathbf{0}$. Furthermore, we assume that y and x_p are wide sense stationary and the quantization noise is independent of y and x_p. We also assume that the quantization noise is independent for the different samples of y. For the formal description of this property, the dispersion matrix $\mathbf{V}$ is used. The dispersion matrix of the quantization noise vector $\mathbf{n}$ is defined as:

$$\mathbf{V}(\mathbf{n}) = \mathrm{E}(\mathbf{n}\mathbf{n}^H) \tag{4.47}$$

The dispersion matrix is often referred to as the ’dispersion’ or ’covariance matrix’. Also, the terms ’variance-covariance matrix’ and ’covariance matrix’ are used interchangeably.

The element on the i-th row and j-th column of $\mathbf{V}(\mathbf{n})$ is denoted as $\mathbf{V}_{ij}$ and indicates the dispersion between the i-th and j-th element of the noise vector $\mathbf{n}$. For the elements on the diagonal ($i = j$), the dispersion is called variance. The assumption that the quantization noise is independent for the different samples of y means that there is no correlation between the elements of the noise vector $\mathbf{n}$ which implies that $\mathbf{V}_{ij} = 0$ for $i \neq j$. Because y is a wide sense stationary process, the variance of the elements of the noise vector $\mathbf{n}$ is equal for all elements: $\mathbf{V}_{ii} = \sigma_q^2$, for $i = 1, ..., T$. The above is shortly described by:

$$\mathbf{V}(\mathbf{n}) = \mathrm{E}(\mathbf{n}\mathbf{n}^H) = \sigma_q^2\mathbf{I} \tag{4.48}$$

where $\mathbf{I}$ is the identity matrix with $\mathbf{I}_{ii} = 1$ for $i = 1, .., T$ and $\mathbf{I}_{ij} = 0$ for $i, j = 1, .., T$ and $i \neq j$. We will use the value σ_q^2 below.

Given expression 4.46: $\mathbf{x}_2 = \mathbf{\Gamma} \cdot \mathbf{a} - \mathbf{n}$, where $\mathbf{x}_2$ and $\mathbf{\Gamma}$ are known and $\mathrm{E}(\mathbf{n}) = \mathbf{0}$, an estimate $\hat{\mathbf{a}}$ of $\mathbf{a}$ is obtained by using the LS algorithm (see [72]). If quantization and dithering is used, an estimate $\tilde{\mathbf{a}}$ of $\hat{\mathbf{a}}$ is obtained. The expected value of $\tilde{\mathbf{a}}$ equals $\hat{\mathbf{a}}$: $\mathrm{E}(\tilde{\mathbf{a}}) = \hat{\mathbf{a}}$. The dispersion matrix of the estimate $\tilde{\mathbf{a}}$, which can be regarded as a measure of the noise on the estimates, equals (see [72], p. 378):

$$\mathbf{V}(\tilde{\mathbf{a}}) = \sigma_q^2(\mathbf{\Gamma}^H\mathbf{\Gamma})^{-1} \tag{4.49}$$

If T is increased, the vector $\mathbf{x}_2$ and the columns of the matrix $\mathbf{\Gamma}$ become longer (see expressions 4.33 to 4.35). If the columns of the matrix $\mathbf{\Gamma}$ become longer, the elements of the matrix $(\mathbf{\Gamma}^H\mathbf{\Gamma})$ generally become larger, especially the elements on the diagonal. Because of the inversion of the matrix $\mathbf{\Gamma}^H\mathbf{\Gamma}$, the elements of the dispersion matrix $\mathbf{V}(\hat{\mathbf{a}})$ become smaller for longer columns of $\mathbf{\Gamma}$. The most simple case is the case where $\mathbf{\Gamma}$ consists of a single column. Then, $\mathbf{\Gamma}^H\mathbf{\Gamma}$ becomes a single value equal to the sum of the squares of the absolute value of the elements of $\mathbf{\Gamma}$. If longer vectors are used, more values are added leading to larger values for $\mathbf{\Gamma}^H\mathbf{\Gamma}$. The inverse $(\mathbf{\Gamma}^H\mathbf{\Gamma})^{-1}$ becomes smaller, leading to a smaller dispersion value.

In general, if the length of the vectors is doubled (T is doubled), the values of the elements of the dispersion matrix $\mathbf{V}(\tilde{\mathbf{a}})$ are divided by a factor 2. So, the effects of quantization can be reduced by using more data as an input for the LS algorithm. The parameter T, which indicates the number of samples within one cycle, is therefore called 'Integration Length'. Expression 4.44 shows that the power of the quantization noise increases with a factor 4 if the number of bits b of the ADC is reduced with one. This means that for every bit less in the quantization process the length of the vectors needs to be quadrupled to obtain estimates of the polynomial coefficients a with equal accuracy.

When using coarse quantization (3 bits or less) in the feedback path, in combination with normal polynomials, simulations showed that the behavior of the LS predistorter becomes instable (see figure 4.12). The reason for this is that the inversion of the matrix $\mathbf{\Gamma}^H\mathbf{\Gamma}$, within expression 4.37, cannot always be executed. This stability issue is described in [65] where it is suggested to use orthogonal polynomials to solve the instability problem. To further analyze the effects of coarse quantization in the LS predistorter, we used orthogonal polynomials. Figure 4.13 gives the performance as a function of the length T of the vectors with and without dithering in case of 2-bit quantization in the feedback path using the Direct Learning scheme.

In case of 2-bit quantization in the feedback path and no dither signal added, it is clear that the LS predistorter increases the ACI levels dramatically. This is caused by the correlation between the quantization noise and the original feedback signal y. Because of this correlation, the performance does not improve if the length of the vectors T is increased. If a dither signal is added, the performance is improved and depends on the Integration Length T. If T is increased, the performance increases. For an Integration Length of 8k samples, the algorithm reduces the ACI levels below the ACI levels in case no predistortion is used. We have not simulated Integration Lengths larger than 32k. This is because simulation times become extremely long. Determining ACI levels in case of predistortion and 1 lsb dither, required approximately half a week simulation time on a desktop PC with an Intel Pentium 4 processor, running at 2.4 GHz. Because the trend of reduced ACI levels for longer Integration Lengths is clearly visible, we did not simulate beyond Integration Lengths of 32k samples.

The analysis, presented above, uses the Direct Learning scheme and orthogonal polynomials. Results of the Direct Learning scheme in combination with normal polynomials are not presented. This combination did not lead to stable

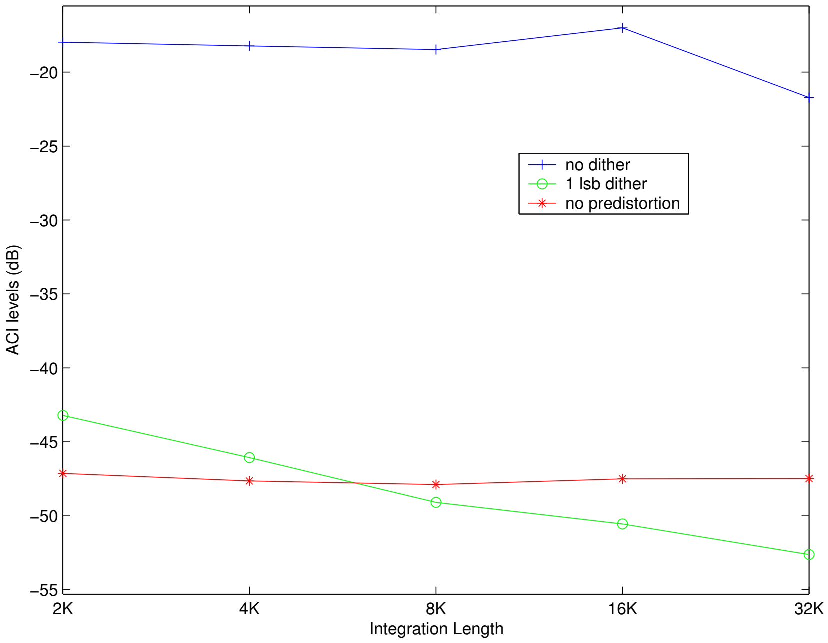


Figure 4.13: Performance LS predistorter with 2-bit quantization, Direct Learning and orthogonal polynomials.

behavior in case of 2-bit quantization, as was expected.

For the Indirect Learning scheme, the analysis using the dispersion matrix cannot be used. The quantization generates quantization noise (errors) on the PA output y and this quantized signal is used to fill the matrix $\mathbf{\Gamma}$, see expressions 4.34 and 4.35, where $\gamma_{k\tau}(t) = \gamma_k(y(t-\tau))$. For filling the matrix, powers of $y(t-\tau)$ are generated. Therefore, the elements of the matrix $\mathbf{\Gamma}$ have different noise powers. The effects of these noise powers on the final estimate $\tilde{\mathbf{a}}$ of $\hat{\mathbf{a}}$, are not known. As far as we know, they cannot be derived easily either. We therefore will only present the simulation results for 2-bit quantization using orthogonal polynomials (see figure 4.14) without further analysis.

The performance in case of predistortion and quantization is worse than in case no predistortion is used. Adding a dither signal does improve the performance to such an extend that a small suppression of ACI levels is obtained compared to the no-dither case. However, the ACI-levels are still above the levels in case no predistortion is used. Furthermore, increasing the Integration Length does not reduce the ACI levels. Obviously, quantization leads to systematic errors in the estimation of the polynomial coefficients, which limits the performance. A final remark is made concerning the polynomial coefficients $\mathbf{a}$, $\hat{\mathbf{a}}$ and $\tilde{\mathbf{a}}$. In the remainder of this thesis, we will use $\hat{\mathbf{a}}$ to indicate an estimate of $\mathbf{a}$, irrespective whether quantization is used or not and we will not use $\tilde{\mathbf{a}}$ anymore.

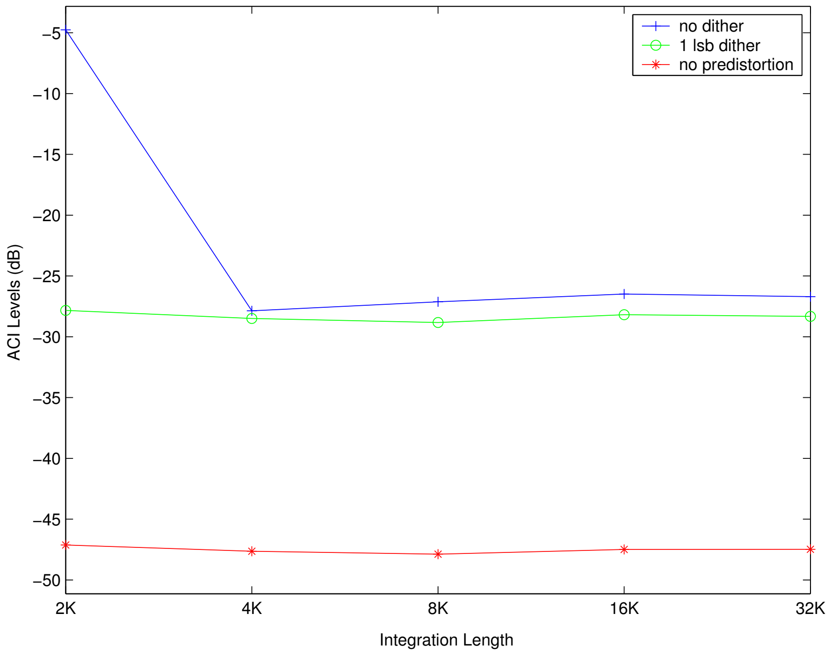


Figure 4.14: Performance LS predistorter with 2-bit quantization, Indirect Learning and orthogonal polynomials.

4.4.4 Overall system timing

In this section, we analyze the timing of the overall system for a typical scenario. The system timing depends on the value of the parameter w, the weight given to newly obtained predistorter settings (see section 3.7, expression 3.12). Furthermore, system timing is also based on more practical issues like the available processing capacity. Below we will describe the dependency on both aspects.

If newly obtained settings are given weight w, the old settings are given weight $1 - w$. For decreasing values of w, the contribution of previously obtained information concerning the non-linearity of the PA, is increasing. However, the system will respond more slowly to changes in the behavior of the PA.

In the simulations described in section 4.4.1, we used $w = \frac{1}{3}$. Reducing this value, giving the newly obtained estimates less importance, did not affect the performance of the algorithm considerably. Furthermore, after starting up the simulations, the algorithm gave stable reduction of ACI levels after 10 cycles where each cycle contains 8k samples. So in total, after 80k samples, the system stabilizes. This number of samples can be used to calculate the time it takes for the system to stabilize. If we refer to a UMTS system, the sampling frequency equals 49 MHz in case of an oversampling factor of 9.8 and a bandwidth of 5 MHz. 80k samples are obtained in approximately 1.7 ms. So, the system stabilizes after approximately

1.7 ms.

The behavior of the PA can change because of aging and temperature changes. Aging will surely not change the PA behavior on millisecond timescales. The temperature changes of the PA, on these timescales, can be caused by changes of the number of user-signals transmitted simultaneously via the PA (self-heating). The number of user-signals, transmitted on a single carrier by a UMTS base station, can change per UMTS frame of 10 ms. Thus, the number of users per carrier is constant within a frame. The time it takes the predistorter to adapt to a change in PA behavior and the time for which the PA behavior is constant, are within the same order of magnitude. For IEEE802.11b and IEEE802.11g the sampling frequency equals 20 MHz. 80k samples are obtained in approximately 4 ms. Since the power output of the PA is constant, self-heating is not an issue for these standards and the algorithm will be able to track the environmental temperature changes.

System timing is not only based on the value of the parameter w. The available processing capacity is important as well. There are numerous ways to implement the Predistortion Control block. We will not elaborate different alternatives but choose one implementation. For a typical scenario, we estimate the number of processor cycles that are needed on a Texas Instrument Floating Point Digital Signal Processor out of the C67x family. We chose this processor because it is tailored towards DSP applications, it is widely used and benchmarks are available (see [76]). Based on this, we estimate the amount of time needed to execute the predistortion control algorithm of the LS predistorter and determine the update rate of the predistorter.

As a typical scenario, we chose the UMTS application with sufficient resolution in the feedback path. We estimate the overall time it takes to sample data which acts as input for the LS predistorter, the overall time it takes for the algorithm to calculate new predistorter settings and the effects of these figures on the system functionality.

The time it takes to gather sufficient samples to calculate new predistorter settings is based on the sampling rate and the number of samples used by the LS predistorter. In our simulations, the number of samples, used initially, equals 8k (T = 8k) and the sampling rate equals 49 MHz. With this sampling rate, gathering 8k samples takes approximately 167 μs.

The tasks that have to be executed by the Predistortion Control block basically are:

- Generating the powers $\gamma_{k\tau}$ (see section 4.2.2.4). We use a memory depth of 2 ($\tau = 0, 1$) and only odd polynomials up to the fifth degree ($k = 1, 3, 5$). Therefore, to every sample, six values $\gamma_{k\tau}$ are related. Generation of these values can efficiently be implemented by means of a look-up table, realized in Random Access Memory (RAM) or Read Only Memory (ROM). These values can be generated in real-time, sample-by-sample. This process will therefore not influence the overall system timing.

- Executing the LS predistortion control algorithm: $\hat{\mathbf{a}} = (\mathbf{\Gamma}^H\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^H\mathbf{x}_2$ (expression 4.37). The result of the algorithm is the vector $\hat{\mathbf{a}}$. In the Indirect Learning approach, this vector, containing the coefficients of a memory polynomial, is directly used within the Predistortion block. In the Direct Learning approach, the Predistortion block has to use the inversion procedure, described in section 4.2.2.5, to predistort the signal. In this analysis, we restrict ourselves to the timing analysis of the Predistortion Control block and the consequences of its implementation on overall system behavior. Executing the inversion procedure, in case of Direct Learning, will be part of the Predistortion block.

In appendix C, the execution time for the LS predistorter on a 300 MHz Texas Instruments C67x processor, is estimated. The execution of the LS predistortion algorithm approximately takes 2.3 milliseconds.

So, processing 167 μs of data takes approximately 2.3 ms. Because we average new predistorter settings with old settings, it will take more than 2.3 ms to respond to changes of the non-linearity of the PA. The fraction of data that is used to track the behavior of the PA equals $167\ 10^{-6}/2.3\ 10^{-3} \times 100\ \% \approx 7.3\ \%$.

When using coarse quantization, the Integration length (the number of samples on which a new estimate of the predistorter non-linearity is based) has to increase. The required processing time will increase linearly with the Integration Length.

4.4.5 Evaluation of the LS predistorter

Based on the performance analysis of the LS predistorter, the conclusions are:

- The LS predistorter can be used to reduce the non-linearity of PAs.

- The complexity (the number of complex multiplications) of the Predistortion Control block scales linearly with the number of samples on which an estimate of the coefficients of a memory polynomial is based. The memory polynomial is used to model the non-linear behavior of the PA and the estimated coefficients are used to predistort the signal to be transmitted.

- The Indirect Learning scheme cannot be used in combination with coarse quantization. Furthermore, normal polynomials can also not be used in combination with coarse quantization.

The only scheme that can be used in combination with coarse quantization is the Direct Learning scheme using orthogonal polynomials. For this scheme:

- The effects of noise on the feedback signal can be reduced by using more data to estimate the polynomial coefficients.
- Coarse quantization in the feedback path, without using a dither signal, results in quantization noise which is partly correlated with the original signal. This prevents the LS predistorter from functioning correctly.
- The correlation between the quantization noise and the original feedback signal can be reduced by adding a dither signal to the signal before the ADC. We used uniformly distributed noise, independent for the real- and imaginary part.
- The resolution of the ADC in the feedback path can be interchanged with the Integration Length in case a dither signal is added. Using less bits in the quantization of the feedback signal can be compensated by using more data. Using more data implies that the complexity of the digital signal processing part increases.

Concerning the Analog-Digital Codesign of a digital predistorter for PAs, we have seen that the digital complexity of the LS predistorter is considerable. It scales linearly with the number of samples used to determine the non-linear behavior of the PA.

We have also seen that we can reduce the complexity of the analog part of the predistortion system by reducing the resolution of the ADC in the feedback path. However, in that case, more data is necessary to estimate the non-linearity of the PA and hence the complexity of the digital part is increased. Concerning the processing capacity needed to handle the increased complexity, there basically are two options:

- While increasing the size of the dataset because of reduced resolution of the ADC in the feedback path, the processing capacity is kept constant. The consequence is that, after capturing the necessary data, the time it takes to calculate new predistorter settings will increase. In applications where the non-linear characteristics of the PA only change slowly, this is acceptable. If the processing capacity is relatively small and the PA characteristics change relatively fast, the digital predistortion might become too slow.
- While increasing the size of the dataset, the processing capacity is increased accordingly, keeping the time it requires to calculate new predistorter settings constant. So, when reducing the complexity of the analog part by

reducing the resolution of the ADC in the feedback path, the digital processing capacity is increased.

The effects of both options, slower responses or increased complexity, are not always acceptable. For that reason, in the next chapter, we will search for a predistortion control algorithm:

- which has a lower complexity than the LS predistorter.
- of which the complexity does not scale with the number of samples on which new predistorter settings are based.

In the context of the Analog-Digital Codesign of a digital predistorter for PAs using the Y-chart approach, we decide to make use of the feedback path from 'Performance Numbers' back to 'Functional Specification' (see figure 1.11).

Chapter 5

The Crosscorrelation Predistorter

The most important conclusion of the performance analysis of the LS predistorter is that it is desirable to go through a second iteration of the design process according to the Y-chart approach. This second iteration has to start with the ’Functional Specification’ also called ’Application Design’. Within the application design phase, we have to develop a new algorithm which has a constant complexity, independent of the number of samples on which new predistorter settings are based. This new predistortion control algorithm will be called ’Crosscorrelation predistorter’ [1]. Besides this explicit feedback from ’Performance Numbers’ to ’Functional Specification’, there is no need to change the architectures. Within the second iteration, we use the same architecture as in the first iteration. So, within the second iteration there is no need for an ’Architecture Specification’ activity.

Because we plan to change the ’Application’, we need to check the ’Mapping’ after which we can start the ’Performance Analysis’ of the second iteration. The functional specification, mapping and performance analysis are described in this chapter.

5.1 Functional specification

5.1.1 ACI-level requirement

For the Crosscorrelation predistorter, we use the same requirements concerning the allowed ACI-levels as for the LS predistorter (see section 4.2.1).

[1] The Crosscorrelation predistorter has been the subject of the following publications: [4], [5], [6] and [7]

5.1.2 Algorithm specification

As for the LS predistorter, the Crosscorrelation predistorter is based on alternative 2 (see figure 5.1).

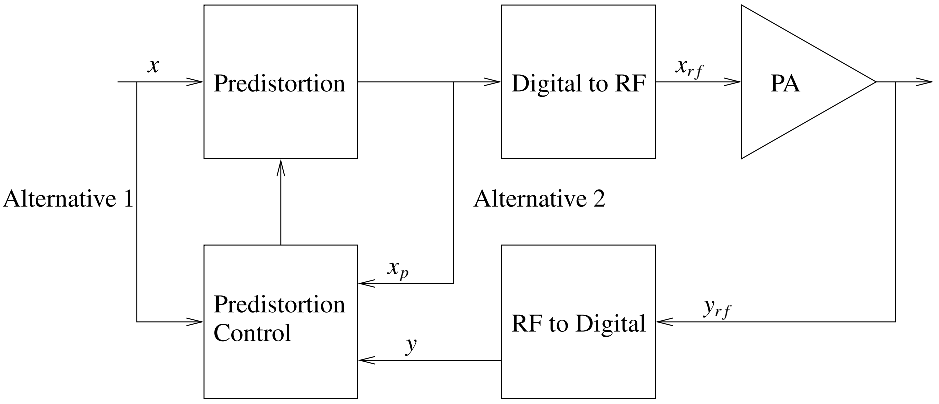


Figure 5.1: Block diagram of Digital Predistortion with feedback

The predistorted signal x_p and the feedback signal y are used to determine the baseband equivalent non-linear behavior of the PA. The signal y can be written as a function of the signal x_p (Direct Learning) or vice versa (Indirect Learning). This is generally described in section 4.2.2.

In the LS predistorter, an estimate of a signal, generally named x_2, is constructed via linear combination of polynomial basisfunctions $\gamma_{k\tau}$, using the least squares criterion. In the Crosscorrelation predistorter however, the signals $\gamma_{k\tau}$ are not combined directly. We first determine the crosscorrelation function of x_2 with a reference signal r and the crosscorrelation functions of $\gamma_{k\tau}$ with r, where r will be specified furtheron. We call these crosscorrelation functions x_2' and $\gamma_{k\tau}'$ respectively. The relation between the time signals $x_2(t)$ and $\gamma_{k\tau}(t)$ is determined by the coefficients $a_{k\tau}$ (expression 4.32). Below we will show that the relation between the crosscorrelation functions x_2' and $\gamma_{k\tau}'$ is also determined by $a_{k\tau}$. The starting point is expression 4.32. We repeat this expression:

$$x_2(t) = \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} a_{k\tau}\gamma_{k\tau}(t) \tag{4.32}$$

The crosscorrelation functions x_2' and $\gamma_{k\tau}'$ are defined as:

$$x_2'(n) = \lim_{L\to\infty} \frac{1}{L} \sum_{u=-\frac{1}{2}L+1}^{\frac{1}{2}L} r^*(u)x_2(u+n) \tag{5.1}$$

$$\gamma_{k\tau}'(n) = \lim_{L\to\infty} \frac{1}{L} \sum_{u=-\frac{1}{2}L+1}^{\frac{1}{2}L} r^*(u)\gamma_{k\tau}(u+n) \tag{5.2}$$

where * indicates the complex conjugate. Replacing x_2 in expression 5.1 with the right part of expression 4.32 leads to

$$x_2'(n) = \lim_{L\to\infty} \frac{1}{L} \sum_{u=-\frac{1}{2}L+1}^{\frac{1}{2}L} r^*(u)\left(\sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} a_{k\tau}\gamma_{k\tau}(u+n) \right) \tag{5.3}$$

By changing the order of the summations, we obtain:

$$\begin{aligned} x_2'(n) &= \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} a_{k\tau}\left(\lim_{L\to\infty} \frac{1}{L} \sum_{u=-\frac{1}{2}L+1}^{\frac{1}{2}L} r^*(u)\gamma_{k\tau}(u+n) \right) \\ &= \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} a_{k\tau}\gamma_{k\tau}'(n) \end{aligned} \tag{5.4}$$

The relation between the time signals $x_2(t)$ and $\gamma_{k\tau}(t)$ is determined by the coefficients $a_{k\tau}$ but also the relation between the crosscorrelation functions $x_2'(n)$ and $\gamma_{k\tau}'(n)$ is determined by $a_{k\tau}$. Later on in the description of the algorithm we also do some additional processing on the crosscorrelation functions. Below we will show the correctness of these processing steps.

If we taper the crosscorrelation functions with a tapering function $h(n)$, the following holds:

$$\begin{aligned} h(n)\cdot x_2'(n) &= h(n)\cdot \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} a_{k\tau}\gamma_{k\tau}'(n) \\ &= \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} a_{k\tau}\big(h(n)\cdot\gamma_{k\tau}'(n)\big) \end{aligned} \tag{5.5}$$

This means that, if we multiply the functions $x_2'(n)$ and $\gamma_{k\tau}'(n)$ with the same function $h(n)$, the relation between the resulting functions is still determined by the coefficients $a_{k\tau}$.

Furthermore, we will apply the DFT on the tapered crosscorrelation functions:

$$\begin{aligned}\mathrm{DFT}\big(h(n)\cdot x_2'(n)\big) &= \mathrm{DFT}\Big(h(n)\cdot\sum_{k=1}^{K}\sum_{\tau=0}^{\tau_{max}-1} a_{k\tau}\gamma_{k\tau}'(n)\Big)\\ &= \sum_{k=1}^{K}\sum_{\tau=0}^{\tau_{max}-1} a_{k\tau}\mathrm{DFT}\big(\big(h(n)\cdot\gamma_{k\tau}'(n)\big)\big) \qquad (5.6)\end{aligned}$$

At the end we multiply the spectra with a function $f(n)$, similar to tapering the crosscorrelation function:

$$\begin{aligned}f(n)\cdot\mathrm{DFT}\big(h(n)\cdot x_2'(n)\big) &= f(n)\cdot\mathrm{DFT}\Big(h(n)\cdot\sum_{k=1}^{K}\sum_{\tau=0}^{\tau_{max}-1} a_{k\tau}\gamma_{k\tau}'(n)\Big)\\ &= \sum_{k=1}^{K}\sum_{\tau=0}^{\tau_{max}-1} a_{k\tau}\cdot\\ &\quad\Big(f(n)\cdot\mathrm{DFT}\big(\big(h(n)\cdot\gamma_{k\tau}'(n)\big)\big)\Big) \qquad (5.7)\end{aligned}$$

The relation between the 'modified' versions of x_2 and $\gamma_{k\tau}$ is still determined by the coefficients $a_{k\tau}$. In a predistorter system, $x_2(t)$ and $\gamma_{k\tau}(t)$ are known and within the LS predistorter, they are directly used to determine the coefficients $a_{k\tau}$ by means of least-square error estimation. The general idea behind the Crosscorrelation predistorter is not to use the 'time' signals $x_2(t)$ and $\gamma_{k\tau}(t)$ but to use the 'modified' versions $f(n)\cdot\mathrm{DFT}\big(h(n)\cdot x_2'(n)\big)$ and $f(n)\cdot\mathrm{DFT}\big(h(n)\cdot\gamma_{k\tau}'(n)\big)$ instead. The advantage is that the processing requirements can be reduced considerably and can be made independent from the number of samples, used to estimate the polynomial coefficients $a_{k\tau}$.

We will now describe the processing steps in more detail. We first have to determine the reference signal $r(t)$ which is crosscorrelated with $x_2(t)$ and $\gamma_{k\tau}(t)$. We chose to crosscorrelate with a clipped version x_Q of the complex signal x_p (see figure 5.2).

The conversion to a clipped signal is defined as:

$$r(t) = x_Q(x_p) = \mathrm{sign}(\mathrm{Re}(x_p)) + i\cdot\mathrm{sign}(\mathrm{Im}(x_p)) \qquad (5.8)$$

where sign() is defined as:

$$\mathrm{sign}(x) = \begin{cases}-1 & x<0\\ 1 & x\geq 0\end{cases} \qquad (5.9)$$

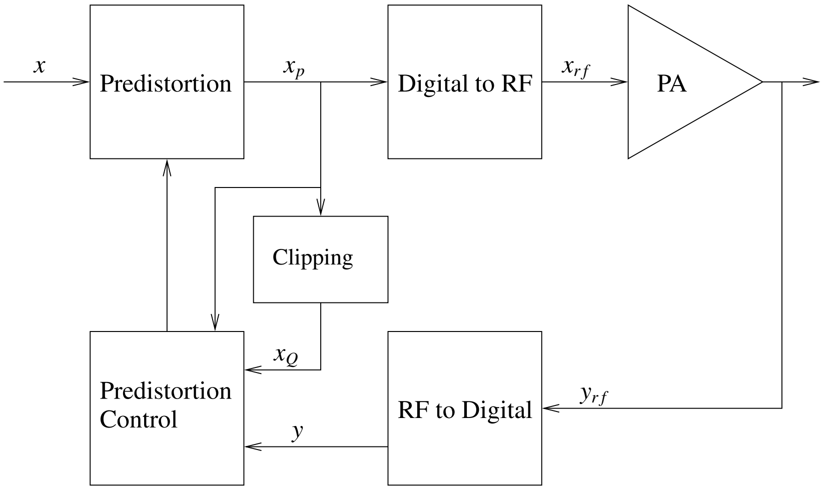


Figure 5.2: Digital Crosscorrelation Predistortion

Using a clipped signal as one of the operands of a crosscorrelation instead of the full precision representation, drastically reduces the complexity. The four real multiplications involved in a complex multiplication become very simple to implement: if the first operand of a real multiplication equals 1, the second operand is left unchanged. If the first operand equals -1, the sign of the second operand is reversed.

In the following, we describe the algorithm that has been implemented in Matlab and used in the simulations.

In figure 5.3, the generation of the crosscorrelation function x_2' is illustrated schematically. The crosscorrelation functions $\gamma_{k\tau}'$ are generated similarly.
The signal x_2 is crosscorrelated with the reference signal r (see expression 5.1). As reference signal we use x_Q ($r = x_Q$). From the complex conjugate x_Q^*, we select $T - N + 1$ samples that are 'in the middle' of the total set of T consecutive samples available to the predistorter. These samples are piecewise multiplied with an equal number of samples of x_2. These samples originate from 'earlier' time-windows than the time-window from which the samples for x_Q are selected ($n < 0$), from a time-window that exactly matches the time-window for x_Q ($n = 0$) and from 'later' time-windows ($n > 0$). This is indicated in the upper part of figure 5.3. After piecewise multiplication, the products are summed, leading to a single value for every shifted time-window of x_2. Piecewise multiplication and summation are realized by the matrix-vector multiplication as indicated in the lower part of figure 5.3. Every resulting value is based on an equal number of products: ($T - N + 1$). Since in practice $T >> N$, the number of products on which the resulting values are based, approximately equals the Integration Length T. An algebraic description of crosscorrelation is given below.

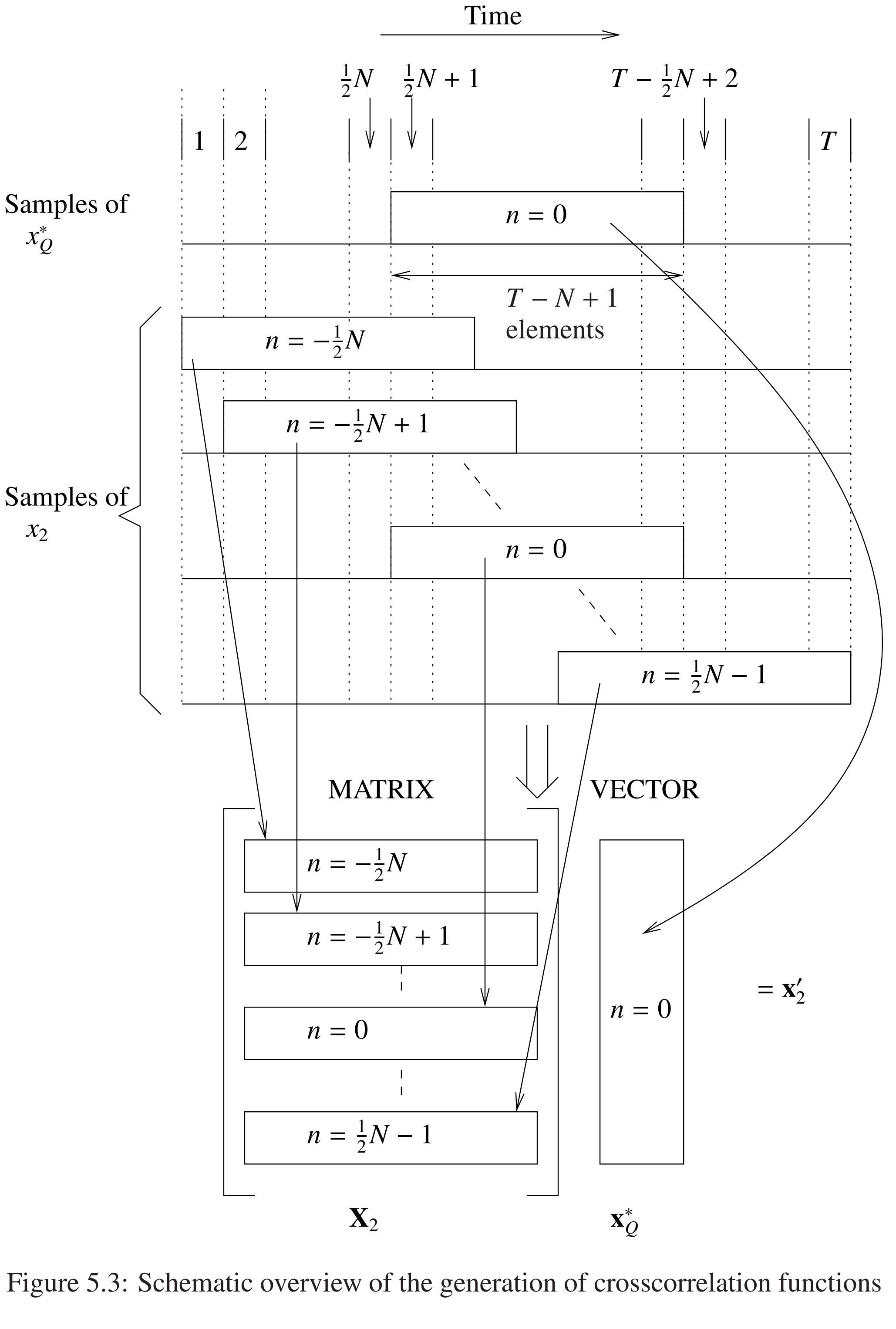


Figure 5.3: Schematic overview of the generation of crosscorrelation functions

The samples, selected according to the time windows given above, are stacked into vectors and the vectors for x_2 and $\gamma_{k\tau}$ are stacked into matrices:

$$\mathbf{x}_Q = \left[x_Q(x_p(\frac{1}{2}N+1)), ..., x_Q(x_p(T-\frac{1}{2}N+1)) \right]^T \tag{5.10}$$

We define the matrix $\mathbf{X}_2$ as:

$$\begin{aligned} \mathbf{X}_2 = & \Big[[x_2(1), ..., x_2(T-N+1)]^T, ..., \\ & [x_2(N), ..., x_2(T)]^T \Big]^T \end{aligned} \tag{5.11}$$

Similarly, we define $\mathbf{\Gamma}_{k\tau}$ for $k = 1, ..., K$ and $\tau = 0, ..., \tau_{max} - 1$:

$$\begin{aligned} \mathbf{\Gamma}_{k\tau} = & \Big[[\gamma_{k\tau}(x_1(1)), ..., \gamma_{k\tau}(x_1(T-N+1))]^T, ..., \\ & [\gamma_{k\tau}(x_1(N)), ..., \gamma_{k\tau}(x_1(T))]^T \Big]^T \end{aligned} \tag{5.12}$$

The crosscorrelations are then easily described by means of matrix-vector products:

$$\boldsymbol{\gamma}'_{k\tau} = \mathbf{\Gamma}_{k\tau}\mathbf{x}_Q^* \tag{5.13}$$

$$\mathbf{x}'_2 = \mathbf{X}_2\mathbf{x}_Q^* \tag{5.14}$$

The crosscorrelation vectors $\boldsymbol{\gamma}'_{k\tau}$ and $\mathbf{x}'_2$ are tapered with a tapering function to reduce the effects of the limited number of values of the correlation function (in the ideal case N is infinite). There are several options (Hamming-, Hanning-, Blackman-tapers) and simulations have shown that the performance of predistortion is not influenced significantly by the choice of the tapering functions. We use the Hanning taper:

$$h(j) = \frac{1}{2}(1 - \cos(2\pi(j-1)/(N-1))) \tag{5.15}$$

$$\mathbf{H} = \mathrm{diag}(h(1), ..., h(N)) \tag{5.16}$$

where 'diag($h(1), ..., h(N)$)' indicates a diagonal $N \times N$ matrix where $\mathbf{H}_{ii} = h(i)$. A Discrete Fourier Transform is applied to the tapered crosscorrelation vectors:

$$\mathbf{f}_{k\tau} = \mathbf{W}\mathbf{H}\boldsymbol{\gamma}'_{k\tau} \tag{5.17}$$

$$\mathbf{f} = \mathbf{W}\mathbf{H}\mathbf{x}'_2 \tag{5.18}$$

where $\mathbf{W}$ equals the DFT kernel. The elements w_{pq} of the kernel are defined as:

$$w_{pq} = e^{-i2\pi\frac{p-1}{N}(q-1)} \tag{5.19}$$

If the vectors (spectra) $\mathbf{f}_{k\tau}$ are concatenated into a matrix $\mathbf{F} = [\mathbf{f}_{10}, ..., \mathbf{f}_{K0}, ..., \mathbf{f}_{1\tau_{max}-1}, ..., \mathbf{f}_{K\tau_{max}-1}]$ and the vector $\mathbf{a}$ is defined as $\mathbf{a} = [a_{10}, ..., a_{K0}, ..., a_{1\tau_{max}-1}, ..., a_{K\tau_{max}-1}]^T$, the memory polynomial predistorter is described in the frequency domain as:

$$\mathbf{f} = \mathbf{F}\mathbf{a} \tag{5.20}$$

The least squares solution minimizes the *absolute* error over the frequency domain. Because the spectra have a relatively low power spectral density in the adjacent channels, the absolute error in the adjacent channels might be small, but the *relative* errors can be (too) large. For that reason, we want to minimize the relative error. This is realized by normalizing the spectra with $\mathbf{f}$:

$$g_{k\tau}(n) = \frac{f_{k\tau}(n)}{f(n)} \tag{5.21}$$

$$\mathbf{g}_{k\tau} = [g_{k\tau}(1), ..., g_{k\tau}(N)]^T \tag{5.22}$$

$$\mathbf{G} = [\mathbf{g}_{10}, ..., \mathbf{g}_{K0}, ..., \mathbf{g}_{1\tau_{max}-1}, ..., \mathbf{g}_{K\tau_{max}-1}] \tag{5.23}$$

The N element vector $\mathbf{g}$ is defined via:

$$g(n) = \frac{f(n)}{f(n)} = 1, \quad n = 1, ..., N \tag{5.24}$$

The normalized version of expression (5.20) becomes:

$$\mathbf{g} = \mathbf{G}\mathbf{a} \tag{5.25}$$

The least-squares solution $\hat{\mathbf{a}}$ equals:

$$\hat{\mathbf{a}} = (\mathbf{G}^H\mathbf{G})^{-1}\mathbf{G}^H\mathbf{g} \tag{5.26}$$

To reduce the effects of noise on the correlation functions $\mathbf{g}_{k\tau}$, we select only those vector elements which represent the power in the main channel and the first adjacent channels leading to vectors $\mathbf{g}$ and $\mathbf{g}_{k\tau}$ with limited length.

5.1.3 Implementation aspects

The complexity of the LS predistorter is at least $O(T)$; the number of complex multiplications scales linearly with the number of samples used to determine the polynomial coefficients $\hat{\mathbf{a}}$. In the Crosscorrelation predistorter, the size of the vectors $\mathbf{f}$ and $\mathbf{f}_{k\tau}$ is reduced to N (instead of T). In general, N is much smaller than T. If an FFT is used to transform the vectors from the time domain into the frequency domain, the complexity is $O(N^2 \log N)$. The reduction of the length of the vectors is due to the crosscorrelation. This crosscorrelation is easy to implement. Due to the clipping of x_p, no full-precision complex multiplications are required so the overall complexity of the Crosscorrelation predistorter is determined by the FFT.

To determine the complexity in more detail, we look more carefully at the different steps of the algorithm after the crosscorrelation: applying the Hanning taper, applying the DFT, normalizing the spectra and calculating the least-squares solution. We will deal with these steps separately.

Applying the Hanning taper is described functionally in expressions 5.17 and 5.18. In these expressions, the Hanning taper values are placed at the diagonal of a matrix and this matrix is multiplied with the different crosscorrelation functions. In principle, N^2 multiplications are involved. However, this can be implemented more efficiently by a piece-wise multiplication of the crosscorrelation values with the Hanning taper values, involving N multiplications. The Hanning taper is applied to $\mathbf{x}'_2$ and $\gamma'_{k\tau}$. This leads to $N \cdot (K\tau_{max} + 1)$ multiplications. Note that the Hanning taper values are real values, while the crosscorrelation values are complex values.

The DFT can by implemented by a Fast Fourier Transform (FFT) in case $N = 2^p$, where $p \in \mathbb{N}$. The number of complex multiplications involved in this case are: $N^2 \log N$. In total $K\tau_{max} + 1$ spectra have to be calculated leading to approximately $N^2 \log N \cdot (K\tau_{max} + 1)$ complex multiplications.

Normalizing the spectra is realized by multiplication of the spectra with the element-wise reciprocals of a vector. First the reciprocals of the complex elements of a vector of length N have to be determined. The calculation of the reciprocal of a complex value involves four real multiplications and the calculation of the reciprocal of a real value ($\frac{1}{a+i \cdot b} = \frac{a-i \cdot b}{a^2+b^2}$). The reciprocals have to be calculated once, leading to $4N$ real multiplications in addition to the calculation of the reciprocal of a real. Multiplying the spectra with the element-wise reciprocals of a vector results in $N \cdot K\tau_{max}$ complex multiplications.

Concerning the calculation of the least-squares solution, the complexity is basically determined by the following two operations: $\mathbf{G}^H\mathbf{G}$ and $\mathbf{G}^H\mathbf{g}$. This is valid in case $K\tau_{max}$ is small compared to N, which is, in our applications, generally the case. In our simulations (see section 5.3.1), N equals 64 while using only odd polynomials up to the fifth degree ($K = 5$ and $k = 1, 3, 5$) and $\tau_{max} = 2$ which implies that '$K\tau_{max}$' represents the value 6. The number of complex multiplications within $\mathbf{G}^H\mathbf{G}$ equals $(K\tau_{max})^2 \cdot N$ and within $\mathbf{G}^H\mathbf{g}$, the number of complex multiplications equals $K\tau_{max} \cdot N$. The matrix inversion $(\mathbf{G}^H\mathbf{G})^{-1}$ requires $(K\tau_{max})^3$ complex multiplications (see [73], pg. 48). The total number of complex multiplications equals therefore $K\tau_{max}(K\tau_{max} + 1) \cdot N + (K\tau_{max})^3 \approx K\tau_{max}(K\tau_{max} + 1) \cdot N$. The complexity of the LS predistorter approximately equals $K\tau_{max}(K\tau_{max} + 1) \cdot T$ (see section 4.2.3). The expression for the complexity of the LS predistorter resembles the expression of the 'least-squares solution' part of the Crosscorrelation predistorter to a large extent; the factor T in the expression for the LS predistorter is replaced by a factor N for the Crosscorrelation predistorter. In a practical scenario, T equals 8k and N equals 64. So for this scenario, we have reduced the complexity of the calculation of the least squares solution with a factor 128. Note that this reduction is not valid for the crosscorrelation as a whole because of the additional tapering, FFT and normalization. In appendix C, we compare the LS- and Crosscorrelation predistorter for a typical scenario when implemented on a specific processor and the results are discussed further in section 5.3.4.

5.2 Mapping

Compared to the mapping of the first iteration of the Analog-Digital Codesign of a digital predistorter for PAs, the architectures are equal but the application has changed. Basically we have four additional stages: the crosscorrelation stage, applying the Hanning taper, the DFT (or FFT) and the normalization of spectra. Furthermore, the algorithm which determines the least squares solution has changed.

All processing steps mentioned above are mapped onto the Digital Processing part of the architecture specification (see figure 4.1). Based on this, the mapping

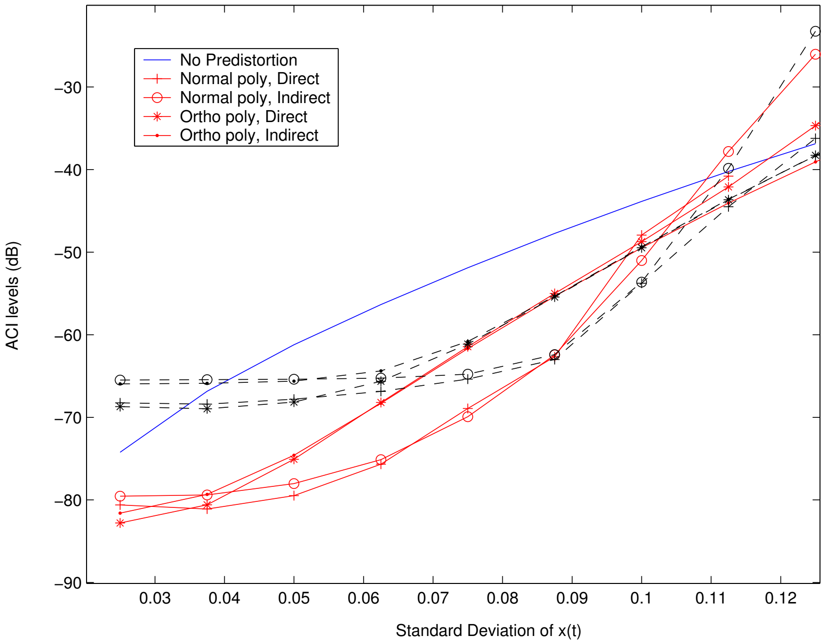


Figure 5.4: Performance Crosscorrelation predistorter

of the application onto the architectures (with different resolutions) is similar to the mapping in the first iteration.

5.3 Performance analysis

5.3.1 Performance in case of high resolution quantization

The simulator, used for the evaluation of the LS predistorter, has been adapted to simulate the Crosscorrelation predistorter. The clipping function has been added and the Predistortion Control block (see figure 5.2) has been extended with the algorithm described above. For the simulations, we used a 64 point FFT to realize the DFT ($N = 64$). For the PA model, we used the Wiener-Hammerstein PA model described in section 3.5. We used the same settings as for the LS predistorter (see Table 4.1). The performance for the different schemes, as a function of the standard deviation at the input of the predistorter, is given in figure 5.4 and the compression is given in figure 5.5. In figure 5.4, the dashed curves show the results obtained for the LS predistorter (see figure 4.9).

For standard deviations smaller than roughly 0.09, the Crosscorrelation predistorter has a better performance (lower ACI levels) than the LS predistorter. However, the same effects as for the LS predistorter are visible: the performance

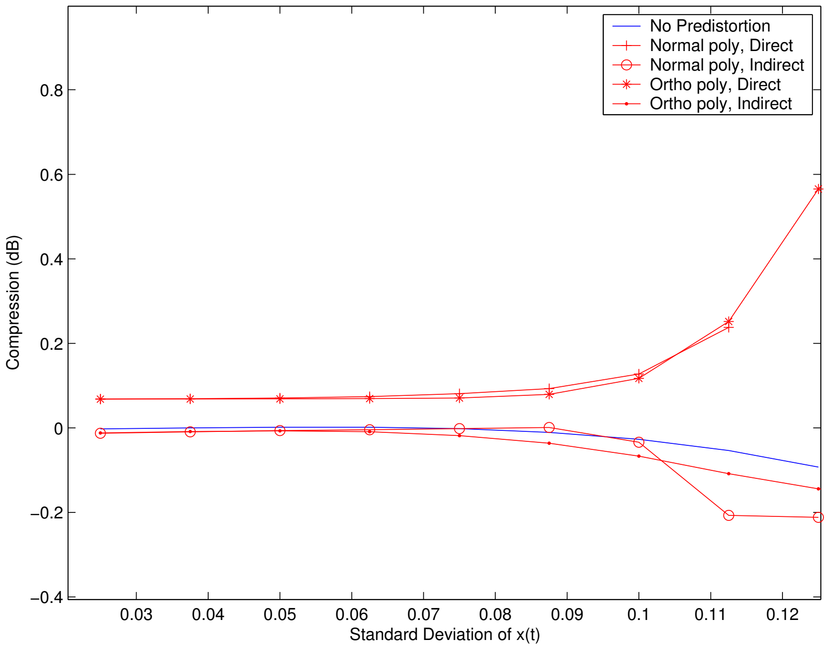


Figure 5.5: Compression Crosscorrelation predistorter

when using normal polynomials is higher than when using orthogonal polynomials. We already indicated that this is mainly due to the PA model that we used. In this model, the non-linearity was introduced by normal polynomials. Because of this, the normal polynomials used in the Predistortion Control block, better fit the PA model. For standard deviations larger than 0.1, the performance of the Crosscorrelation predistorter is comparable with the performance of the LS predistorter.

The most striking aspect however is the higher performance of the Crosscorrelation predistorter in relation to the performance of the LS predistorter for standard deviations smaller than roughly 0.09. This higher performance is due to the fact that the Crosscorrelation predistorter minimizes the error (using the least-squares criterion) in the frequency domain. We normalize the spectra to have a constant relative error in the frequency domain. This means that we practically only use the calculated power in the adjacent channels to determine the polynomial coefficients $a_{k\tau}$. The result is that mainly the power in the adjacent channels is reduced when applying the polynomial coefficients $a_{k\tau}$ within the Predistortion block. This approach contrasts with the approach within the LS predistorter. Within the LS predistorter, the polynomial coefficients $a_{k\tau}$ are determined in such way that the squared-error of the time-samples is minimized. In many cases, this solution will lead to a suppression of the power in the adjacent channels as well, but this is not guaranteed.

For larger input standard deviations, the distortion becomes so severe that the LS- and Crosscorrelation predistorters give similar performance.

For the Direct Learning scheme, the compression (see figure 5.5) is less than the compression of the PA model without predistortion. The Indirect Learning scheme leads to an increase of the compression for standard deviations larger than 0.09.

5.3.2 Effects of quantization

The design of the Crosscorrelation predistorter only changes the Predistortion Control block of the system. The feedback path of the predistortion system is not changed compared to the feedback path of the LS predistorter. We therefore only repeat the expression for the SNR of the dithered and quantized feedback signal (expression 4.45):

$$\mathrm{SNR} = 6.02b + 4.77 - 20\log k - 3.01d \tag{4.45}$$

where b indicates the number of bits, k indicates the amplitude peak-to-average ratio of the input signal and d the range, in LSB, of the uniformly distributed dither noise.

5.3.3 Performance in case of coarse quantization

For the Crosscorrelation predistorter, we also simulated the effects of quantization in the feedback path of the predistorter. The same settings as for the analysis of the effects of quantization on the LS predistorter are used; the standard deviation at the input equals 0.0875. In figure 5.6, the performance of the Crosscorrelation predistorter, as a function of the resolution of the ADC, is given. The dashed curves show results of the LS predistorter.

The Crosscorrelation predistorter, in case the Direct Learning scheme in combination with normal polynomials is used, is only stable if more than 8 bits are used for the quantization. For the three other schemes, the algorithm is stable if more than 3 bits are used. When using the Direct Learning scheme with orthogonal polynomials, the Crosscorrelation predistorter is stable even for single-bit quantization.

In comparison with the LS predistorter, the Crosscorrelation predistorter is more sensitive to quantization errors. This is because, within the Crosscorrelation predistorter, we minimize the relative error in the frequency domain. As mentioned in section 5.3.1, we practically only use the power in the adjacent channels to calculate the polynomial coefficients $a_{k\tau}$. As will be shown below, coarse quantization introduces distortion which results in significant power at frequencies

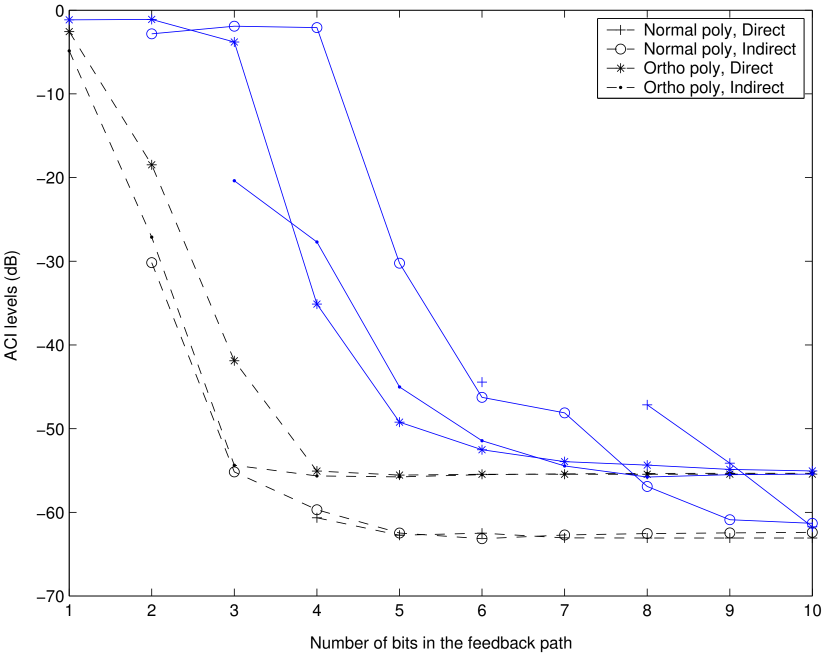


Figure 5.6: Quantization effects on the Crosscorrelation predistorter

other than the original frequencies. So, (distortion-)power, due to quantization is added to the adjacent channels. Because we optimize on the relative power in the adjacent channels, already a small amount of quantization power in the adjacent channels can have significant effects on the performance of the Crosscorrelation predistorter. This is in contrast with the LS predistorter. The LS predistorter determines the polynomial coefficients based on the least-squared error of the time-samples. So, what results in improved performance in case of high-resolution quantization in the feedback path, results in a decreased performance in case of coarse quantization in the feedback path. To understand the effects of quantization, a detailed analysis of coarse quantization will be given below.

5.3.3.1 Effects of quantization on the measured correlation function

From the results presented in figure 5.6, we see that the Crosscorrelation predistorter is sensitive to quantization effects. Already for less than 8 bits, the Crosscorrelation predistorter using normal polynomials and Direct learning, becomes unstable. Obviously, the quantization in the feedback path has more influence than the non-linearity of the PA.

In this section, we will first develop a model to analyse the effects of coarse quantization in the feedback path. Second, we will use the model to quantify

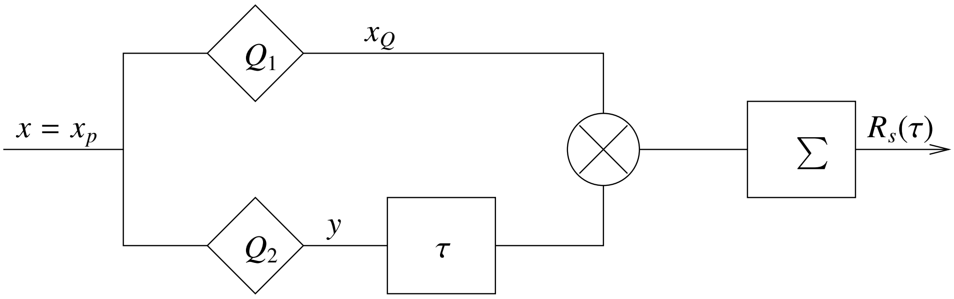


Figure 5.7: Model for analyzing quantization effects

the effects on the measured correlation function and third, dithering is included. For the analysis of the quantization effects we assume that the effects of the non-linearity of the PA can be neglected compared to the effects of coarse quantization. The consequence of this assumption is that the input of the PA model equals the output of the PA model, which on its turn equals the original signal. Thus, $x_p = x$. x is a complex signal where both the real and imaginary part are modeled as seperate Gaussian noise processes with equal power and equal bandwidth. The real and imaginary parts are thus statistically independent. It therefore suffices to separately analyze the effects on the real part of the signal and the imaginary part of the signal. Because both parts are considered to be equal Gaussian processes, we only need to analyze either the real or the imaginary part. For the analysis we assume that x is a real signal with $\sigma_x = 1$. Figure 5.7 represents the model used. Q_1 represents the clipping function converting x_p into x_Q (see figure 5.2) and Q_2 represents the ADC within the feedback path. First, we want to develop a general model without using the knowledge that Q_1 represents a clipping function. Thus, quantizers Q_1 and Q_2 are generally defined as:

$$Q_1(x) = \left(\sum_{i=-2^{b_1-1}+1}^{2^{b_1-1}-1} \Phi(g_1 x - i) \right) - 2^{b_1-1} + \frac{1}{2} \tag{5.27}$$

$$Q_2(x) = \left(\sum_{i=-2^{b_2-1}+1}^{2^{b_2-1}-1} \Phi(g_2 x - i) \right) - 2^{b_2-1} + \frac{1}{2} \tag{5.28}$$

Φ is Heaviside's step function (see expression 3.6). The relation between the amplitude span of the input signal and the input range of the quantizers Q_1 and Q_2 is determined by g_1 and g_2. We first determine g_1 and g_2 given the amplitude span of the input signal, the number of bits b and the required Headroom factor.

To realize the single-bit quantization, g_1 is chosen much smaller than one. Suppose the standard deviation of x equals 1 and the amplitude peak-to-average ratio is smaller than 4. Then, for $g_1 = \frac{1}{4}$, Q_1 effectively is a single-bit quantizer. For Q_2, we also assume that the standard deviation of the input signal x equals 1 and that the ratio between the maximum input amplitude and the standard deviation of the input signal does not exceed 4. The standard deviation of the signal after multiplication with the granularity factor g_2 equals $1 \cdot g_2 = g_2$ and the maximum amplitude $4 \cdot g_2 = 2^2 g_2$. The total span of the signal equals $2 \cdot 2^2 g_2 = 2^3 g_2$. According to expressions 5.27 and 5.28, the distance between the reference levels of the quantizers equals 1 and the output of the quantizer can be coded using $b_2 =^2 \log(2^3 g_2)$ bits. The granularity factor g_2 can then be written as a function of the number of bits via: $g_2 = 2^{b_2 - 3}$.

In appendix D, section D.3.2, we show that the measured correlation function R_s can be expressed as a function of the normalized autocorrelation function $\rho(\tau)$ of x. We repeat the result:

$$R_s(\rho(\tau)) = (1 + \epsilon_1) g_1 g_2 \rho(\tau) + \frac{1}{2}\epsilon_2 \tag{D.28}$$

where

$$\begin{aligned}
\epsilon_1 &= \sum_{m=1}^{\infty} 2(e^{-2\pi^2 m^2 g_1^2} + e^{-2\pi^2 m^2 g_2^2}) \\
\epsilon_2 &= \sum_{m_1=1}^{\infty} \sum_{m_2=1}^{\infty} \frac{2}{m_1 m_2 \pi^2} e^{-2\pi^2 (m_1^2 g_1^2 + m_2^2 g_2^2)}. \\
&\quad \sinh(4\pi^2 g_1 g_2 m_1 m_2 \rho(\tau))
\end{aligned}$$

From expression D.28, two effects can be deduced. The measured correlation function R_s is a scaled version of the original correlation function ρ plus a distortion component determined by ϵ_2 which introduces spurious components or spectral distortion. Note that the first term, $(1 + \epsilon_1) g_1 g_2 \rho(\tau)$, does not introduce spurious components.

To obtain the normalized correlation, the measured correlation (R_s) has to be divided by the product of the standard deviations at the outputs of the quantizers Q_1 and Q_2:

$$\rho_s(\rho) = \frac{R_s(\rho)}{\sqrt{P_{Q_1}}\sqrt{P_{Q_2}}} \tag{5.29}$$

where P_{Q_i} represents the power at the output of quantizer Q_i, for $i = 1, 2$. For the single-bit quantizer Q_1, the output power is easily determined. The output equals either $\frac{1}{2}$ or $-\frac{1}{2}$, so $P_{Q_1} = \frac{1}{4}$. For Q_2, the power of the output signal equals the autocorrelation of that signal for $\tau = 0$. The model in figure 5.7 can be used to determine the power at the output of quantizer Q_2 where $\tau = 0$ and Q_1 is replaced by Q_2. In this case, the output $R_s(0)$ of the model represents the power of the output Q_2. This power is then calculated using expression D.28 where $\tau = 0$ and g_1 is replaced by g_2. Because ρ represents the normalized autocorrelation, $\rho(0)$ equals 1. This results in:

$$\begin{aligned} P_{Q_2} =& (1+\epsilon_1)g_2^2 + \frac{1}{2}\epsilon_2 \\ \epsilon_1 =& \sum_{m=1}^{\infty} 4(e^{-2\pi^2 m^2 g_2^2}) \\ \epsilon_2 =& \sum_{m_1=1}^{\infty}\sum_{m_2=1}^{\infty} \frac{2}{m_1 m_2 \pi^2} e^{-2\pi^2 g_2^2 (m_1^2+m_2^2)}. \\ & \sinh(4\pi^2 g_2^2 m_1 m_2) \end{aligned} \tag{5.30}$$

With expression 5.29 and the calculated values for P_{Q_1} and P_{Q_2}, we can analyze the effects of the ADCs on the measured correlation function quantitatively. In the Crosscorrelation predistorter, Q_1 is a single bit quantizer. If Q_2 is a single-bit quantizer as well, the normalized correlation coefficient is known (see [15]):

$$\rho_s(\rho) = \frac{2}{\pi} \arcsin(\rho) \tag{5.31}$$

Since this result is known, we will calculate the correlation for three other resolutions of Q_2: 2, 3 and 4 bits, using expression 5.29. This correlation is presented, as a function of the input correlation ρ, in figure 5.8, for ρ close to 1.

Three effects are clearly visible. First, because of the single-bit AD conversion by Q_1, a decorrelation ($\rho_s < \rho$) is introduced reducing the measured correlation coefficient for all input correlation coefficients. Second, for input correlations far from one (say $\rho < 0.86$), the relation between ρ_s and ρ is almost linear where a smaller number of bits implies a decreased correlation due to the increased quantization noise introduced by Q_2. Third, for correlations close to 1 (> 0.86), the relation between ρ_s and ρ is non-linear. The non-linearity becomes more severe for decreasing number of bits of the Q_2. The increase in correlation is stronger for a low number of bits. This is because, for a decreasing resolution of Q_2, the quantization noise produced by Q_2 becomes more and more correlated with the quantization noise produced by Q_1. In the ultimate case, Q_2 also is a single bit quantizer and the correlation between the outputs of Q_1 and Q_2 will be 1 for $\rho = 1$. This can also be seen from expression 5.31.

The basic idea behind the use of a dithering signal as introduced in section 4.4.2, is to reduce the correlation ρ of the signals at the input of the quantizers and

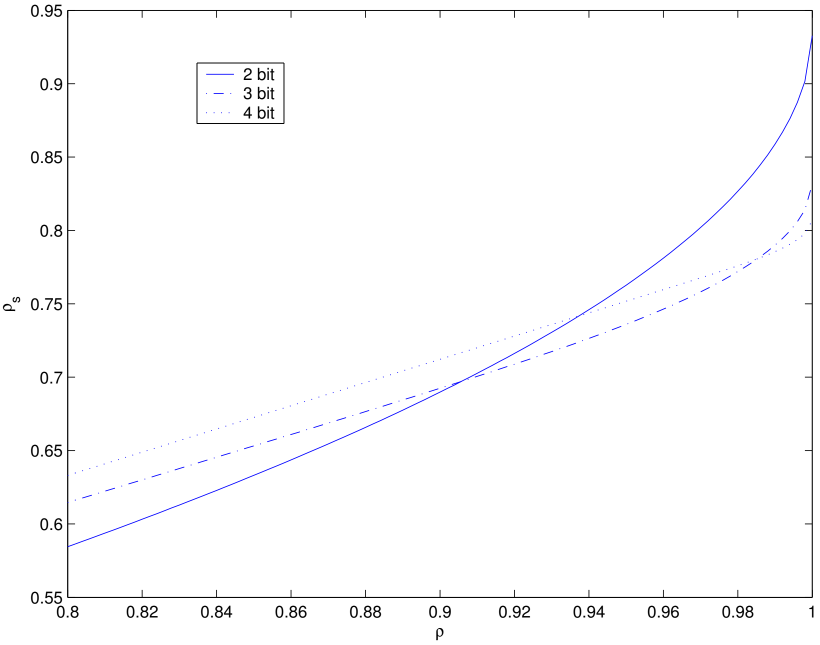


Figure 5.8: Measured correlation versus real correlation

therefore use more linear portions of the relations in figure 5.8. The model for analyzing quantization effects as shown in figure 5.7, is extended in figure 5.9.

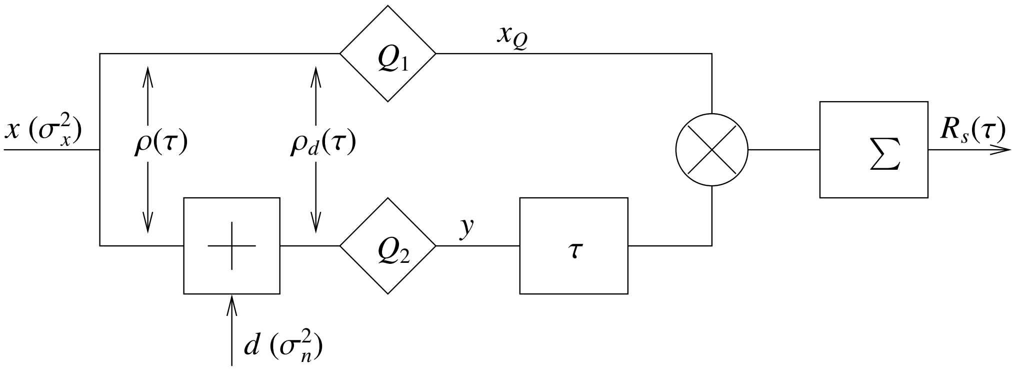


Figure 5.9: Model for analyzing quantization effects including dither

A dither signal d, with power σ_n^2 is added to the input signal x which has power σ_x^2. As dithering signal, we use noise with a uniform distribution which has no correlation with input signal x. This signal is added before quantization at Q_2. The normalized autocorrelation function of x equals $\rho(\tau)$. The normalized crosscorrelation function of x and $x + d$ equals $\rho_d(\tau)$. This crosscorrelation function is given as:

$$\rho_d(\tau) = \frac{\sigma_x}{\sqrt{\sigma_x^2 + \sigma_n^2}} \rho(\tau) \tag{5.32}$$

By adding the dithering signal, the input correlation is reduced and the non-linear errors on the measured correlation function are reduced.

Now we know the systematic error on the normalized correlation function, introduced by quantization, and we know that this error can be reduced by dithering, we have to determine the effect on the SNR of the measured correlation function. The dithered and quantized signal y is crosscorrelated with x_Q. The correlation r_Q becomes:

$$\begin{aligned} r_Q = \mathrm{E}(y \cdot x_Q^*) = \mathrm{E}((x + n_Q) \cdot x_Q^*) = \\ \mathrm{E}(x \cdot x_Q^*) + \mathrm{E}(n_Q \cdot x_Q^*) \end{aligned} \tag{5.33}$$

where n_Q indicates the quantization noise. If the power of the dithering signal is sufficient, the non-linear errors on the measured correlation function are removed and consequently, the quantization noise n_Q and x_Q are uncorrelated. This implies that r_Q is an unbiased estimate of the correlation function $r = \mathrm{E}(x \cdot x_Q^*)$. The variance on the correlation function r_Q equals:

$$\begin{aligned} &\mathrm{E}\left[\left(y \cdot x_Q^*\right)^2 - \left(\mathrm{E}(y \cdot x_Q^*)\right)^2\right] = \mathrm{E}\left[\left(y \cdot x_Q^*\right)^2 - \left(\mathrm{E}(x \cdot x_Q^*)\right)^2\right] = \\ &\mathrm{E}\left[\left((x + n_Q) \cdot x_Q^*\right)^2 - \left(\mathrm{E}(x \cdot x_Q^*)\right)^2\right] = \\ &\mathrm{E}\left[\left((x \cdot x_Q^*) + (n_Q \cdot x_Q^*)\right)^2 - \left(\mathrm{E}(x \cdot x_Q^*)\right)^2\right] = \\ &\mathrm{E}\left[(x \cdot x_Q^*)^2\right] + \mathrm{E}\left[(n_Q \cdot x_Q^*)^2\right] - \left(\mathrm{E}(x \cdot x_Q^*)\right)^2 = \\ &\mathrm{E}(n_Q \cdot x_Q^*)^2 \end{aligned} \tag{5.34}$$

Since x_Q equals either 1 or -1, the variance equals $\sigma_{n_Q}^2$. The result is that the SNR of the peak of the correlation function equals the SNR of the signal after dithering and quantization. This SNR was given by expression 4.45, which is repeated here:

$$\mathrm{SNR} = 6.02b + 4.77 - 20 \log k - 3.01 \times d \tag{4.45}$$

By integration of the signal after multiplication, the SNR can be increased. Doubling the number of samples to be integrated (doubling T) increases the SNR

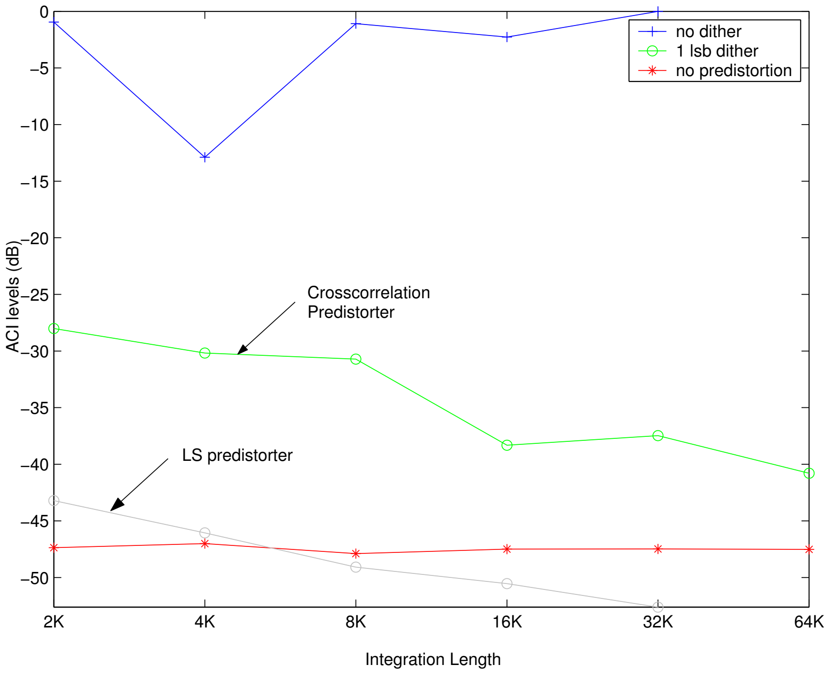


Figure 5.10: Performance Crosscorrelation predistorter with 2-bit quantization, Direct Learning and orthogonal polynomials

with 3.01 dB. The overall formula for the SNR on the crosscorrelation function including dithering, quantization and integration (SNR_{dqi}) then becomes:

$$\begin{aligned}\mathrm{SNR}_{dqi} =& 6.02b + 4.77 - 20\log(k)\\ &- 3.01 \times (d - {}^{2}\log(T))\end{aligned} \tag{5.35}$$

5.3.3.2 Performance of the Crosscorrelation predistorter

As for the LS predistorter, we simulated the effect of coarse quantization on the performance of the Crosscorrelation predistorter. We used the Direct Learning scheme with orthogonal polynomials since this scheme was also used to analyze the effects of quantization on the LS predistorter. Figure 5.10 gives the performance as a function of the length T of the vectors (Integration Length), with and without a dither signal in case of 2-bit quantization in the feedback path.

For the case a dither signal is used, the suppression obtained by the LS predistorter is shown is well. For the Crosscorrelation predistorter, we see the same effects as for the LS predistorter. In case no dither signal is added, the effect of coarse quantization is that the Crosscorrelation predistorter is increasing the ACI

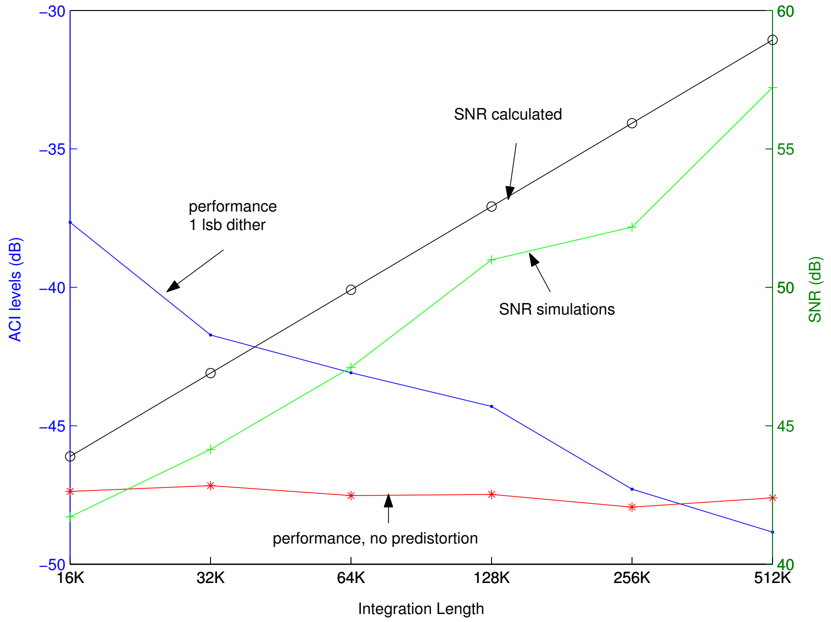


Figure 5.11: Performance and SNR of the Crosscorrelation predistorter with 2-bit quantization, Direct Learning and orthogonal polynomials

levels instead of reducing them. The effects of coarse quantization can be reduced by adding a dither signal and using longer Integration Lengths. For the same Integration Length, the Crosscorrelation predistorter results in higher ACI levels (less performance) than the LS predistorter.

In expression 5.35, the SNR of the correlation function including dithering, quantization and integration (SNR_{dqi}), depends on the Integration Length T. So, for a certain Integration Length, we can determine the performance of the Crosscorrelation predistorter, but we can also calculate the theoretical SNR according to expression 5.35 and determine the actual SNR in the simulations. The four values are given for different Integration Lengths in figure 5.11.

From this figure, we see that the Crosscorrelation predistorter using coarse quantization, reduces the ACI levels below the levels in case no predistortion is used for an Integration Length of 512k samples.

The SNRs of the correlation functions, which resulted in the performance figures presented in figure 5.11, are plotted together with the SNR values, calculated using expression 5.35. We see that both the calculated and simulated SNR increase for increasing Integration Length. There is a more or less constant difference between the calculated and simulated SNR. The expression for the calculation of the SNR assumes uniformly distributed quantization noise. However, in case of two-bit quantization, this assumption is not valid. In combination with a

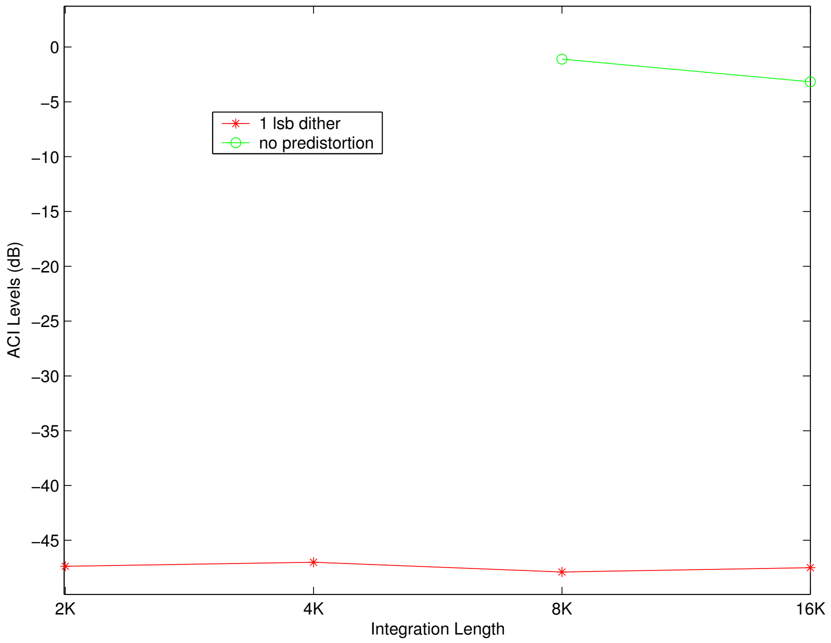


Figure 5.12: Performance Crosscorrelation predistorter with 2-bit quantization, Indirect Learning and orthogonal polynomials

mid-riser quantizer, this leads to lower SNRs. In case of 2-bit quantization and 1 LSB uniform dither, the expected SNR will approximately be 2 dB lower than the ideal SNR. Taking this into consideration, one can deduct from figure 5.11 that the simulated SNR corresponds reasonably well with the expected SNR. More important however, figure 5.11 illustrates the noise reduction effects of longer Integration Lengths.

Furthermore, note that the ACI levels for Integration Lengths of 16k, 32k and 64k samples differ slightly from the results found in figure 5.10 (-38.3 dB, -37.5 dB and -40.8 dB resp. in figure 5.10 and -37.7 dB, -41.7 dB and -43.1 dB resp. in figure 5.11). This is caused by the the random generation of the phases of the signal source (see section 3.3) and the randomness of the dither noise. In section 4.4.1 we have determined, via simulations, that the standard deviation on the estimates of the ACI levels is 3 dB. The differences for Integration Lengths of 16k, 32k and 64k, shown in figures 5.10 and 5.11 are not larger than what is to be expected.

For the sake of completeness, the performance of the Crosscorrelation predistorter as a function of the Integration Length, using the Indirect Learning scheme, is given in figure 5.12.

The Crosscorrelation predistorter does not stabilize in case of quantization and the absence of a dither signal. The reason that the Direct Learning scheme stabilizes and the Indirect Learning scheme does not is that a memory polynomial is a better approximation of the non-linear behavior from input to output than from output to input. The relation from input to output is described by the PA model which is a Wiener-Hammerstein model (filter-polynomial-filter) which resembles a memory polynomial. So, as indicated before, which scheme is most suitable depends on the PA model that is used. In practice this means that, which scheme is most suitable depends on the actual PA that is used. So, before choosing between the Direct- or Indirect Learning scheme, some measurements have to reveal the dominating effects. In case a dither signal is used, the predistorter stabilizes for Integration Lengths longer that 8k but the performance does not increase for increasing Integration Length. This behavior resembles the behavior of the LS predistorter in case of coarse quantization using the Indirect Learning scheme (see figure 4.14).

5.3.4 Overall system timing

In this section, we analyze the system timing of the Crosscorrelation predistorter. The application is the same as the application for the system timing analysis of the LS predistorter (section 4.4.4); a UMTS application with a 49 MHz sampling rate. We assume sufficient resolution in the feedback path. We use 8k samples to determine the non-linearity of the PA. The tasks that have to be executed in the predistortion control block are:

- Generating the powers $\gamma_{k\tau}$ (see section 4.2.2.4). We use a memory depth of 2 ($\tau = 0, 1$) and only odd polynomials up to the fifth degree ($k = 1, 3, 5$). Therefore, to every sample, six values $\gamma_{k\tau}$ are related. As for the LS predistorter, we expect this function to be implemented by means of a look-up table realized in Random Acces Memory (RAM) or Read Only Memory (ROM). The system timing will not be limited by these look-up operations.

- Crosscorrelating the powers $\gamma_{k\tau}$ and the signal x_2 with the clipped reference signal (expressions 5.13 and 5.14). Since the signals $\gamma_{k\tau}$ and x_2 are crosscorrelated with a clipped signal, this function is most efficiently implemented by means of standard up/down adders, available as discrete components or easily implemented in bit-level programmable devices. It is not efficient to use a microprocessor for the execution of these crosscorrelations because the use of a clipped signal can not be exploited. Considering the sampling rate for this specific application (49 MHz), the crosscorrelation operation will not limit system timing.

- The remaining tasks are: applying the Hanning taper (expressions 5.17 and 5.18), applying the Fast Fourier Transform (also expressions 5.17 and 5.18), normalizing the spectra (expressions 5.21 to 5.24) and calculating the least squares solution $\hat{\mathbf{a}}$ (expression 5.26). These tasks are analyzed when executed on a Texas Instrument C67x Floating Point DSP in appendix C.

The system timing is determined by the time it takes the DSP to calculate the PA non-linearity. The result given in appendix C is that a 300 MHz Texas Instruments C67x DSP requires 48 μs for the calculations indicated above. Sampling the data for this specific application requires 167 μs. Using the Crosscorrelation predistorter, all data can be used to estimate the non-linear characteristics of the PA. For the LS predistorter, the DSP takes 2.3 ms to produce new estimates of the PA non-linearity leading to an effective data usage of 7.3 %. One of the consequences of the difference in required processing capacity is that the Crosscorrelation predistorter will respond much faster to changes in PA characteristics than the LS predistorter. For the LS predistorter, it will take at least 2.3 ms to respond to changes. Because we do not directly use new estimates of the PA non-linearity but we average it with a previous setting, the response to changes wil take longer than 2.3 ms for the LS predisorter. The Crosscorrelation predistorter responds 50 times faster.

Since we analyzed only one specific application on one specific processor, the figures that result from the overall system timing analysis, are not of highest importance. However, it is justified to conclude that, by adding relatively simple hardware to calculate the crosscorrelation functions, the Crosscorrelation predistorters can respond faster to changes in the behavior of a PA than the LS predistorter.

5.3.5 Evaluation of the Crosscorrelation predistorter

To evaluate the overall performance of the Crosscorrelation predistorter, we relate the relevant performance- and cost metrics of the Crosscorrelation predistorter to the metrics of the LS predistorter. The metrics are summarized in table 5.1

	LS predistorter	Crosscorrelation predistorter
Complexity	large	small
Complexity scalability	scales with Integration Length	fixed
Analog Complexity	scales with resolution ADC	scales with resolution ADC
Quantization sensitivity	< 4 bit	< 10 bit
Average ACI reduction	good	good

Table 5.1: Comparison between the LS predistorter and the Crosscorrelation predistorter

The Crosscorrelation predistorter has a lower digital complexity than the LS predistorter. For the LS predistorter, the number of complex multiplications scales linearly with the number of samples T on which a new estimate of the PA model characteristics is based. For the Crosscorrelation predistorter, the number of complex multiplications is of order $N\,{}^2\log N$ where N equals the number of lags of the crosscorrelation function. In general, T is much larger than N (one or two orders of magnitude).

The quantization sensitivity is defined as the resolution-range of the ADC in the feedback path, for which the performance of digital predistortion is clearly affected. The Crosscorrelation predistorter is more sensitive to the non-linear effects of quantization in the feedback path (see figure 5.6). To deal with coarse quantization below 3 bits, both predistorters need a dither signal, added to the analog feedback signal before quantization. The effects of lowering the number of bits in the ADC in the feedback path can be compensated by increasing the Integration Length. To achieve equal performance, the Crosscorrelation predistorter requires a longer Integration Length than the LS predistorter. However, the complexity of the digital processing part (the Predistortion Control block) of the Crosscorrelation predistorter is not increased if the Integration Length is increased. For the LS predistorter, the digital complexity increases linearly with the Integration Length.

For both the LS- and Crosscorrelation predistorter, the average reduction of the power level in the adjacent channels is good. If we look at an ACI level of -50 dB (an important limit for the GSM, UMTS, IEEE802.11b and IEEE802.11g standards), the maximum standard deviation of the signal at the input of our PA model equals approximately 0.08, if no predistortion is used. If predistortion is used, irrespective whether this is an LS predistorter or a Crosscorrelation predistorter, the maximum input standard deviation equals approximately 0.1, giving an increase of power at the input of the predistorter of 2 dB. This is a significant increase.

The simulator contains a general signal source not related to a specific standard and a specific PA model. When using a predistorter with a specific standard (= a specific signal source) and a specific PA, the performance will differ from the performance reported. Any situation requires additional simulations.

The last remark concerns the update rate of the predistorter. As mentioned, there is a relation between the resolution of the ADC in the feedback path and the length of the vectors used to calculate the crosscorrelation functions (the Integration Length). If the resolution is lowered, the length of the vectors has to be increased. Using longer vectors implies that the rate at which the predistorter is updated, is lowered. The digital predistorter should, however, be able to track the changes of the PA characteristics. On the time-scales used within digital predistortion, the effects of aging can be neglected and the changes are dominated by the temperature variations. Thus, the resolution of the ADC should be chosen such that the digital predistorter is still able to track the effects of the temperature changes. When using the LS predistorter in combination with a Texas Instruments C67x DSP, the update rate will be limited by the processing capacity needed. When using coarse quantization, the amount of data needed to estimate the non-linearity of the PA will increase and the time between two updates of the predistorter will increase accordingly. For equal resolutions in the feedback path, when using coarse quantization, the Crosscorrelation predistorter will require more data than the LS predistorter. However, the Crosscorrelation predistorter can process this data much faster than the LS predistorter. System level timing of the Crosscorrelation predistorter will therefore not be determined by the processing capacity but by the data gathering process.

We cannot draw a general conclusion on which predistorter will lead to the fastest update rate. This has to be analyzed for every specific application and every specific implementation of the Predistortion Control block. However, the processing capacity required by the Crosscorrelation predistorter will be significantly smaller.

Chapter 6

Validation of results

In this chapter, we present the results of simulations using information provided by Philips Semiconductors. First we will present the performance and compression of the different predistortion schemes, using a PA model provided by Philips. Second we will use data, obtained from measurements at the input and output of a PA, to construct a model of the PA. These measurements were also done by Philips Semiconductors. Both the LS- and Crosscorrelation predistorters construct a memory polynomial in the Direct Learning approach which can act as a PA model. The original model provided by Philips and the models constructed by the LS- and Crosscorrelation predistorters are excited with the measured input data. The resulting output data sets are compared with the measured data in both the time- and frequency domain. At the end of this chapter, we will briefly reflect on the process on designing a digital predistortion system using coarse quantization, based on the Y-chart approach.

6.1 Introduction

The analysis of the LS- and the Crosscorrelation predistorters in chapters 4 and 5, was done via simulations using PA models from literature. To realize a hardware setup would have required too much effort in relation to the goal of the project. However, we validated our approach using data provided to us by Philips Semiconductors, Nijmegen. This data consisted of:

- A model of a PA, based on measurements on an actual PA. The PA is a device, dedicated to 2.11-2.17 GHz WCDMA operation, biased in class AB. The typenumber of the device is BLF4G22-100 and it is realized in the Philips fourth generation LDMOST technology.

- The actual results of measurements, consisting of a sampled stimulus signal and a sampled response signal.

We used the first part, the model of a PA, within our simulator and determined the performance of the different predistortion schemes. The actual measurements are used to extract a PA model using the algorithms within the LS- and Crosscorrelation predistorters.

6.2 Digital Predistortion

The specific PA model, provided by Philips, is described by the following expression:

$$y(n) = f(|x(n)|) \cdot e^{i \cdot g(|x(n)|)} \cdot \frac{x(n)}{|x(n)|} \tag{6.1}$$

where

$$\begin{aligned} f(x) &= 0.0204 + 0.8445\ x + 0.2918\ x^2 - \\ &\quad 0.0588\ x^3 - 0.0729\ x^4 + 0.0175\ x^5 \\ g(x) &= 0.0660 - 0.0388\ x + 0.1963\ x^2 - \\ &\quad 0.2777\ x^3 + 0.1140 x^4 - 0.0146 x^5 \end{aligned}$$

The polynomials $f(x)$ and $g(x)$ describe the so-called AM-AM or Amplitude-Amplitude distortion and AM-PM or Amplitude-Phase distortion respectively. This PA model has been incorporated into the simulator. In the model we see that, besides odd polynomial basis functions, even polynomial basis functions are present as well. In the simulations described in chapters 4 and 5 and in appendix B, we only use odd polynomial basis functions up to the fifth degree. This was done to reduce the complexity of the final predistortion system. Furthermore, adding even polynomials and adding polynomials with a degree higher than five, does not change the algorithms fundamentally.

To cope with the PA model provided by Philips, we have to incorporate even polynomial basis functions up to the fourth degree as well. The general model for the behavior of a PA, used in the Predistortion Control block becomes (see also expression 4.32):

$$x_2(t) = \sum_{k=0}^{5} \sum_{\tau=0}^{\tau_{max}-1} a_{k\tau} \gamma_k(x_1(t-\tau)) \tag{6.2}$$

where

$$\begin{aligned}\gamma_k &= \phi_k, \text{ when using normal polynomials}\\ \gamma_k &= \psi_k, \text{ when using orthogonal polynomials}\end{aligned}$$

and

$$\phi_k(x) = x \mid x \mid^{k-1}$$

$$\begin{aligned}\psi_0(x) &= x \mid x \mid^{-1}\\ \psi_1(x) &= x\\ \psi_2(x) &= 4 \mid x \mid x - 3x\\ \psi_3(x) &= 15 \mid x \mid^2 x - 20 \mid x \mid x + 6x\\ \psi_4(x) &= 56 \mid x \mid^3 x - 105 \mid x \mid^2 x + 60 \mid x \mid x - 10x\\ \psi_5(x) &= 210 \mid x \mid^4 x - 504 \mid x \mid^3 x + 420 \mid x \mid^2 x\\ &\quad - 140 \mid x \mid x + 15x\end{aligned} \tag{6.3}$$

The orthogonal polynomials are from [66].

Using the PA model provided by Philips and the extensions described above, we determined the performance of the LS- and Crosscorrelation predistorters. The performance and corresponding compression for the LS predistorter are presented in figures 6.1 and 6.2.

The performance and corresponding compression for the Crosscorrelation predistorter are presented in figures 6.3 and 6.4.

If we first concentrate on the ACI levels in case no predistortion is used, we see that the average power in the adjacent channels decreases for decreasing standard deviations of the input signal x, except for small input standard deviations. This is caused by the offset (the zero-order term) in the AM-AM distortion ($f(x)$ in expression 6.1) of the PA model. Although we can determine this offset by incorporating a zero-order term in the memory polynomial, we cannot compensate for it simply because amplitudes below the offset cannot be generated. So, for small input standard deviations, the effect of this offset becomes relatively large.

Both the LS- and Crosscorrelation predistorters perform well for standard deviations up to 0.15. The ACI levels decrease for increasing standard deviations because of the reduced effects of the offset in the AM-AM distortion. Beyond a standard deviation of 0.15, the ACI levels increase. For the LS predistorter, the Indirect Learning schemes lead to unstable behavior for relatively large input standard deviations. For the Crosscorrelation predistorter, the Indirect Learning schemes show better performance than the Direct Learning schemes for standard deviations beyond 0.175.

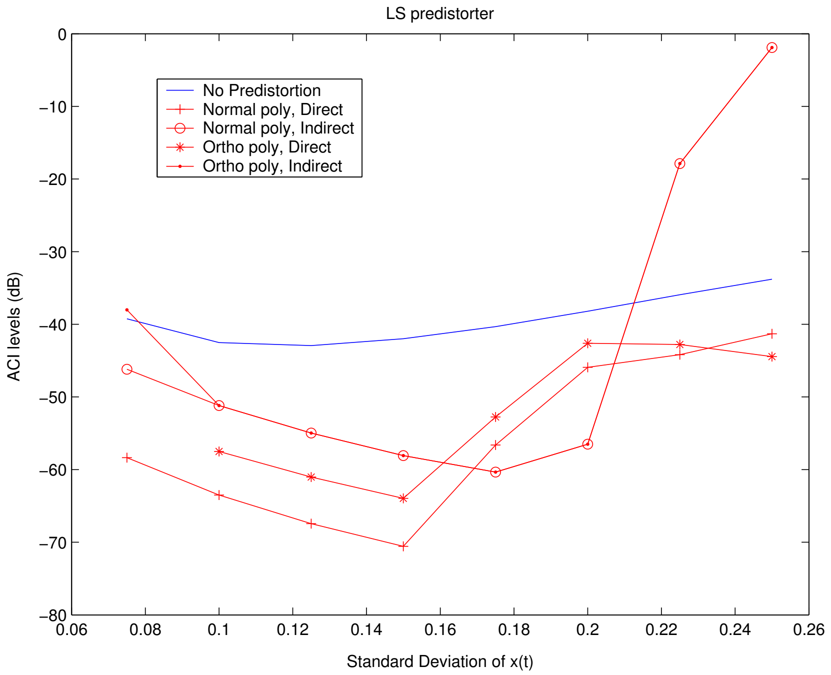


Figure 6.1: Performance LS predistorter using the Philips PA model

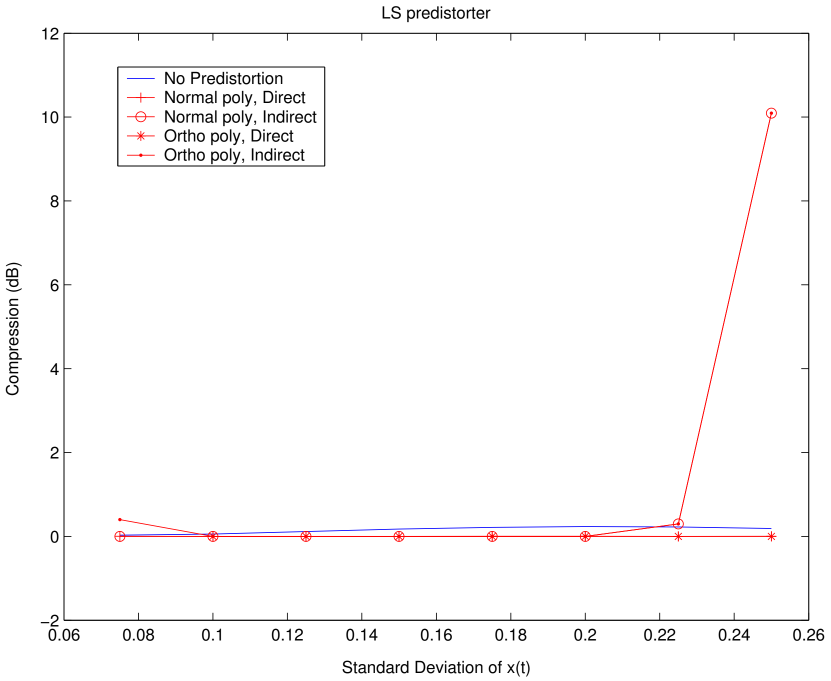


Figure 6.2: Compression LS predistorter using the Philips PA model

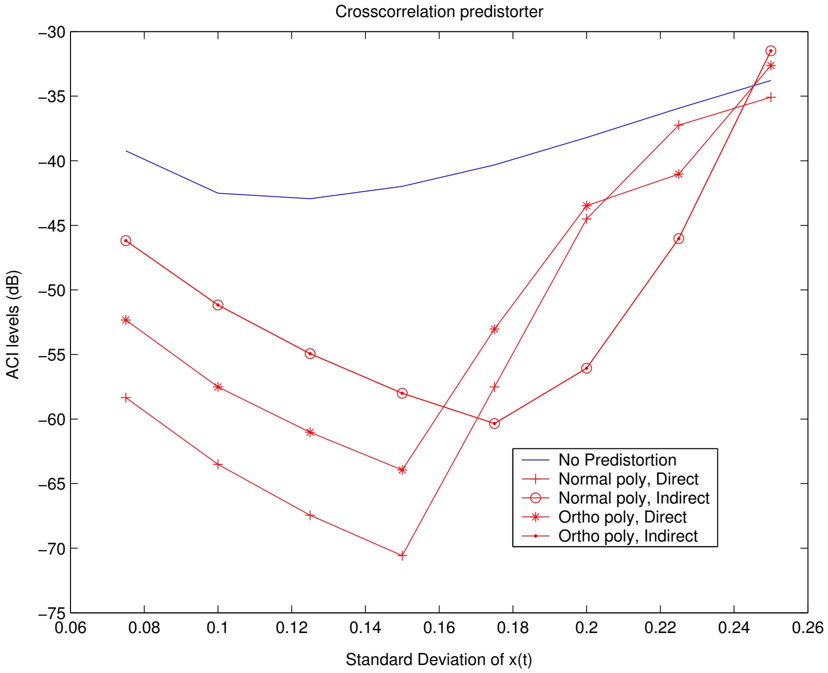


Figure 6.3: Performance Crosscorrelation predistorter using the Philips PA model

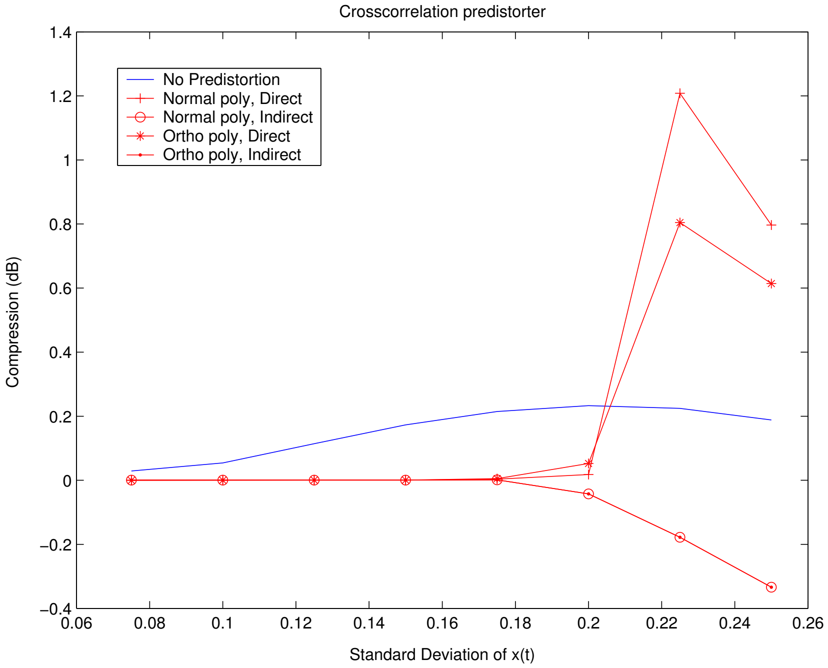


Figure 6.4: Compression Crosscorrelation predistorter using the Philips PA model

Concluding, both the LS- and Crosscorrelation predistorter lead to reduced ACI levels when using the PA model provided by Philips. Especially for relatively large input standard deviations, the Crosscorrelation predistorter, using the Indirect Learning scheme, shows the best performance.

6.3 PA model extraction

Besides the PA model, Philips Semiconductors provided us with a data set, consisting of approximately 50k samples of an input stimulus of a PA and the corresponding samples of the response. This data cannot be used in our standard simulations resulting in ACI levels and compression figures since the data only gives the relation between the input and output for a limited number of samples. Because of predistortion, the input signal changes (slightly) and because of the limited size of the dataset, it is not guaranteed that the corresponding output can be found within the dataset. However, the data can be used to construct a PA model. Philips used the data to construct the model presented in the previous section.

Within the Direct Learning schemes of our simulator, we also determine a model of the behavior of the PA (see figure 4.6); the PA is modeled as a memory polynomial and within the Predistortion Control block, the coefficients of the memory polynomial are determined. We use two different methods leading to the LS- and the Crosscorrelation predistorters. Together with the Philips model, we have three models. If we excite these three models with the input stimulus, we have three response signals. We will compare these three response signals with the original response signal to determine the quality of the three models.

To construct the models, based on the LS- and Crosscorrelation predistorters, we had to change our simulator slightly. Instead of using the input x_p and the output y of a PA model as inputs for the Predistortion Control block, we used the Philips stimulus- and response data. Within the Predistortion Control block, we used both the algorithms for the LS- and Crosscorrelation predistorter. Initial simulations have shown that the PA models, produced within the Predistortion Control block when using normal polynomials, are almost equal to the ones produced when using orthogonal polynomials. The initial simulations have also shown that the Crosscorrelation predistorter, as described in chapter 5, does not generate an adequate PA model. This is illustrated in figures 6.5 and 6.6.

Figure 6.5 is an AM-AM distortion plot. It presents for all input samples, the absolute value of the output sample as a function of the absolute value of the corresponding input sample. The AM-AM distortion is given for the original measured data provided by Philips. We also excited the PA model, determined by the crosscorrelation algorithm, with the Philips stimulus and determined the corresponding

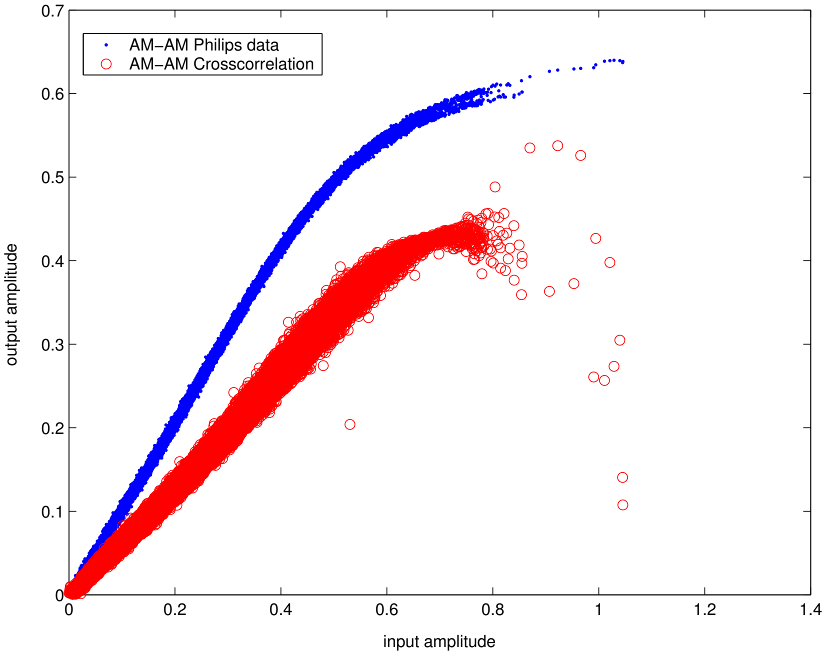


Figure 6.5: AM-AM distortion of the Philips data and the Crosscorrelation model

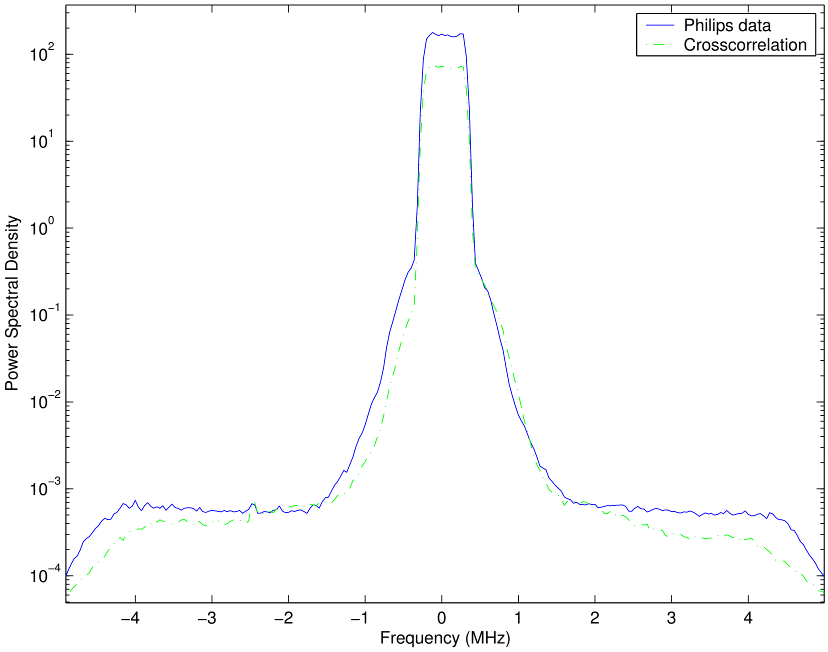


Figure 6.6: Spectra of the Philips data and the Crosscorrelation PA model output

output. The AM-AM distortion of this signal is given as well. The AM-AM distortion plots show some scattering. This is caused by the memory effects; the output amplitude does not only depend on the instantaneous input amplitude but also on previous input amplitudes. The stronger the memory effect, the higher the scattering. It is clear that the Crosscorrelation predistorter overestimates the memory effects leading to high scattering, especially for large input amplitudes. We see, besides the difference in scattering, a relatively large difference in gain. This can also be seen in figure 6.6, in which the power spectral densities of the Philips response and the response of the Crosscorrelation-based PA model are given. The gain difference is caused by the normalization of the spectra in the Crosscorrelation predistorter (see expressions 5.21 to 5.24). This normalization was introduced to force the Crosscorrelation predistorter to reduce the *relative* error between the spectrum at the output of the Predistortion block and the spectrum at the output of the PA model. Because of this normalization of spectra, the relative error is more or less constant over the whole frequency domain, including the primary channel. Errors in the primary channel lead to relatively large variations in power. In this case, we see a reduced power level in the primary channel which leads to less gain in the AM-AM plot (figure 6.5) [1]. When removing the normalization of spectra, proper estimates of the PA are obtained when using the Crosscorrelation predistorter as well. This is illustrated in figure 6.7.

The scattered AM-AM curve for the Crosscorrelation model covers the curve of the Philips data for a large part. For large amplitudes, the deviation between the two curves is largest. The corresponding spectra can be found in figure 6.8 and will be discussed later.

In appendix E, we present the AM-AM and AM-PM distortion plots of the original data, of the polynomial model provided by Philips, of the PA model obtained using the LS predistorter and of the PA model obtained using the Crosscorrelation predistorter in case no normalization of spectra is used.

The spectra of the Philips response data, the response of the model provided by Philips, the response of the PA model using the LS predistorter and the response of the PA model using the Crosscorrelation predistorter without normalization of

[1] The situation where the Philips stimulus and response are offered to the Predistortion Control block corresponds to the case where an input signal with standard deviation 0.25 is offered to the Philips PA model, consisting of the AM-AM and AM-PM polynomials. For the Crosscorrelation predistorter using orthogonal polynomials and the Direct Learning scheme, this leads to positive compression values (see figure 6.4). This means that the output power is higher than what is to be expected (see expression 3.18). This corresponds with lower gain for the extracted PA model; within the Predistortion Control block, the estimated gain of the PA model is lower than the real gain. Within the Predistortion block, this will be compensated by amplifying the input signal. The overall gain will then increase, leading to the positive compression values. So, the results of the Crosscorrelation predistorter, using orthogonal polynomials for an input standard deviation of 0.25, correspond with the underestimated gain of the PA model.

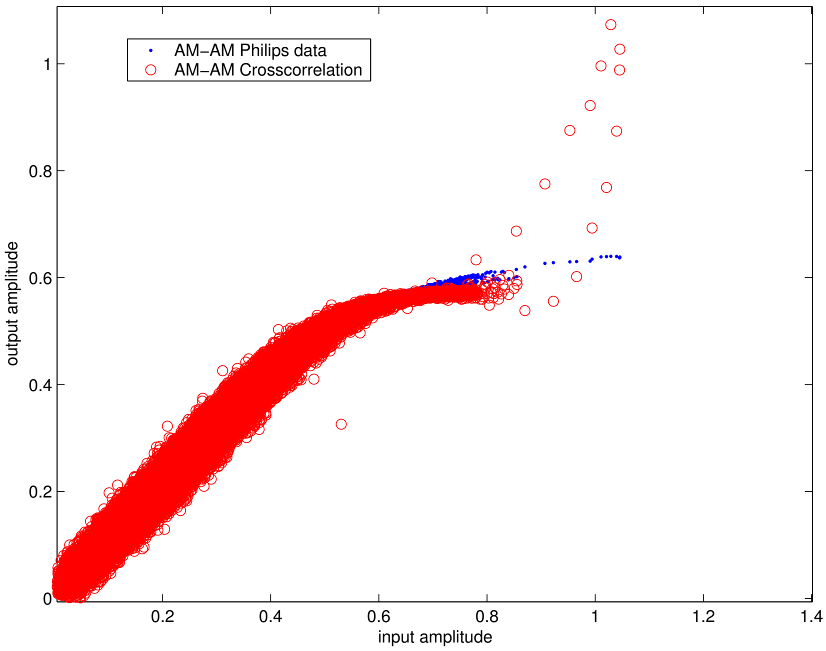


Figure 6.7: AM-AM distortion of the Philips data and the Crosscorrelation model in case no normalization of spectra is used

spectra, are given in figure 6.8.

We see that, outside the primary channel, the estimates of the power of the response of the Crosscorrelation-based PA model is closest to the power of the actual Philips response data.

Besides an analysis of the responses in the frequency domain, we analyzed the responses in the time domain as well. We determined the Mean Square Error (MSE) between the Philips response data and the responses based on the three PA models. The MSEs are given in table 6.1.

PA model	MSE
Philips	$4.0231\ 10^{-5}$
LS based	$1.1\ 10^{-3}$
Crosscorrelation based	$8.1758\ 10^{-4}$

Table 6.1: Mean Square Errors of the PA models

We see that the model provided by Philips Semiconductors gives the best estimate. We also see that the model which gives the smallest MSE in the time domain does not automatically generate a signal with a spectrum that fits the spectrum of

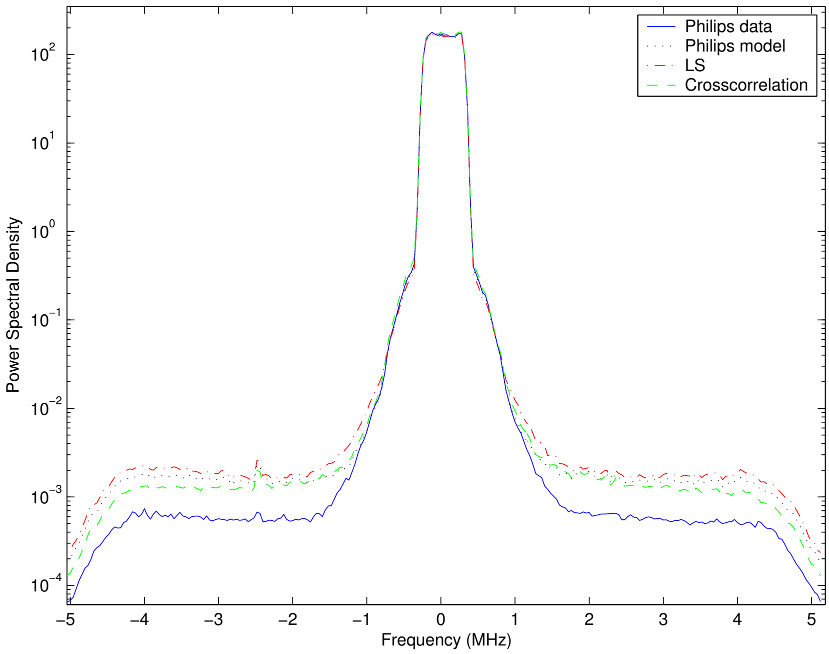


Figure 6.8: Spectra Philips data, Philips model, LS- and Crosscorrelation model

the original response data best. In section 5.3.1, we have argued that this is one of the reasons that the Crosscorrelation predistorter outperforms the LS predistorter for small input standard deviations.

6.4 Conclusion

The simulations, using a PA model based on measurements on a real PA, show that both the LS- and Crosscorrelation predistorter give significant suppression of power in the adjacent channels. To achieve this, we had to extend our simulator. The basic assumption that a PA can be modeled adequately by using only odd polynomials up to the fifth degree, seemed to be incorrect. We had to incorporate even polynomials up to the fourth degree as well. Furthermore we see that the zero-order polynomial term, an offset in the AM-AM plot, results in increasing ACI levels. This offset cannot be corrected.

We also used measured stimulus- and response data to construct a model of the PA. Both the LS- and Crosscorrelation predistorter internally construct such a model. These models, together with the PA model provided by Philips, are used to generate responses using the measured stimulus data as input. The responses are compared with the original response data to determine the quality of the PA models. It makes a difference whether the quality is determined in the time domain or in the frequency domain. In the time domain, we can use the Mean Square Error between the measured response and the generated responses as a measure of the quality. In this case, the PA model provided by Philips resembles the behavior of the real PA best. If we use the power in the adjacent channels as an estimate of the quality of the PA model in the frequency domain, the Crosscorrelation predistorter resembles the real PA best.

Summarized, the simulations using a model based on measurements and using real data, confirm that both the LS- and Crosscorrelation predistorter are capable of reducing power in the adjacent channels. Both predistorters can also be used to generate models of a PA, using measured data.

6.5 Reflection on the design process

In chapters 4 and 5, we have described two iterations of a design exploration using the Y-chart approach. These iterations are briefly described below.

6.5.1 The first iteration: the LS predistorter

In the first iteration, we have gone through all stages of the Y-chart approach. These stages are: ’Architecture Specification’, ’Functional Specification’, ’Mapping’ and ’Performance Analysis’ (see figure 1.11). The most important aspects within the different stages are described below.

- Architecture Specification. We have specified different architectures consisting of an analog processing part, a digital processing part, an ADC in the feedback path and a DAC in the forward path. The different architectures have different resolutions for the ADC in the feedback path.

- Functional Specification. We have specified the LS predistortion algorithm.

- Mapping. We have mapped the LS predistortion algorithm onto the digital processing part.

- Performance Analysis. We have analyzed the performance of the LS predistorter in case of different resolutions of the ADC in the feedback path. The complexity of the LS predistorter is large and scales with the number of samples used to estimate the non-linear behavior of the PA. Both the large complexity and the dependency of the complexity on the number of samples to be analyzed, influence the system timing. By making use of the feedback path from ’Performance Numbers’ to ’Functional Specification’, we start a second iteration. The goal of the second iteration will be to reduce the complexity and to make the complexity independent of the number of samples to be analyzed.

6.5.2 The second iteration: the Crosscorrelation predistorter

In the second iteration, we have gone through the ’Functional Specification’, the ’Mapping’ and ’Performance Analysis’ stages of the Y-chart approach. The ’Architectures’, on which the ’Application’ can be mapped, are not changed. The most important aspects of the different stages are described below.

- Functional Specification. We have specified the Crosscorrelation predistortion algorithm.
- Mapping. Also the Crosscorrelation predistortion algorithm is mapped onto the digital processing part of the architectures.
- Performance Analysis. It is shown that the performance of the Crosscorrelation predistorter concerning suppression of power within the adjacent channels is comparable with the performance of the LS predistorter. Furthermore, the complexity of the algorithm has reduced for typical cases and is independent of the number of samples to be analyzed. However, the Crosscorrelation predistorter is more vulnerable to coarse quantization than the LS predistorter. This can be compensated by analyzing more data to estimate the non-linear behavior of the PA. When using coarse quantization, system timing will therefore not be determined by the complexity of the algorithm but by the time it takes to acquire sufficient samples.

The results of the two iterations are two alternatives for digital predistortion of PAs: the LS- and the Crosscorrelation predistorter. When using coarse quantization, the system timing of the LS predistorter depends on the data processing capacity that is available, and for the Crosscorrelation predistorter on the data acquisition time. Which solution is best depends on the system specifications and the realization of the digital processing part. Once these aspects are known, a final ’Functional Specification’ activity should start, based on the ’Performance Numbers’.

Chapter 7

Introduction to Radio Astronomy

7.1 Introduction

In chapters 7, 8 and 9, we describe a second case study to illustrate the Analog-Digital Codesign process using the Y-chart approach. In this case study, the design of a Tied Array Adder (TAA) for a Radio Astronomy receiver is described. The TAA is part of the Westerbork Synthesis Radio Telescope (WSRT) and we investigate the implementation in both the digital and analog domain. The design is elaborated in the context of Analog-Digital Codesign.

The WSRT is built and exploited by ASTRON, the Netherlands Foundation for Research in Astronomy. ASTRON is an institute of NWO (Nederlandse Organisatie voor Wetenschappelijk Onderzoek), an organization that stimulates scientific research in the Netherlands. The goal of ASTRON is 'to promote the orderly and successful development of the science of astronomy in the Netherlands'. One of the tasks is to develop, realize and exploit Astronomical instrumentation and more in particular, Radio Astronomical instrumentation. Radio Astronomical instrumentation is a subject not many people are familiar with so, in this chapter, a brief introduction on Radio Astronomy and -instrumentation will be given. In chapters 8 and 9, the focus will be on the Analog-Digital Codesign of a TAA.

7.2 Radio Astronomy

The starting point of this brief introduction on Radio Astronomy is Planck's Radiation Law (from [81], p. 76) concerning the transmitted power of a perfect radiator:

$$B = \frac{2kv^3}{c^2} \cdot \frac{1}{e^{\frac{hv}{kT}} - 1} \tag{7.1}$$

where

$B =$ brightness (Watt m^{-2} Hz^{-1} sterradian^{-1})
$h =$ Plancks′s constant (= 6.63 10^{-34} joule sec)
$v =$ frequency (Hz)
$c =$ velocity of light in vacuum (= 3 10^{8} m sec^{-1})
$k =$ Boltzmann′s constant (= 1.38 10^{-23} joule K^{-1})
$T =$ temperature (K)

This expression gives the transmitted power of the electromagnetic energy per unit area, per unit solid angle, per unit bandwidth (Watt m^{-2} sterradian^{-1} Hz^{-1}) for an idealized object, the ’blackbody’. A blackbody is a perfect radiator (see [77]). For this introduction on Radio Astronomy, there are two important aspects concerning the law mentioned above:

1. If an object has a temperature which is above the absolute minimum temperature ($T > 0$ K), this object radiates electromagnetic energy.
2. The energy is, in principle, transmitted at all frequencies. Note that the radiated energy does not have equal power at all frequencies.

Besides the thermal blackbody radiation there are two other sources of so-called continuum radiation (emission): thermal emission from ionized gas and synchrotron radiation. Thermal emission from ionized gas is caused by a free electron whose direction of movement is deflected when it is passing near a proton. Because of this deflection, electromagnetic energy is transmitted over a relatively large frequency range. Synchrotron radiation is non-thermal emission where electrons are being accelerated to very high energies in a strong magnetic field. In this process, electromagnetic energy is transmitted. It is not within the scope of this thesis to explain these effects in more detail.

The effects mentioned above (blackbody radiation, emission from ionized gas and synchrotron radiation) involve a relatively large frequency band and are therefore called continuum emissions. Spectral line emissions and absorptions involve a relatively small part of the electromagnetic spectrum. Spectral line *emission* occurs when atoms are bombarded with particles such as electrons or protons. A collision may give the atom enough energy to change it from the stationary energy state to a higher energy state. Excited atoms spontaneously fall-back to states of lower energy where energy is emitted in the form of electromagnetic radiation of a specific frequency. Spectral line *absorption* occurs when electromagnetic radiation passes through a cloud of atoms. Radiation of certain frequencies is absorbed. The specific frequencies are related to the discrete energy levels of the

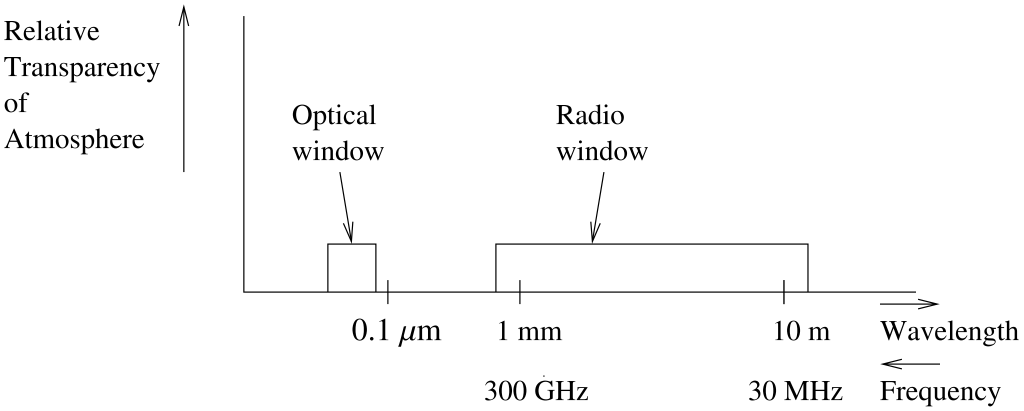


Figure 7.1: Relative transparency of the earth's atmosphere at different wavelengths

atoms. Neutral hydrogen (H1) is one of the most abundant elements in space comprising about 70% of the mass and 90% of the atoms in interstellar gas. It radiates energy in a spectral line at 1420 MHz (21 cm wavelength). Another set of spectral lines ranges between 1612 MHz and 1720 MHz (18 cm). They are caused by absorption through Hydroxyl (OH) clouds.

A spectral line enables the measurement of movement of objects. If the object which is radiating or absorbing, is moving, the frequency of the line changes. This is well-known as the Doppler effect. Since essentially all of the galaxies are moving away from us according to the big-bang theory, the frequencies of the spectral lines become lower. The spectral lines are said to be shifted towards the 'red end' of the spectrum. The more distant galaxies are moving away from us faster leading to higher 'redshifts' of spectral lines.

From the above, one can accept that celestial objects radiate within a large frequency range. The radiation emitted by celestial objects can be detected and can be used to make images of these objects. For earth-based observations, this radiation has to pass the earths atmosphere. The earths atmosphere is not transparent at all frequencies. Figure 7.1 indicates very roughly the relative transparency of the atmosphere as a function of wavelength (frequency).

Basically there are two windows in which the atmosphere is relatively transparent: the Optical window and the Radio window. The Optical window enables the detection of electromagnetic radiation that is visible for human beings as well. The Radio window was first discovered by Karl G. Jansky (see [78]) in 1933. The two windows have basically led to two different types of earth-based Astronomical instrumentation: Optical Telescopes and Radio Telescopes. In this chapter, we will concentrate on the design of a part of a Radio Telescope.

Within Radio Astronomy, the two types of emission (continuum and spectral

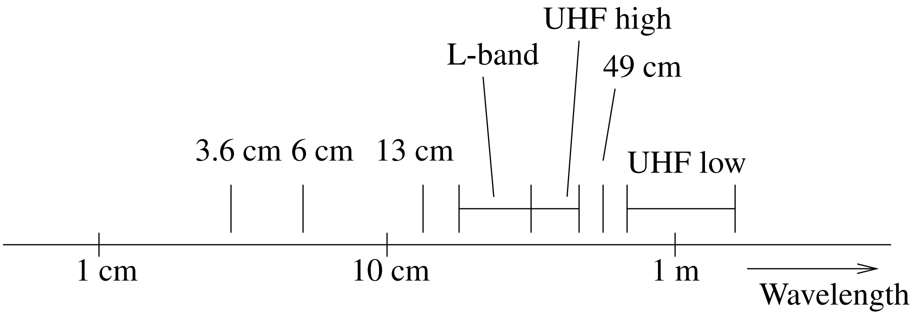


Figure 7.2: Frequency bands of the WSRT Multi Frequency Front End

line emission) lead to two corresponding types of observations. Within continuum observations, the total received power within a relatively large part of the Radio window is measured. This total power value is used to make maps of celestial objects or to measure transient-effects like for example the timing of pulsars and γ-ray bursts. With spectral line observations, a part of the Radio window is split into many narrow frequency bands or frequency channels. A frequency channel is generally more narrow than a spectral line. This way, spectral lines can be detected.

From figure 7.1 we have seen that the Radio window roughly ranges from 30 MHz (10 m) to 300 GHz (1 mm). This window is too wide to be captured with a single receiver. Therefore, a Radio telescope uses different receivers for different parts of the Radio window. For example, the Westerbork Synthesis Radio Telescope (WSRT) in Westerbork, the Netherlands, is equipped with Multi Frequency Front Ends which can operate at at least 7 different frequency bands: 3.6 cm, 6 cm, 13 cm, L-band, UHF high, 49 cm and UHF low (see figure 7.2).
Some of the receivers observe at frequencies around known spectral lines (18 and 21 cm within the L-band). Other receivers (UHF low and UHF high) have a wider frequency window and are used to measure red-shifted spectral lines. The 6 and 49 cm receivers are used for continuum observations. The 3.6 and 13 cm-receivers are mostly used for continuum observations in combination with other Radio Telescopes.

7.3 Radio Telescopes

Radio Telescopes can be categorized into two groups: single-dish and multiple-dish. A single-dish telescope is equipped with a front-end connected to a backend. In multiple-dish telescopes, each dish contains a front-end and these front-ends are connected to a single back-end. Both are schematically illustrated in figure 7.3.

A front-end captures the electromagnetic energy, amplifies the signal and trans-

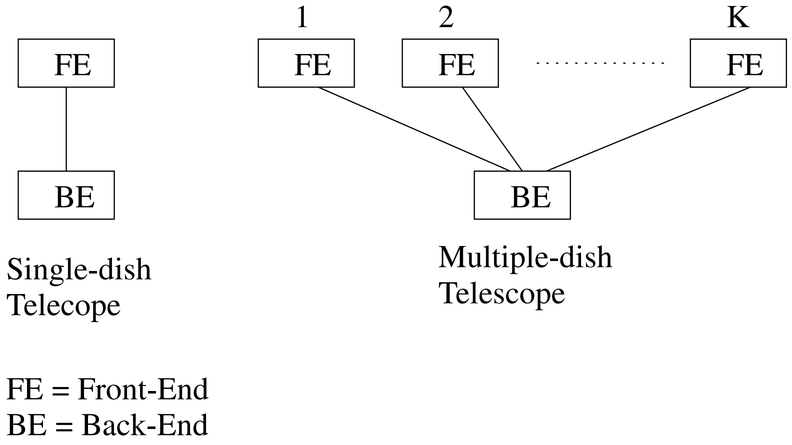


Figure 7.3: Schematic overview of single-dish and multiple-dish telescopes

forms it to a frequency, suitable for transportation of the signal and for processing furtheron. It is important to note that, during the first amplification in the front-end by means of a Low Noise Amplifier (LNA), noise is added to the received astronomical signal. In general this signal is very weak and the power of the added noise exceeds the power of the astronomical signal. The aim of, for example, the WSRT is to be able to detect astronomical signals which are 70 dB below the added receiver noise. In general, within a front-end, the signals are down-converted from the Radio Frequency range (RF) to an Intermediate Frequency range (IF) and filtered. The signals are then transported to a back-end.

The general characteristic of a back-end is that it processes the received signal in real-time to such an extend that the information, extracted from the signal, can be stored. After storing the information, it can be used for off-line processing. In this last step, the information can, for example, be used to make maps of a celestial object. For a single-dish telescope, a generic block diagram of a back-end is presented in figure 7.4 (from [81], p. 241).

The input of the back-end is the IF signal from the front-end. By means of a (tunable) Bandpass Filter, a part of the spectrum of the FE-signal is selected. In analog back-ends, this signal is then directly connected to a Square Law Detector (SLD). *Digital* radiometry was first proposed in [79] for use in radio astronomy. In digital back-ends, the FE-signal is generally frequency-converted to a baseband frequency and then converted from the analog domain into the digital domain. For the brief introduction on Radio Astronomy, these additional steps are not relevant.

The bandwidth of the signal at baseband might be smaller than the bandwidth of the signal at IF. The bandwidth of the signal at the input of the Square Law Detector is generally named the predetection bandwidth $\Delta\nu$. Its value might, for example, be determined by the filters in the front-end or by the Bandpass Filter in

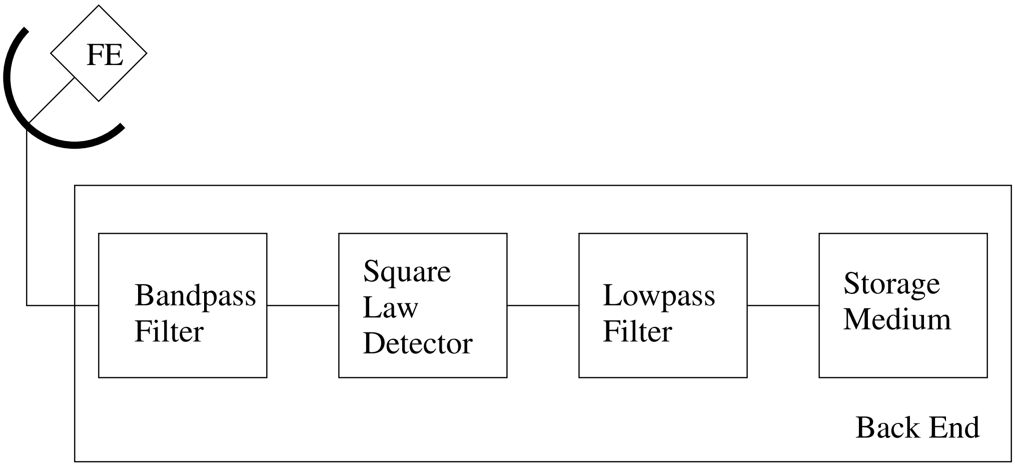


Figure 7.4: Block diagram of a back-end for a single-dish telescope

the back-end.

The general characteristic of the Square Law Detector is that its output is proportional to the power of the input signal. This can, for example, be implemented by means of an analog multiplier in case of an analog back-end, or a digital multiplier in case of a digital back-end. Both inputs of the multiplier are connected to the output of the Bandpass Filter. The output of the Square Law Detector is still fluctuating. If for example the input of the Square Law Detector is a Gaussian signal with a standard deviation σ and zero mean, the average value at the output equals σ^2 and the standard deviation equals $\sqrt{2}\sigma^2$. It is, however, the average value of the output of the Square Law Detector which is the important measure. To reduce the standard deviation, the output of the Square Law Detector is filtered. In an analog back-end, an analog low pass filter is used. In a digital back-end, samples are averaged for a time t called the postdetection integration time. To obtain a further reduction of the standard deviation, the average value of N of these values can be calculated. The output of the Lowpass filter is then stored and the value is used for off-line processing. The spectra at the different stages of a backend are presented in appendix F.

As mentioned before, the signal at the output of the Lowpass Filter contains, besides a measure of the power of an astronomical signal also a measure of the power of the noise, introduced by the Low Noise Amplifier within the receiver.

In figure 7.5, an example of the output of a Lowpass Filter is given as a function of time where, first the receiver is pointed towards a silent part of the sky from which practically no radiation is received (off-source), then towards a strongly radiating celestial object (on-source) and finally again towards a silent part. The increase in power, due to the astronomical source, can easily be detected. When

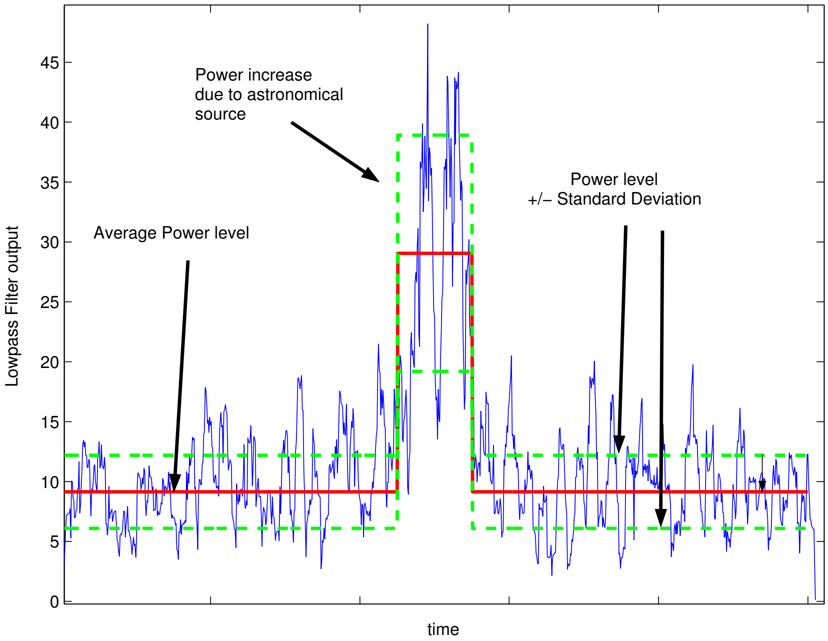


Figure 7.5: Example of the output of the back-end Lowpass Filter

measuring off-source, the power of the signal introduced by the receiver (receiver noise) can be determined. We assume that the 'silent part' is not emitting any radiation. (In practice, from any part of the sky, there always is some radiation emitted. This radiation is called 'background radiation'. For reasons of simplicity, this signal is often considered as a part of the receiver noise even though the signal is correlated for all antennas).

When measuring on-source, the power of the sum of the receiver noise and the astronomical signal can be determined. If the receiver system is stable and the gains within the system are known, the absolute value of the power of the astronomical signal can be determined.

If the power difference between off-source and on-source measurements is small, it becomes more difficult to determine the absolute power of the astronomical signal. A general rule is that the smallest power increase that can be measured equals the standard deviation at the output of the Lowpass Filter. This value is related to a parameter called 'sensitivity'. To give a more formal definition of sensitivity, we have to return to the front-end and determine the power received by the front-end. It is inevitable to adopt the 'temperature language' which is common practice in Radio Astronomy. Temperature, given in Kelvin (K), multiplied with two times Boltzmann's constant (k = 1.38 10^{-23} Watt Hz^{-1} K^{-1}) is a measure of power. Because of this relation, temperature is very often used as a measure of power. An antenna is said to have an antenna temperature T_A if the power it receives per unit frequency (per Hz) equals $2kT_A$ (Watt). A typical Radio Telescope is equipped with a reflector. This reflector reflects the energy incident on it in such way that it is gathered at the front-end. A reflector of a telescope always has a finite surface area (aperture). The effective aperture is denoted with A_e. The power per unit frequency per unit area is called the 'observed flux density of a radio source' S_0 and equals (from [81], p. 99):

$$S_0 = \frac{2kT_A}{A_e} \tag{7.2}$$

where

$$\begin{aligned} S_0 &= \text{ flux density (Watt Hz}^{-1}\text{ m}^{-2}\text{ or Jansky(Jy))} \\ k &= \text{ Boltzmann's constant (= 1.38 10}^{-23}\text{ Watt Hz}^{-1}\text{ K}^{-1}) \\ T_A &= \text{ antenna temperature (K)} \\ A_e &= \text{ effective antenna aperture (m}^2) \end{aligned}$$

The source flux is expressed in Watt per Hertz per square meter. This unit is also called Jansky (Jy) after the discoverer of the Radio window.

Within a front-end, the electronic components add noise to the received signal. This so-called receiver noise corresponds with a receiver temperature T_R and the power of the receiver noise equals $2kT_R$. The temperature of the total signal T_{sys} then equals:

$$T_{sys} = T_A + T_R \tag{7.3}$$

In general, T_R is much higher than T_A (up to 70 dB) and hence the system temperature T_{sys} is much higher than the antenna temperature T_A. For that reason, the fluctuation at the output of the Lowpass filter is dominated by the receiver temperature. Using these definitions and the results from appendix F, we can determine the standard deviation of the output of the lowpass filter of a backend. The power that is received per unit frequency equals $2kT_{sys}$. The total power at the input of the SLD equals $2kT_{sys}\Delta\nu$. This corresponds to the parameter σ^2 in appendix F. In this appendix, the RMS value of the AC component at the output of the lowpass filter, in case of a digital backend, is calculated and equals: $\frac{\sigma^2}{\sqrt{\Delta\nu\ t\ N}} = \frac{2kT_{sys}\ \Delta\nu}{\sqrt{\Delta\nu\ t\ N}}$. In general, this value is normalised to frequency and collecting area to relate it to an increase of the observed flux density of a radio source. The standard deviation on the observed flux density after detection is then given as (from [81], p. 103):

$$\Delta S_{rms} = \frac{2kT_{sys}\ \Delta\nu}{\sqrt{\Delta\nu\ t\ N}} \cdot \frac{1}{\Delta\nu\ A_e} = \frac{2k}{\sqrt{\Delta\nu\ t\ N}} \cdot \frac{T_{sys}}{A_e} \tag{7.4}$$

where

$$\begin{aligned}\Delta S_{rms} &= \text{ standard deviation (Watt Hz}^{-1}\text{ m}^{-2}) \\ T_{sys} &= \text{ system temperature (K)} \\ \Delta\nu &= \text{ predetection bandwidth (Hz)} \\ t &= \text{ postdetection integration time (s)} \\ N &= \text{ number of records averaged (dimensionless)}\end{aligned}$$

The other parameters have been introduced previously. As said ΔS_{rms} corresponds to the smallest power increase that can be measured. Expression 7.4 is used to indicate the minimum increase of power that can be detected called 'sensitivity':

$$\Delta S_{min} = \Delta S_{rms} \tag{7.5}$$

where

$$\Delta S_{min} = \text{ minimum detectable flux density(Watt Hz}^{-1}\text{ m}^{-2})$$

The sensitivity indicates the minimum power per unit area per Hz, a single-dish telescope is able to detect. One effect can be deduced from this expression: the minimum detectable flux density decreases, which means that weaker signals can be detected, if the postdetection integration time (t) and/or the number of records averaged (N) increase. In theory the minimum detectable flux density can be reduced to any value by increasing the $t \cdot N$ product. In practice, this is limited by the stability of the receiver. The minimum detectable flux density also decreases if the predetection bandwidth $\Delta\nu$ increases. This effect is explained in appendix F.

If the system temperature increases, the minimum detectable flux density increases which means that an astronomical source has to radiate stronger to be detected. The system temperature can be lowered (less receiver noise is produced) by cooling the front-end with, for example, Helium. We also see the effect that, if larger Radio Telescopes are used (increasing the effective aperture), weaker sources can be detected.

The size of the Radio Telescope does not only determine the sensitivity but also the resolution. Resolution generally indicates the number of different (celestial) objects a Radio Telescope is able to detect. A very loose analogy is the number of pixels that are available for the representation of an object. A very general measure of the resolution of a Radio Telescope is the half power- or 3 dB beamwidth (W_{3dB}). It is defined as (see also [81], p. 186):

$$W_{3dB} = p\frac{\lambda}{D} \tag{7.6}$$

where

$$\begin{aligned} W_{3dB} &= \text{half power beamwidth(rad)} \\ p &= \text{telescope dependent factor (dimensionless)} \\ \lambda &= \text{wavelength(m)} \\ D &= \text{diameter(m)} \end{aligned}$$

The half power beamwidth is proportional to the wavelength used for the observations and a telescope dependent factor p. The factor p depends on the shape of the telescope reflector and on the method of illumination of the front-end. The half power beamwidth is inversely proportional to the diameter D of the telescope.

As an example, the half power beamwidth of a 25 meter dish telescope operating at 1420 MHz (21 cm), with $p = 1$, equals 8.4 10^{-3} rad which equals approximately 0.5 degrees. A measure of arc that is very often used in Radio Astronomy is 'arc seconds' (arcsec). One degree equals 3600 arcsec. The resolution of the telescope therefore equals 1800 arcsec which is roughly the diameter of the moon as it is observed from the earth. So an area as large as the moon is covered by one half power beam of a 25 meter dish telescope at 21 cm.

7.3.1 Improving telescope performance

We have identified two important parameters that reflect the performance of a Radio Telescope: sensitivity (7.5) and resolution (7.6). For single-dish telescopes, both parameters 'improve' if a larger dish is used. The sensitivity increases (the minimum detectable flux density decreases) because of the increasing effective aperture of the antenna A_e. The resolution increases (the half power beamwidth decreases) because of the increasing diameter of the antenna (D). However, increasing the diameter severely hampers the mechanical construction of the Radio Telescope, especially if the dish has to be steerable. In some cases, a creative solution is found to solve some of the mechanical problems. An example is the Arecibo telescope. The reflector of the telescope covers the faces of a carefully selected valley. This way, a telescope with approximately 305 meter diameter is created.

Another way to increase the resolution and sensitivity is to use multiple dishes. The distance between dishes is generally much larger than the diameter of a single dish. The resolution increases linearly with the maximum of the distances between the telescopes. So, D in expression 7.6 equals the maximum of all distances between any set of two telescopes. The sensitivity of a multiple-dish telescope which consists of identical single dish telescopes, increases if the number of telescopes increases. In section 7.3.2 an expressions for the sensitivity of different types of multiple-dish telescopes, similar to expression 7.5 (7.4), will be presented.

An example of a multiple-dish telescope is the Westerbork Synthesis Radio Telescope in the Netherlands. It consists of 14 identical 25 meter single-dish telescopes. This implies that the effective antenna aperture A_e roughly equals 14 times the effective antenna aperture of a 25 meter single-dish telescope. The maximum distance D between two telescopes equals 3 kilometers. The value D increases from 25 meter to 3 kilometers which means that the resolution increases with a factor 120 (see expression 7.6). The resolution at 21 cm of a 25 meter single-dish telescope was 1800 arcsec and now becomes 15 arcsec.

Future generation Radio Telescopes are planned to have even larger antenna apertures and larger maximum distances between pairs of antennas. The Square Kilometer Array, a concept which should result in a new Radio Telescope within several years, is projected to have an aperture which approximately equals one square kilometer. This is two orders of magnitude more than the aperture of the WSRT at this moment (7000 m^2). The maximum distance between two elements will be around 1000 kilometers. The half power beamwidth at 21 cm becomes 0.05 arcsec. This implies that an area as large as the moon can be filled with 250 million half power beams.

7.3.2 Multiple-dish receivers

7.3.2.1 Delay Compensation

An important aspect in multiple-dish receivers is Pathlength difference- or Delay-compensation. The cause of pathlength differences and the principle behind the compensation of it, are illustrated in figure 7.6.

If the front-ends of a multiple-dish telescope are located in an imaginary flat surface and if the propagation direction of a signal is perpendicular to this surface, the signal amplitudes in the front-end are equal at any moment in time. The imaginary point in the sky from which the signal originates is called the zenith ($\phi = 90^o$). The propagation direction of signals which do not have their origin in the zenith, is not perpendicular to the surface through the front-ends. In figure 7.6 this is illustrated in case of two antennas. Because the angle ϕ is not equal to 90^o, a pathlength difference d exists which leads to a delay τ. This delay has to be compensated in order to detect the signal from direction ϕ correctly. In early multiple-dish telescopes, the pathlength difference was compensated by inserting cables with different length between the front-end and the back-end (for an example see [80]). To give an impression of the length of the cables that have to be inserted, we have to look at the cable velocity of propagation. Some typical values range from 65 % to 85 % of the speed of light in vacuum. So, dependent on the type of cable, the length should range from 65 % to 85 % of the pathlength difference that has to be compensated. For the WSRT, in its current configuration

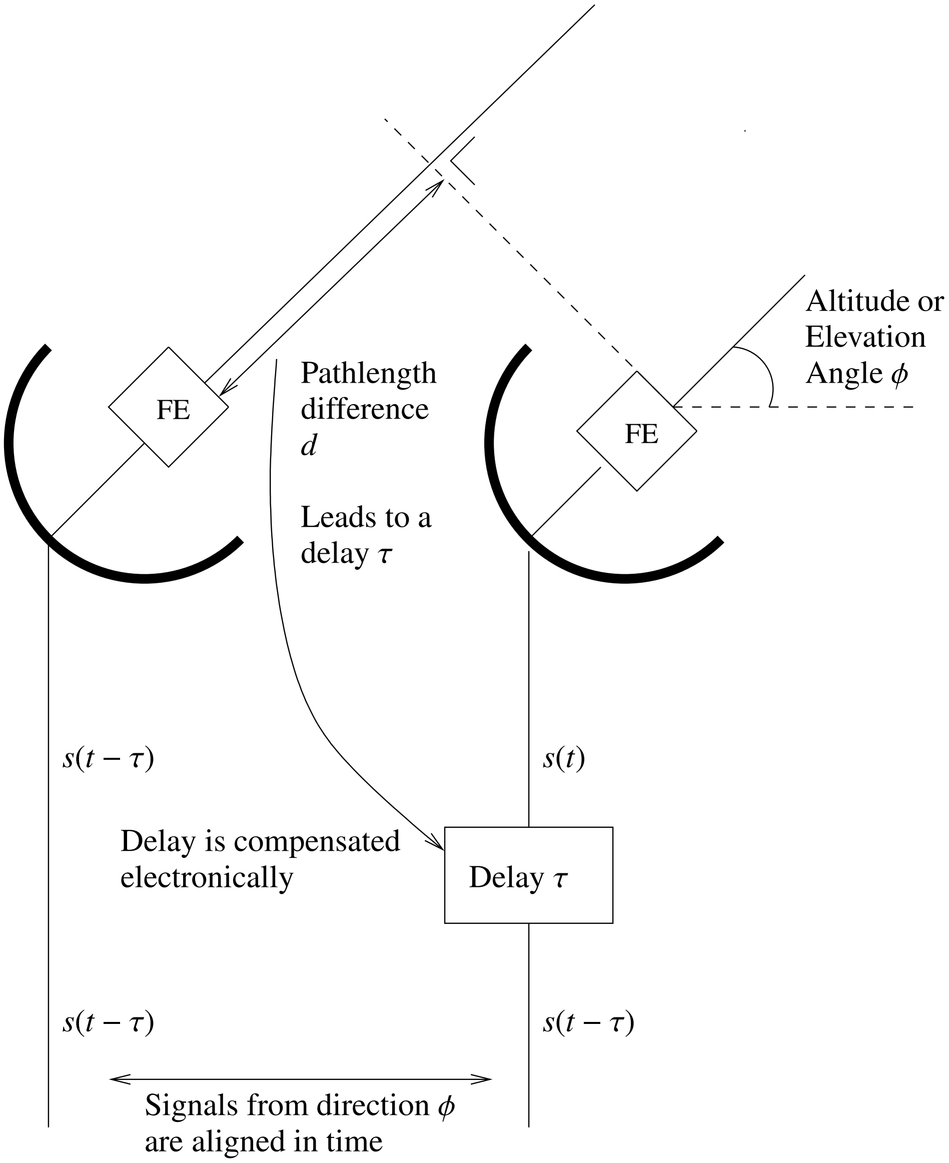


Figure 7.6: Delay difference: cause and compensation

with a maximum distance between two telescopes of approximately 3 km, this leads to maximum cable lengths of 2.5 km.

Nowadays, the compensation is done electronically. In, for example, the WSRT, a combination of techniques is used. A fine-grain delay compensation is realized by shifting the sampling clock of Analog-to-Digital converters for one of the telescopes. A coarse-grain delay compensation is realized by variable-size digital memory buffers. For compensating 3 km of pathlength difference, given a sampling clock of 40 MHz, the required memory equals 400 samples. The sampling clock can be shifted in steps of 195 ps. The accuracy for compensation of pathlength differences therefore equals 6 cm. In the remaining part of this chapter we will assume ideal delay-compensation in multiple-dish telescopes and delay compensation will not be indicated in the remaining block diagrams in this chapter. After delay compensation, the signals from direction ϕ are aligned in time.

There are two different techniques to combine signals from the different dishes within a multiple-dish telescope. First, the signals (telescopes) are grouped into pairs. Each pair is used to determine the signal of the celestial object. The resulting signals from pairs of antennas are processed furtheron, for example, to make maps of the celestial object. This technique is called interferometry. In our analysis, we will restrict ourselves to the analysis of the first stage, the processing of signals from two telescopes.

Second, when a technique is used where the signals from more than two Radio Telescopes are summed, the telescope acts as a phased array. Both techniques, interferometry and phased arrays, will be explained briefly below.

7.3.2.2 Interferometry

When interferometry is used to combine the signals from two telescopes, there are many options. Two basic receiver types are the summing- or simple interferometer and the correlation interferometer. Both are illustrated in figure 7.7:

In the simple interferometer, signals from the two telescopes are summed by an Adder and a Square Law Detector (SLD) produces an output signal which is proportional to the power of the sumsignal. The resulting sum signal consists of two components: the astronomical signal and the sum of the receiver noise introduced by the two front-ends.

In the correlation interferometer, the signals from the two telescopes are multiplied. Because the receiver noise of the two telescopes is uncorrelated, the average value of the output of the multiplier is not affected by the receiver noise and is therefore proportional to the power of the astronomical signal. Note that this is a fundamental difference with the simple interferometer: if the receiver noise in a simple interferometer increases, the average value of the output of the SLD increases.

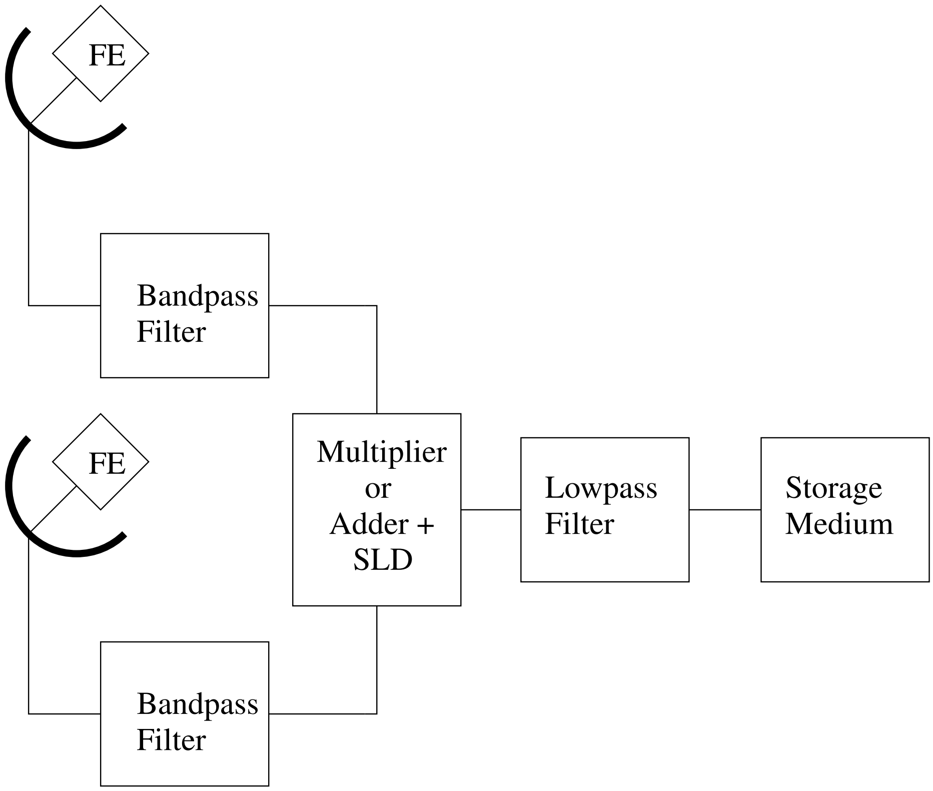


Figure 7.7: Block diagram of an interferometer

The sensitivities for the two interferometer types differ. We introduce the sensitivity constant K_s in expression 7.4 to differentiate the sensitivities of different configurations:

$$\Delta S_{min} = \frac{2kK_s}{\sqrt{\Delta\nu\, t\, N}} \cdot \frac{T_{sys}}{A_e} \tag{7.7}$$

The sensitivity constant K_s depends on the interferometer type that is used. The values are (from [81], p. 258):

- Single dish receiver: $K_s = 1$.
- Simple interferometer: $K_s = \frac{1}{2}$.
- Correlation interferometer: $K_s = \frac{1}{\sqrt{2}}$.

The sensitivity for a single dish receiver, using an SLD is indicated in expressions 7.4 and 7.5 and we use this sensitivity as a reference, leading to $K_s = 1$. To explain the sentitivity constant for other types of receivers, we have to realize that the sensitivity indicates the increase of the flux density that can be detected. This value corresponds with the RMS noise level at the output of the SLD (for the single dish receiver and the simple interferometer) or at the output of the multiplier (for the correlation interferometer). For the simple interferometer, we assume that the telescopes, of which the signals are summed, are identical. Within the two front-ends, statistical independent noise is produced. The adder sums the correlated signals from a radio source and the noise signals from the two front-ends. Compared with the single-dish receiver, the power of the noise, introduced by the receiver, is two times as high (summation of power) and the power of the radio sources is four times as high (summation of RMS values). So, if the DC level of the output of the SLD has to make an equal step for both the single dish receiver and the simple interferometer, the correlated power for the simple interferometer has to increase only a factor $\frac{1}{4}$ compared to the single dish receiver. However, the RMS level of the noise is two times as large. So, to make a step equal to the RMS value at the output of the SLD, the increase of correlated power in case of a simple interferometer has to be $\frac{1}{4} \cdot 2$ times the power increase in case of a single telescope. This leads to the sensitivity constant $K_s = \frac{1}{2}$, for a simple interferometer.

If we compare the correlation interferometer with the single dish receiver, we have to realize that the back-ends in both situations are very much alike. If the SLD, in case of the single-dish receiver, is realized by a multiplier, the only difference is that, in the single dish receiver, the two inputs are connected to the same antenna signals and in the correlation interferometer, the inputs are connected to two different antenna signals. We assume that these signals originate from two identical telescopes. So, an increase of the flux of a radio source leads to an equal increase of the DC component at the output of the multiplier (correlation interferometer) and at the output of the SLD (single-dish receiver). However, the RMS level at the output of the correlation interferometer is $\sqrt{2}$ times as low as the RMS level at the output of the single-dish receiver. This is only valid in case the power, received from the radio source, is small compared to the power produced in the front-ends (see [81], p. 255). This leads to a sensitivity constant $K_s = \frac{1}{\sqrt{2}}$ for the correlation interferometer.

Interferometers lead to smaller minimum detectable flux densities which indicate increased sensitivity. A simple interferometer has a higher sensitivity than a correlation interferometer. So, in theory a simple interferometer would be favorable. In practice, however, a correlation interferometer is preferred because it is less sensitive to increased receiver noise levels in the individual telescopes.

Both simple- and correlation interferometers give an estimate of the power, received from a certain direction. The two antennas of an interferometer are separated by a specific distance. Different antennas (can) have different distances and orientation in the hypothetical surface which connects the front-ends of the different antennas. If there are K antennas within a multiple-dish telescope, there are $K(K-1)/2$ different combinations of two antennas, each combination giving an estimate of the power received from a certain direction. This set of values for different distances and orientations can be used to make maps of celestial object. It is beyond the scope of this thesis to go into the details of this imaging process. For the interested reader we refer to [40].

Several multiple-dish Radio Telescopes have been built. The WSRT has already been mentioned but other examples are the Very Large Array (VLA) in the US, the Australia Telescope Compact Array and the Giant Metrewave Radio Telescope in India. The maximum distances between two telescopes are still relatively small for these telescopes. The largest distance is 36 kilometers for the VLA. Much larger baselines (higher resolutions) can be obtained if signals from telescopes much further apart are combined. This special form of interferometry is called Very Large Baseline Interferometry (VLBI). Several networks exist, for example the European VLBI Network (EVN), the Multi Element Radio Linked Interferometer Network (MERLIN) and the Very Long Baseline Array (VLBA). Multiple-dish telescopes can be part of a VLBI network. The multiple-dish telescope then generally acts as a phased array, producing a single signal to be used within the VLBI network.

7.3.2.3 Phased Arrays

In some applications of Radio Telescopes, imaging (making maps of celestial objects) is not of prime interest. An example has already been mentioned: VLBI. Another prominent example is the observation of pulsars (see [44]). Pulsars are neutron stars which rotate rapidly while emitting a bundle of electro-magnetic radiation. The effect on a single-dish Radio telescope is a very short increase of received power. Typical frequencies for these pulses are between 0.1 Hz and 650 Hz. The power of these pulses can be below the receiver noise of a single antenna. A third application of Phased Arrays within Radio Astronomy is Compound Interferometry. In Compound Interferometry, the telescopes within a multiple-dish telescope are divided into two or more subsets called subarrays. A subarray acts as one phased array and the signals within a subarray are summed. The different sumsignals can be connected to an interferometer. Compound Interferometry is an option to reduce the number of inputs to an interferometer. This way, the data-processing capacity can be used to obtain a higher spectral resolution.

In general, the SNR at the output of a detector of a multiple dish telescope, can be increased by summing the signals from more antennas. An array of telescopes, where the signals from the individual telescopes are summed after path-length compensation, is called a phased array. In Radio Astronomy, a phased array is often referred to as a Tied Array. Because of an increased SNR after summing signals from multiple antennas, the signals used for VLBI and Compound Interferometry have lower noise contributions and pulsars can be detected more accurately.

A phased array is, in principle, a generalization of the simple interferometer. Instead of summing the signals from two telescopes, the signals from K telescopes are summed. This is illustrated in figure 7.8. The signal of interest is equal for all telescopes (is correlated) and the system noise differs for all telescopes (is uncorrelated). The signal of interest adds coherently which implies that signal *voltages* are added. The system noise adds incoherently which means that signal *powers* are summed. The effect on the SNR is discussed below.
The signal S (see figure 7.8) with power σ_S^2, is received by all the telescopes. In the first stages of the front-end, noise is added to this signal. This is schematically indicated by the addition of the noise signals N_1, N_2 to N_M. In all the front-ends, the added noise power is assumed to be equal ($\sigma_{N_i}^2 = \sigma_N^2, i = 1, ..., M$) and it is assumed that there is no correlation between the noise signals. The resulting signals are summed. This can be done at RF frequencies but in many implementations, the signals are downconverted to IF or baseband frequencies after which they are summed. In the summation process, the standard deviation of the signal becomes $M\sigma_S$ so the power of the signal becomes $M^2\sigma_S^2$. Because the noise signals are uncorrelated, the noise power equals $M\sigma_N^2$. The SNR of the resulting sum signal is therefore $M^2/M = M$ times as high as the SNR of a single-dish receiver. Summarizing, the SNR of the sumsignal equals:

$$SNR = \frac{M^2\sigma_S^2}{\sigma_{N_1}^2 + \sigma_{N_2}^2 + ... + \sigma_{N_M}^2} = M \cdot \frac{\sigma_S^2}{\sigma_N^2} \tag{7.8}$$

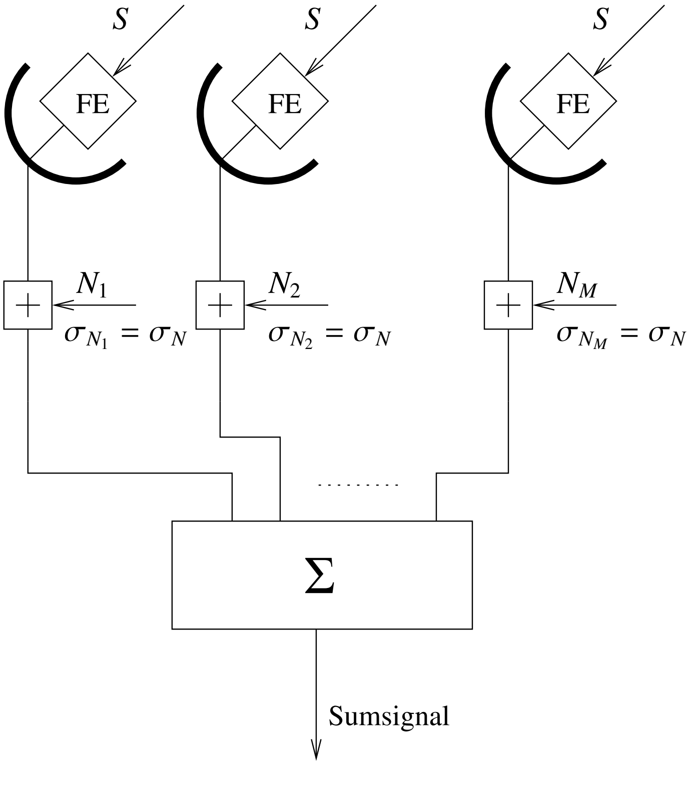

S
S
S
FE
FE
FE
N_1
$\sigma_{N_1} = \sigma_N$
N_2
$\sigma_{N_2} = \sigma_N$
N_M
$\sigma_{N_M} = \sigma_N$
+
+
+
....
Σ
Sumsignal

Figure 7.8: Model for SNR enhancement in a phased array

Chapter 8

A-D Codesign of a Tied Array Adder

In this chapter, we will describe the Analog-Digital Codesign of a Tied Array Adder using coarse quantization. We will start with a brief description of the Westerbork Synthesis Radio Telescope (WSRT). After that we will describe two iterations of the codesign process. In the first iteration, a completely analog solution is briefly described. In the second iteration, a digital solution will be presented.

8.1 Description of the WSRT

Around 1990, ASTRON (the Netherlands Foundation for Research in Astronomy) started the development of new front-ends and a new back-end for the WSRT. The back-end was named DZB. It is a Digital(D) Backend(B) where the 'Z' indicates that it is expected to be the last backend that will be built for the WSRT. The WSRT, including the DZB, is expected to be operational until 2015. After that, a new generation radio telescopes will hopefully be available: the Square Kilometer Array (SKA) and its predecessor the Low Frequency Array (LOFAR). A block diagram of the DZB is given in figure 8.1.

As mentioned in section 7.3.1, the WSRT consists of 14 single-dish telescopes. For the design of the DZB, a number of 16 telescopes is used, allowing future expansion of the array with two additional telescopes. These could, for example, be prototypes of elements of a new generation telescope. Within each single-dish telescope, a front-end captures the signal with two different polarizations. The polarization of the electromagnetic waves can be decomposed in two ways. First, the waves can be decomposed into two linearly polarized waves generally referred to as the x- and y-polarizations. Second, a decomposition in circular components

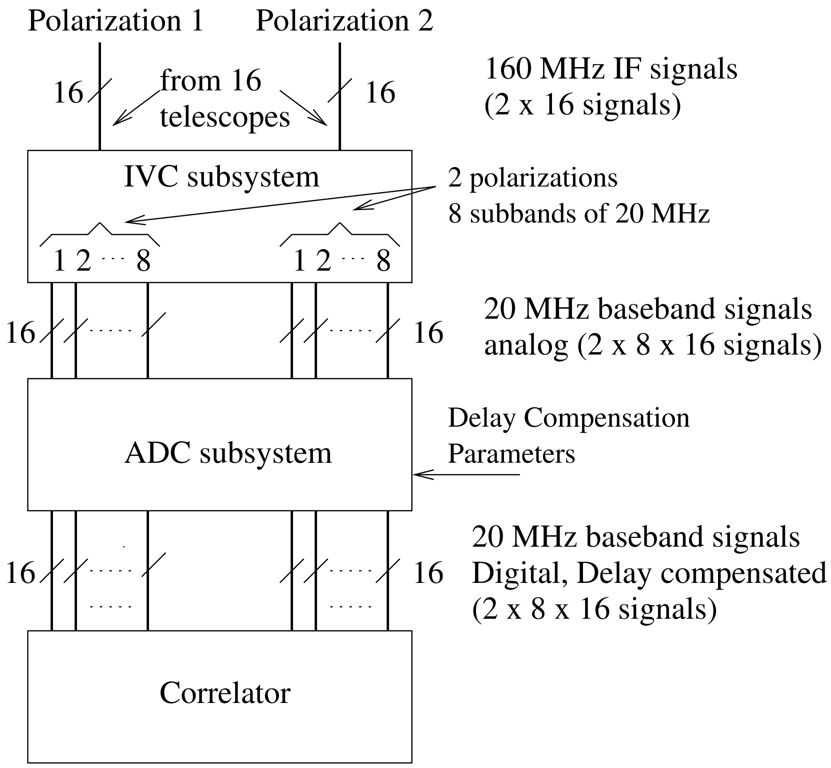


Figure 8.1: Block diagram of the DZB

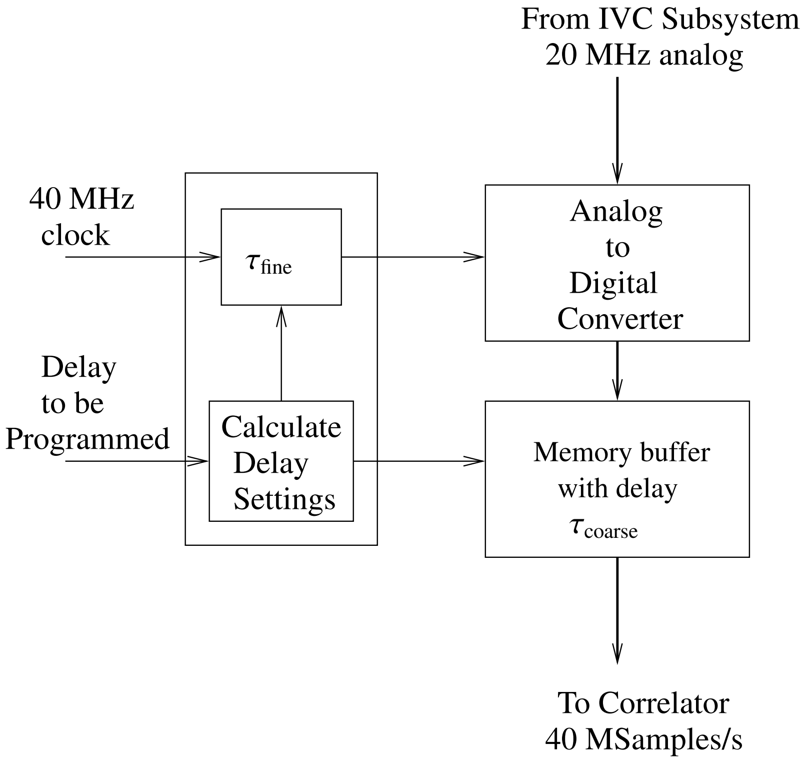


Figure 8.2: Block diagram of the ADC subsystem

(left- and right-polarization) is possible. For the scope of this thesis, polarization simply leads to two signals within the same frequency band for each telescope. The signals from the front-ends occupy a bandwidth of 160 MHz at Intermediate Frequencies (IF). Out of a band of 160 MHz, 8 bands of up to 20 MHz are selected. These 20 MHz bands are frequency converted towards baseband. Baseband signals are alternatively called 'Video' signals. Selection and downconversion are executed by the IF-to-Video Conversion subsystem (IVC subsystem). The outputs of the IVC subsystem are connected to the Analog-to-Digital Conversion subsystem (ADC subsystem). The task of the ADC subsystem is to convert the analog baseband signal into a digital signal and to compensate for the pathlength differences. This is illustrated in figure 8.2.

Every incoming signal from the IVC subsystem is first converted from the analog domain into the digital domain. The resolution of the quantization is 2-bit: a sign and a magnitude bit. If only the sign bit is used further on in the processing, the quantization effectively is single bit. So, within the ADC subsystem, quantization is always 2-bit and the correlator can decide to either use both bits or only the sign bit. The maximum bandwidth is 20 MHz and the sampling frequency equals 40 MHz. This implies that there is no oversampling in case the full bandwidth of

20 MHz is used.

As illustrated in figure 7.6, pathlength differences between signals at different telescopes have to be compensated. Theoretically, the RF signals can be aligned by appropriate delays in the signal paths. In practice, delay compensation is often realized at baseband, after a mixing stage. The appropriate delays should then be effectuated in the signal paths, at baseband, but also the local oscilator signals, feeding the mixing stages, should have the appropriate delays (= phase shifts). Giving the local oscilator signals the appropriate delays is called 'fringe demodulation'. For the scope of this thesis, fringe demodulation is assumed to be implemented correctly in the IVC subsystem and we will not deal with its details.

As mentioned in section 7.3.2.1, two mechanisms are used to compensate the pathlength differences between individual telescopes at baseband: fine delay and coarse delay. The fine delay mechanism is used to introduce delays that are smaller than one clockcycle of 40 MHz. This is realized by shifting the sampling clock. For the WSRT, the sampling clock can be shifted in steps of 195 ps. To implement delays longer than one clockcycle of 40 MHz, a digital buffer is used which delays the output of the ADC a programmable number of 40 MHz clockcycles. The maximum delay is based on the hypothetical case that the Dwingeloo telescope, a 25 meter single dish telescope, is connected to the WSRT. This leads to a maximum baseline of 30 km. The maximum programmable delay therefore equals 100 μs which leads to a memory size of at least 4k samples. After the AD conversion and pathlength compensation, the digital signals are fed into the Correlator, see figure 8.1. In general, signals which have their origin in the same frequency slot and same polarization are correlated with each other. This leads to 16 sets (2 polarizations and 8 subbands of 20 MHz within a band of 160 MHz) of 16 signals (14 antennas + 2 spare inputs). Within a set, 16 autocorrelation functions and $\frac{16 \cdot 15}{2} = 120$ crosscorrelations can be determined. The DZB backend can determine all the auto- and crosscorrelations (local interferometry).

In the phased array application, the basic operation is to sum signals from different antennas. Before the signals can be summed, the pathlength differences have to be compensated. Within the scope of this thesis, a device that compensates pathlength differences and sums the resulting signal is named: Tied Array Adder (TAA). The output of the TAA is, in practice, connected to an SLD (in this context also named a 'total power receiver') for example for pulsar observations and/or to a correlation interferometer. The correlation interferometer can correlate the output of the TAA with a signal from a telescope located at a far distance (VLBI) or with the output of another TAA (Compound Interferometry). Especially for the VLBI application, an analog representation of the output of the TAA is required.

In this chapter, we will discuss the Analog-Digital Codesign of a TAA. During this design process, we will use the Y-chart approach. In the first iteration, we will analyze an analog solution. The result will merely serve as a reference for the

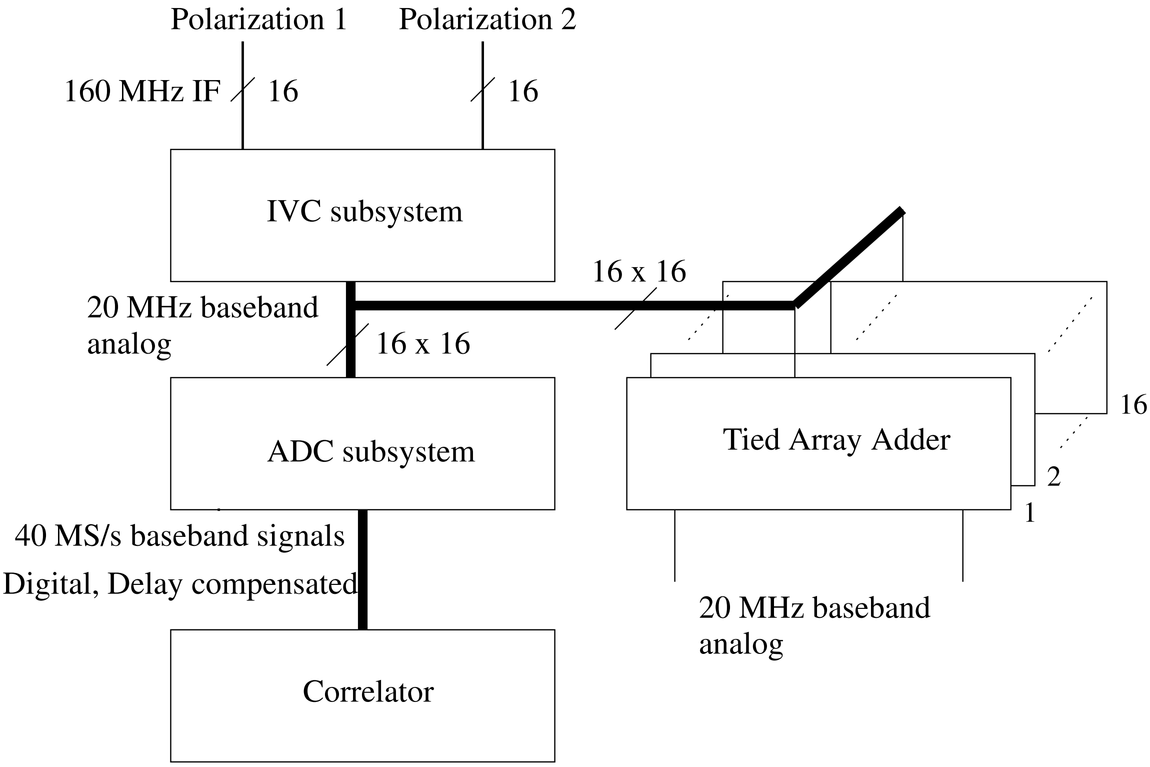


Figure 8.3: Connection of the Analog TAA to the DZB

second iteration where we analyze a digital solution, using coarse quantization.

8.2 The Analog Tied Array Adder

8.2.1 Architecture specification

The first step in the Analog-Digital Codesign of a TAA is the architecture specification. In this step, we basically design an architecture on which the application can be mapped. The starting point will be to realize the TAA completely in the analog domain. The TAA will be connected to the analog outputs of the IVC subsystem as indicated in figure 8.3.

We use 16 identical units where one unit is dedicated to one polarization and one subband. The outputs of the TAA are analog signals and no data converters (DACs or ADCs) will be involved. The advantage of using a completely analog architecture is that no quantization errors are introduced. In the performance analysis of the analog TAA (section 8.2.4) we will further discuss the advantages and disadvantages of the analog architecture.

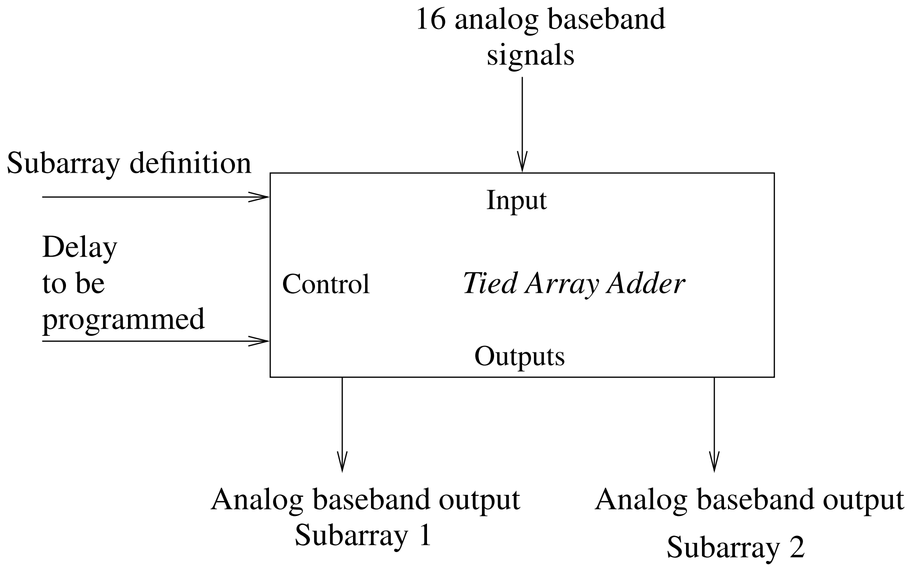

16 analog baseband signals
Subarray definition
Delay to be programmed
Input
Control
Tied Array Adder
Outputs
Analog baseband output Subarray 1
Analog baseband output Subarray 2

Figure 8.4: Input-Output definition of the Analog Tied Array Adder

8.2.2 Functional specification

The Analog TAA is connected to the analog outputs of the IVC subsytem. This way, the selectivity of the IVC subsystem is used to select the proper frequencies and bandwidth. In the remainder of this chapter, we will assume that the signal bandwidth at the input of the TAA always equals 20 MHz. The functionality of the Analog TAA is not affected if bands with smaller width are selected by the IVC subsystem.

The inputs and outputs of the Analog TAA, dedicated to one subband and one polarization, are illustrated in figure 8.4.

Signals from the IVC subsystem originate from the same polarization and same subband of 20 MHz within the total band of 160 MHz. Within an Analog TAA, it should be possible to define two subarrays for every subband and every polarization. Within a subarray, any signal out of the 16 signals available, can be selected. For every subarray, the maximum number of signals that can be summed equals 16. The pathlength difference has to be compensated. Both the subarray definition and the pathlength difference compensation parameters have to be programmed via a digital interface. The outputs of the Analog TAA are the sums of the two subarrays.

An important specification concerns the isolation within a subarray. If a signal is not selected to participate within a subarray, this signal should not be present at all in the output of the subarray. This, however, is an ideal situation. Especially in analog adders, a small fraction of the non-selected signal is still present in the sumsignal. The ratio of the power of this fraction and the power of the original signal is called isolation. It is often specified in dB. The required isolation for the WSRT equals -70 dB.

For the first design iterations, the design of the pathlength compensation and the summation is important. These parts are the specific parts of a TAA. The design of the digital interface to program the delay and to define the subarrays, is a general design issue not specific to a TAA and we will therefore not focus on the design of that part.

8.2.3 Mapping

Within the mapping phase, the functionality, alternatively called application, is mapped onto the architecture. The basic functionality consists of pathlength compensation and summation of signals. Concerning the architecture, we have stated that we will use an analog processing platform, without defining the structure within this platform. So, the mapping basically maps the pathlength compensation and summation of signals onto an analog platform.

8.2.4 Performance analysis

The design of the Analog TAA concerns the design of basically two parts: the pathlength compensation and the summation of signals.

As indicated in section 7.3.2.1, pathlength compensation can be implemented in the analog domain by using waveguides of different lengths. The pathlength difference that has to be compensated, corresponds with a certain length of a waveguide (cable). A standard set of waveguides of different length can be used and selected waveguides can be switched into the signal path. Since the propagation speed in waveguides is in the order of 75 % of the propagation speed in free space, the length of the cables has to be approximately 75 % of the maximum distance between telescopes.

For the summation of analog signals, power combiners can be used. The input of the power combiners are the signals that are compensated for pathlength differences. The isolation that can be obtained by means of power combiners is roughly -30 to -40 dB.

8.2.4.1 Evaluation

Already a very brief design shows some of the great obstacles in the design of an Analog TAA. First, the lengths of the waveguides to compensate for pathlength differences. If, for example, the delay difference caused by a 30 km baseline has to be compensated, waveguides with a length of approximately 22 km have to be used. This becomes very impractical. Second, to obtain an isolation of -70 dB does not seem to be attainable if the signals are summed in the analog domain. For that reason, we have to look for an alternative solution. Refering to the Y-chart approach (see figure 1.11), the basic idea is to change the 'Architecture', making use of the feedback path from 'Performance Numbers' to 'Architecture Specification' within the Y-chart approach. We will use a digital architecture instead of an analog architecture. The analog solution is, however, important to act as a reference for the design of the Digital TAA.

8.3 The Digital Tied Array Adder

The second iteration of the Analog-Digital Codesign of a Tied Array Adder concerns the design of a digital TAA.

8.3.1 Architecture specification

The first step in the second iteration of the Analog-Digital Codesign of a TAA is architecture specification. For the second iteration, we assume that the TAA is realized in the digital domain. An architecture which has the same interfaces as the analog TAA contains ADCs as the first stage, converting the analog outputs from the IVC subsystem into the digital domain. However, the ADC subsystem also contains ADCs, resulting in digital outputs. The basic idea is to use these digital outputs. The resolution of the coarsely quantizing ADCs within the ADC subsystem is one- or two-bits. The performance analysis phase of this iteration (section 8.3.4) will therefore mainly focus on the analysis of the effects of coarse quantization.

One of the consequences of the choice to implement the TAA in the digital domain is that the resulting sumsignal has to be converted from the digital domain into the analog domain. The resulting basic architecture, dedicated to one subband and one polarization, is illustrated in figure 8.5.

8.3.2 Functional specification

Within the second interation of the Analog-Digital Codesign of a TAA, we do not change the functional specification. So, the functional specification, as it is described in section 8.2.2, is still valid. Refering to the Analog-Digital Codesign process using the Y-chart approach (figure 1.11) we do not use the possibility for explicit feedback from the 'Performance Numbers' to the 'Functional Specification'.

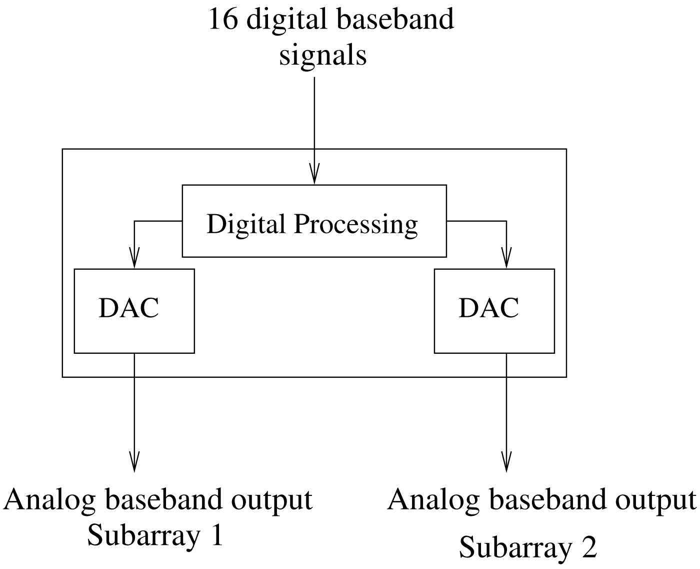

16 digital baseband signals
Digital Processing
DAC
DAC
Analog baseband output Subarray 1
Analog baseband output Subarray 2

Figure 8.5: Basic architecture of the Digital TAA

8.3.3 Mapping

In the mapping phase of the design of a digital TAA, we map the functionality onto the architecture. The basic functionality consists of pathlength compensation and summation. For the digital TAA, the starting point is to realize the pathlength compensation and the summation of signals in the digital domain. The resulting digital sumsignal has to be converted into an analog signal by means of a Digital-to-Analog Converter (DAC).

If we take a closer look at the ADC subsystem of the DZB, we see that in that part of the backend, pathlength compensation is already implemented in the digital domain. Instead of duplicating the digital pathlength compensation part, it is reasonable to assume that it is more efficient to use the pathlength compensation system of the ADC subsystem. So, we map the pathlength compensation onto the ADC subsystem. The summation, however, will be completely mapped onto the digital processing part of the TAA (see figure 8.5). The connection of the TAA to the DZB is illustrated in figure 8.6.

There are 16 sets (2 polarizations times 8 subbands) of 16 signals (14 telescopes + 2 spares). The IVC subsystem selects a frequency band of up to 20 MHz. The ADC quantizes and samples the signals and compensates the pathlength differences. The resulting signals are presented to 16 identical TAAs. On a TAA, two subarrays can be defined and within a subarray the digital signals are summed digitally. The selection of a signal within a subarray is simply implemented by letting the signal participate in the digital summation process or not.

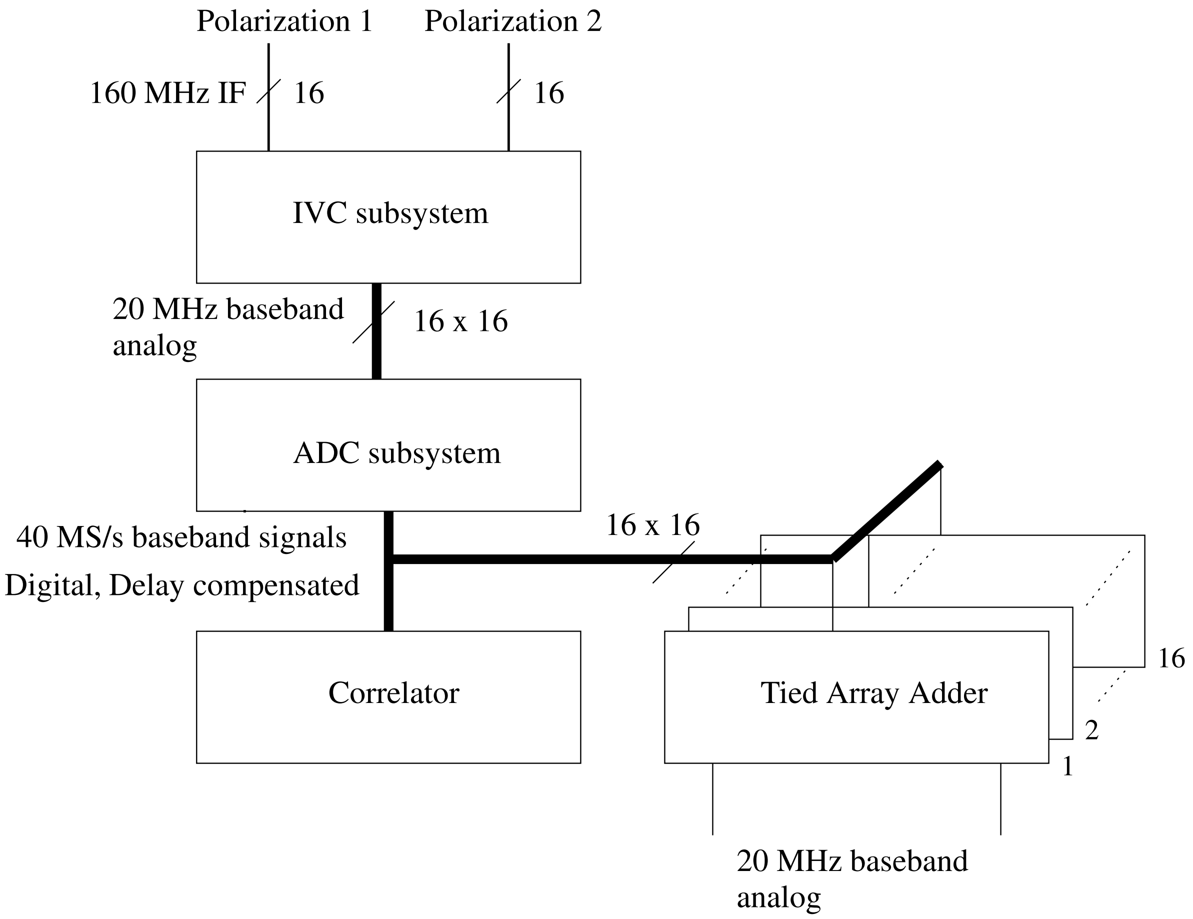


Figure 8.6: Connection of the Digital TAA to the DZB

For each set, there are 2 outputs. All these outputs have to be converted from the digital into the analog domain.

8.3.4 Performance analysis

There are two main advantages of the mapping, described above. First, no pathlength compensation has to be implemented since the pathlength compensation part of the ADC subsystem is used. Second, the signals that have to be summed are in a digital representation and the selection of signals within a subarray is realized digitally as well. The result is that a large isolation is obtained.

An aspect that has not been covered yet is that the quantization in the ADC subsystem is coarse. As mentioned in section 8.1, the analog input from the IVC subsystem is quantized into 2 bits: a sign and magnitude bit. If only the sign bit is used by the TAA, the quantization effectively is single-bit.

Because of the quantization, a quantization error is introduced on every signal. Note that this quantization error is not introduced in the Analog TAA. So, to really compare the Analog TAA with the Digital TAA, the effects of coarse quantization have to be evaluated more thoroughly. The next chapter will be dedicated to this evaluation.

Chapter 9

Evaluation of a Tied Array Adder

The input of the Digital TAA, as described in chapter 8, consists of signals that are quantized into two- or single-bit values. Two- and single-bit quantization are very often referred to as four- and two-level quantization respectively. The number of levels indicates the number of possible values at the output of the quantizer. In a backend that was previously operational at the WSRT, the Digital Continuum Backend (DCB), 3 level quantization was used. In this chapter[1], we will investigate the effects of quantization before summation in case of two-, three- and four-level quantization. A generic model is developed for determining the probability density of the output of a digital adder. This probability density is then used to determine the efficiency of a total power receiver. The sumsignal is also used in correlation interferometers where the sumsignal is correlated with the signal from, for example, a VLBI partner. In this application, the efficiency is expressed in so-called 'degradation factors'. The efficiency of a total power receiver and the degradation factors of a correlation interferometer in combination with a TAA had already been analyzed in [48]. That analysis is restricted to single-bit quantization (two-level quantization) before summation and single-bit quantization after summation. In this chapter, we use the approach, presented in [48] and extend it to three- and four-level quantization before summation and to two-bit correlation after summation. At the end of this chapter, we will reflect on the process of designing a Tied Array Adder, based on the Y-chart approach.

9.1 Introduction

Several authors considered the effects of coarse quantization in an interferometer using a Tied Array ([40], [82] p. 27, [83] p. 181 and [84]). Degradation in sensitivity due to coarse quantization (2, 3 and 4 level) in correlation interferometry

[1] Large parts of this chapter have been published in publication [8]

was considered by many authors and is summarized in [40].

Because of quantization before summation within a digital TAA, quantization noise is added. The signal before quantization consists of two parts: the astronomical signal, which is the signal of interest to the astronomer, and the noise produced by the analog parts of the receiver. By quantization, a third signal is added: the quantization noise. In expressions concerning the SNR of a signal, the astronomical signal is defined as 'Signal' and the sum of the noise produced in the analog parts and the quantization noise is defined as 'Noise'. So, by adding quantization noise, the Signal-to-Noise Ratio (SNR) of the signal is decreased. The decreased SNR of a single signal will lead to a decreased SNR after summation of signals from different telescopes within the TAA. This, on its turn, will lead to decreased SNRs at the output of a total power receiver and at the output of a correlation interferometer, connected to the output of a TAA. The SNR loss is characterized by the efficiency, in case of a total power receiver and degradation, in case of a correlation interferometer. These are coefficients (less than 1) which are multiplied with the SNR, in case an ideal analog TAA is used, to obtain the SNR in case a digital TAA is used. The problem is to find these efficiencies and degradation factors.

9.1.0.1 Generic description of a digital Tied Array system

A generic elaborated block diagram of a digital Tied Array system is given in figure 9.1.

Although this is a generic block diagram, we will adopt the terminology that is used within the DZB of the WSRT. This, however, will not limit the applicability. At the top we see the analog baseband signals Y_1 to Y_M from the multiple dishes, connected to the ADC subsystem. Within the ADC subsystem, the power of a signal is controlled via an Automatic Gain Control (AGC) circuit. The functionality of an AGC will be explained later on. The signal is quantized by an ADC. The TAA sums the signals from different telescopes in the digital domain which results in the digital signal Σ. This signal is converted into the analog domain by means of a DAC. The output of the TAA is the analog signal Y_Σ^d. The subscript Σ indicates that the signal is a sum of signals and the superscript d indicates that the summation is done in the digital domain. The sumsignal can be used by a total power receiver or by a correlation receiver. We assume an analog total power receiver. For the correlation interferometer, there basically are two options. The output of the TAA is correlated with a telescope which is part of the same multple dish receiver. In this case, there is a direct connection from the output of the TAA to an AGC, indicated by a short dashed arrow in figure 9.1. In this case, the AGCs, ADCs and correlator in the bottom part of figure 9.1 are part of the same back-end as the AGC's and ADC's in the top part of the figure.

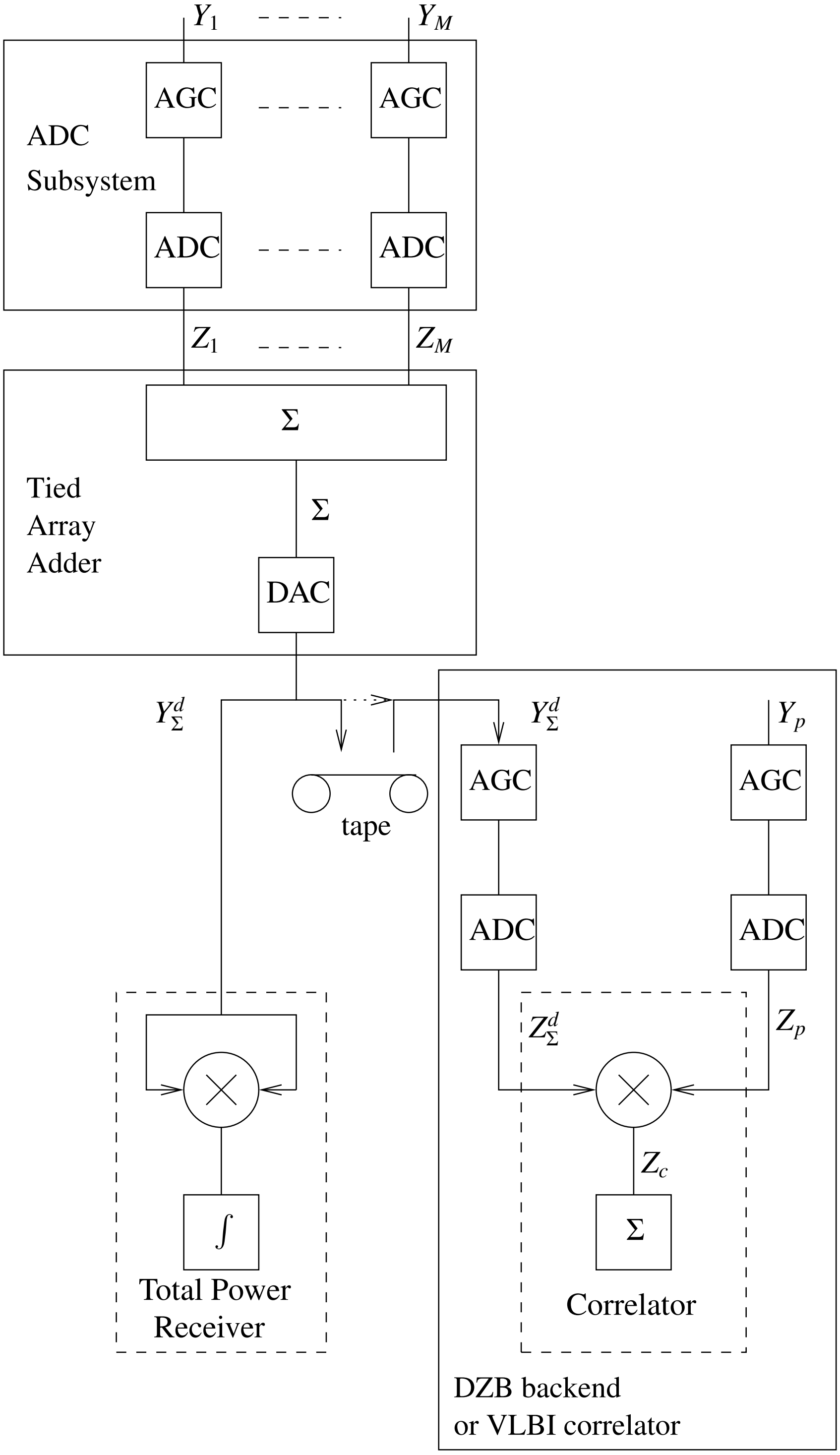


Figure 9.1: Block diagram of a digital Tied Array system

In the second case, there is a large geographical distance between the multiple dish receiver which generates the sumsignal Y_{Σ}^{d}, and the receiver which generates the signal Y_p with which the sumsignal is to be correlated. In this case, the so-called baseline between the multiple dish receiver and the 'partner' receiver is very large and the observation technique is called Very Large Baseline Interferometry (VLBI). In this case, there is no direct connection between the receivers generating the signals Y_{Σ}^{d} and Y_p and the VLBI correlator because the distances are too large. For that reason, the signals are stored on tape and the tapes are transported to the VLBI correlator. By playing the tapes at the site where the VLBI correlator is located, the signals, recorded at the multiple dish receiver and at the partner receiver, can be correlated. This also explains the concatenation of a DAC within the TAA and the AGC and ADC within the VLBI correlator. This sequence is superfluous from a functional point of view. However, the interface at the input of the VLBI correlator is standardized. It requires an analog signal at the input eventhough this is not optimal from a realization point of view.

The analog baseband signals Y_1 to Y_M, Y_{Σ}^{d} and Y_p are connected to the ADCs via an Automatic Gain Control (AGC) circuit. The purpose of the AGC is to maintain a constant RMS level at the input of the ADC. An ADC includes a 2, 3 or 4-level quantizer and a sampler. The combined functionality of the AGC together with the ADC is illustrated in figure 9.2.

We use a single model to describe the three different quantizers. The starting point is the 4-level quantizer. The model of the 4-level quantizer can be used to describe the 3-level and 2-level quantizers as well. The 4-level quantizer divides the input amplitude range into 4 non-overlapping sub-ranges and with each sub-range, a decimal value is associated. The input amplitude range is divided into 4 sub-ranges by 3 reference levels: the '+' level, the '0' level and the '-' level. The '+' and '-' levels are proportional to the RMS input level of the input signal. For the 4-level quantizer, the '+' and '-' levels are at the $+\sigma$- and $-\sigma$- levels respectively, where the power of the signal at the input of an AGC equals σ^2. The '0' level is at the DC level of the input signal. This way, the four sub-ranges are defined. Associated with each sub-range is a value V_i, $i = 1, ..., 4$. If the instantaneous amplitude of the input signal is above the '+' level, the output of the ADC equals V_4. If the input is between the '0' and '+' levels, the output equals V_3. If the input is between the '-' and '0' levels, the output equals V_2 and if the input is below the '-' level, the output of the ADC equals V_1. In our analysis we use the values $V_4 = 3$, $V_3 = 1$, $V_2 = -1$ and $V_1 = -3$ for the 4-level quantizer.

The 3-level and 2-level quantizers are obtained by changing the positions of the reference levels and by assigning other values to V_4 to V_1.

For the 3-level quantizer, the '+' and '-' levels are located at 0.6σ and -0.6σ respectively. Furthermore, $V_4 = 1$, $V_3 = 0$, $V_2 = 0$ and $V_1 = -1$. Effectively, the '0' level is not used because $V_3 = V_2$. Note that we use a decimal representation

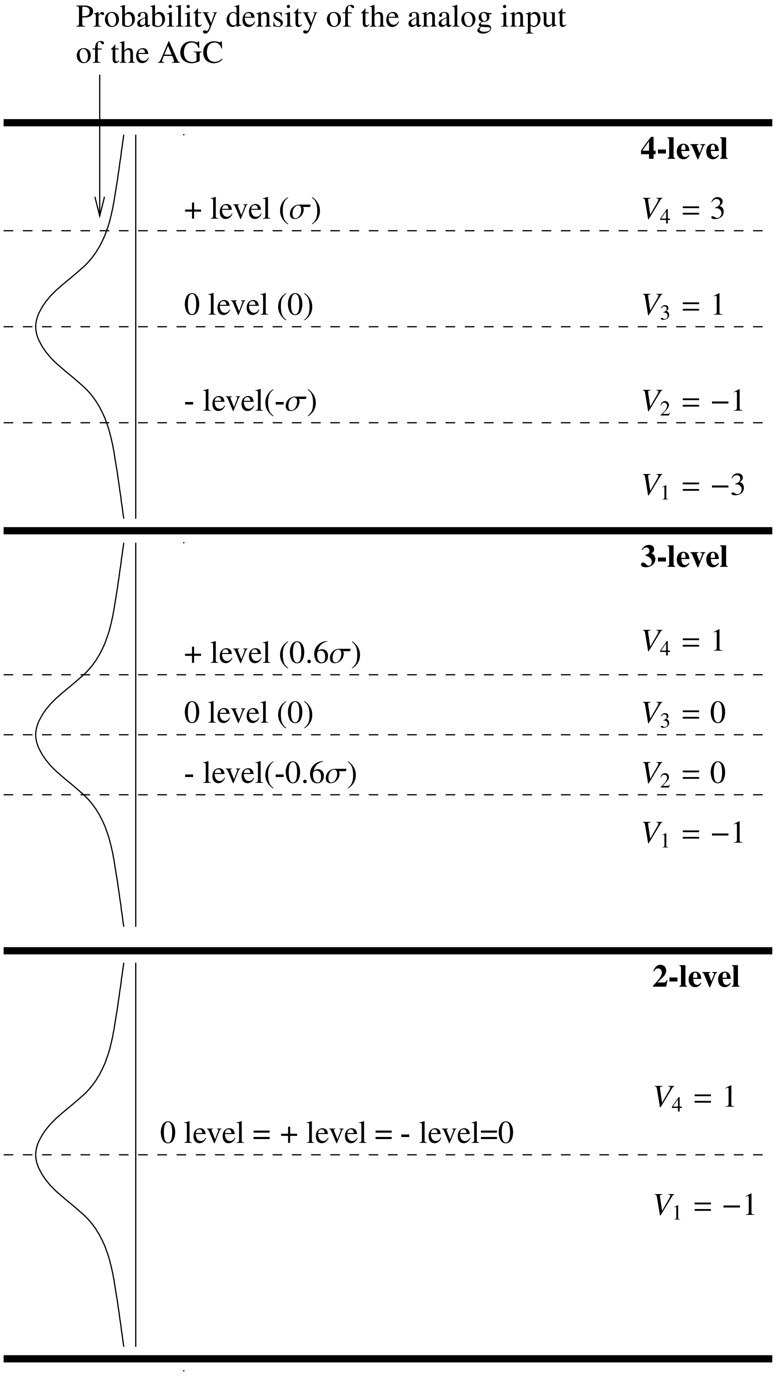


Figure 9.2: Combined functionality of the AGC and the ADC for 4-level, 3-level and 2-level quantization

for the output of the quantizer. When implementing the quantizer, two bits are necessary to encode the three states. From this point of view, 3-level quantization is not optimal.

For 2-level quantization, the '+', '-' and '0' levels are all located at the DC level of the input signal. Because of the location of the reference levels, the output of the ADC consists of a sequence of values V_1 and V_4, more specifically, a sequence of -1 and 1.

Quantization at fixed fractions of the RMS levels of the input is often used in correlation interferometers to optimize the SNR after correlation (see [85], [86] and [87]).

A generalized quantizer has three reference levels: the '+' level at $l\sigma$, the '0' level at 0 and the '-' level at $-l\sigma$. The 4-level quantizer is specified by: $l = 1$, $V_4 = 3$, $V_3 = 1$, $V_2 = -1$ and $V_1 = -3$. The 3-level quantizer is specified by: $l = 0.6$, $V_4 = 1$, $V_3 = 0$, $V_2 = 0$ and $V_1 = -1$. The 2-level quantizer is specified by: $l = 0$, $V_4 = 1$, $V_3 = -$, $V_2 = -$ and $V_1 = -1$, where '$-$' indicates 'don't care'.

After defining the decimal values at the output of different quantizers, we can define the digital summation in the TAA. The summation sums the decimal values of the outputs of the quantizers which results in a decimal value. In a real implementation, this decimal result will also have a binary representation.

The functionality of the DAC within the TAA is to convert the decimal value into a voltage or current which is proportional to the decimal value.

The delay compensation, within the ADC subsystem and between the multiple dish receiver and the VLBI partner, is assumed to be executed correctly and is therefore ignored in this context.

9.2 Probability density of the sumsignal

In this section, we determine the probability density of the sumsignal Y_Σ^d. This distribution is then used to determine the efficiency of the total power receiver. After that, we combine the probability density of Y_Σ^d with the probability density of the signal Y_p to determine the degradation factor of the correlation interferometer. In order to determine the probability density of the analog sumsignal Y_Σ^d, we first determine the probability density of the digital sumsignal Σ via the model depicted in figure 9.3.

The analog antenna signals at the input of the AGCs of the multiple dish receiver, consist of a signal common to all antennas (the correlated part N_0) and noise produced in the analog receiving systems (the uncorrelated part N_i, for $i = 1, ..., M$). N_0 to N_M are modeled as Gaussian noise sources with zero mean. The noise produced in the receiver systems (N_1 to N_M) is assumed to have equal power and without loss of generality we can say that the noise power of N_0, the

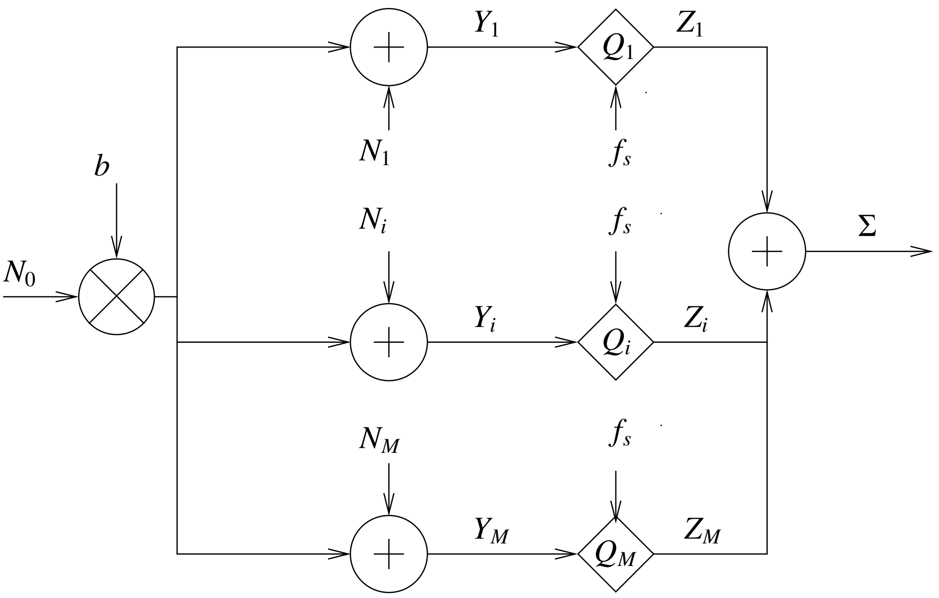


Figure 9.3: Noisemodel of the Tied Array Adder

correlated noise source, equals the power of N_1 to N_M and equals 1. The signals Y_1 to Y_M in figure 9.3 correspond to the signals at the input of the ADC subsystem (see figure 9.1). Within a quantizer Q_i, $i = 1, ..., M$, the functionalities of an AGC and an ADC are combined. The amount of correlation between two arbitrary signals Y_i and Y_j is determined by the multiplication factor b [2]. Any pair of signals out of the set Y_1 and Y_M has a joint probability density function with correlation coefficient r, where $r = b^2/(b^2 + 1)$, see [88], p. 484.

In case the noise sources N_i ($i = 1, ..., M$) have equal power, the signals N_i have equal statistical properties and will further be denoted as N. The same is valid for Y_i and for Z_i, the quantized version of Y_i. Y_i and Z_i will be denoted as Y and Z. N_0, N, Y and Z have instantaneous values n_0, n, y and z.

First we will determine the probability that the output of a single quantizer has the value V_1. After that, the probabilities for the output values V_2, V_3 and V_4 can be obtained easily. These probabilities are then used to determine the probability density of the sumsignal Σ.

The output Z of a quantizer equals V_1 if the corresponding input Y is below the '-' level (see figure 9.2): $z = V_1$ if $y < -l\sigma$. Because $y = n + bn_0$, the following is also valid: $z = V_1$ if $n + bn_0 < -l\sigma$. Suppose we know n_0, we can rewrite this

[2] In the previous chapters and in appendix D, b is used to indicate the 'resolution' or 'number of bits' of an ADC. In this chapter, we will use b to indicate the RMS value of the correlated part of two signals and the resolution of ADCs is indicated by the expressions '2-level', '3-level' and '4-level'.

into: $z = V_1$ if $n < -bn_0 - l\sigma$. n represents an instantaneous value of a Gaussian noise source which has a normal distribution. We can use this to determine the conditional probability that z equals V_1:

$$\begin{aligned} P_Z(z = V_1 \mid n_0) &= \int_{-\infty}^{-bn_0 - l\sigma} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}n^2} dn \\ &= \int_{-\infty}^{-l\sigma} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(n-bn_0)^2} dn \end{aligned} \quad (9.1)$$

In a similar way we can deduct:

$$P_Z(z = V_2 \mid n_0) = \int_{-l\sigma}^{0} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(n-bn_0)^2} dn \quad (9.2)$$

$$P_Z(z = V_3 \mid n_0) = \int_{0}^{l\sigma} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(n-bn_0)^2} dn \quad (9.3)$$

$$P_Z(z = V_4 \mid n_0) = \int_{l\sigma}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(n-bn_0)^2} dn \quad (9.4)$$

The conditional probability density function of the output of a single quantizer can then be described as:

$$\begin{aligned} P_Z(z \mid n_0) &= P_Z(z = V_1 \mid n_0)\, \delta(z - V_1) \\ &+ P_Z(z = V_2 \mid n_0)\, \delta(z - V_2) \\ &+ P_Z(z = V_3 \mid n_0)\, \delta(z - V_3) \\ &+ P_Z(z = V_4 \mid n_0)\, \delta(z - V_4) \end{aligned} \quad (9.5)$$

The multiplication with the delta functions δ forces the density to have a finite probability at the four different output values of a single quantizer. Using the conditional probability densities of the different quantizers, the conditional probability density $P_\Sigma(s \mid n_0)$ of the sumsignal Σ can be determined. Suppose, for $i = 1, 2, 3$ and 4, k_i is the number of quantizers which have a decimal output value V_i and suppose the decimal value of the signal after summation Σ equals s, then the following condition is met:

$$V_1 k_1 + V_2 k_2 + V_3 k_3 + V_4 k_4 = s \quad (9.6)$$

Because the total number of antennas equals M:

$$k_1 + k_2 + k_3 + k_4 = M \tag{9.7}$$

From the set of M antennas, there is a subset of k_i antennas having the value V_i. There are $\begin{pmatrix} M \\ k_i \end{pmatrix}$ possible subsets with k_i antennas. The probability that there are k_i antennas having the output value V_i equals $P_Z(z = V_i \mid n_0)^{k_i}$. The conditional probability density of the sumsignal therefore becomes:

$$\begin{aligned} P_\Sigma(s \mid n_0) = & \sum_{k_1,k_2,k_3,k_4} \begin{pmatrix} M \\ k_1 \end{pmatrix} \begin{pmatrix} M \\ k_2 \end{pmatrix} \begin{pmatrix} M \\ k_3 \end{pmatrix} \begin{pmatrix} M \\ k_4 \end{pmatrix} \\ & P_Z(z = V_1 \mid n_0)^{k_1} \cdot P_Z(z = V_2 \mid n_0)^{k_2} \cdot \\ & P_Z(z = V_3 \mid n_0)^{k_3} \cdot P_Z(z = V_4 \mid n_0)^{k_4} \end{aligned} \tag{9.8}$$

$$\begin{gathered} \text{and for all } k_1, k_2, k_3, k_4, \\ V_1k_1 + V_2k_2 + V_3k_3 + V_4k_4 = s, \text{and} \\ k_1 + k_2 + k_3 + k_4 = M \end{gathered}$$

The source N_0 is a Gaussian noise source with power equal to 1, thus the expression for the (unconditional) probability density function of the digital sumsignal $P_\Sigma(s)$ equals:

$$P_\Sigma(s) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}n_0^2} \cdot P_\Sigma(s \mid n_0) dn_0 \tag{9.9}$$

To give a few examples, this expression is used to generate the probability density of the sumsignal Σ for the case of 14 antenna's, 4 level addition and $r = 0.01$, $r = 0.5$ and $r = 0.99$ (b=0.101, b=1 and b=9.95 resp.). The results are presented in figure 9.4.

We see that for a small correlation coefficient, the probability density of the sumsignal very much resembles the probability density of a Gaussian signal. When increasing the correlation coefficient r, the 'tails' of the probability density are cut. For a correlation coefficient close to one, the probability density resembles the probability density of the output of a single ADC multiplied by 14.

From figure 9.1, we see that the digital sumsignal is converted to an analog signal by a DAC. We assume that this is an ideal component and therefore:

$$P_{Y_\Sigma^d}(s \mid n_0) = P_\Sigma(s \mid n_0) \tag{9.10}$$

$$P_{Y_\Sigma^d}(s) = P_\Sigma(s) \tag{9.11}$$

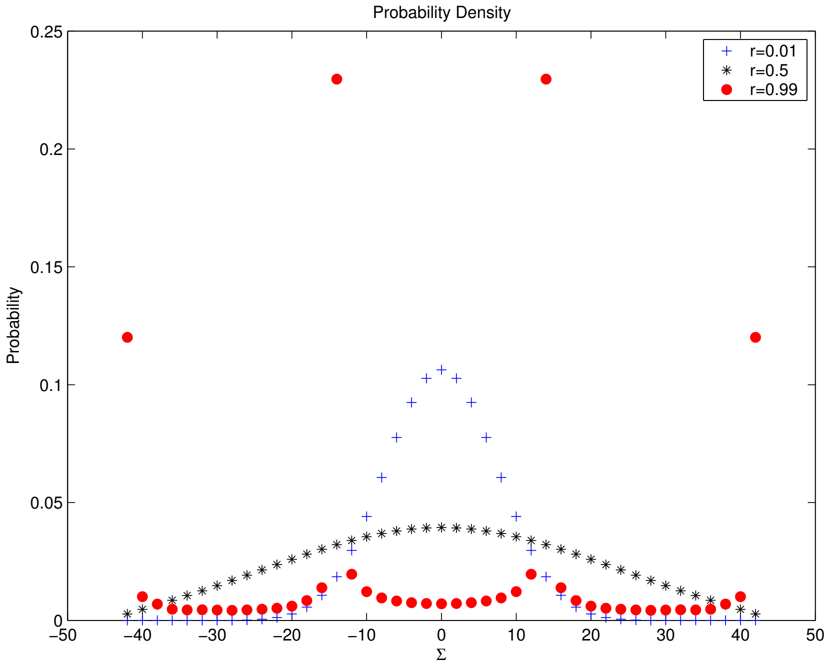


Figure 9.4: Probability density of the sumsignal Σ using 14 antenna's and 4 level addition

In the next section, these probability functions will be used to determine the efficiency in case the sumsignal is connected to a total power receiver and to determine the degradation in case the sumsignal is connected to a correlation interferometer.

9.3 Total Power Detection

9.3.1 The Total Power Detection Efficiency

In this section, the efficiency (η^{tpd}, where the superscript *tpd* indicates 'Total Power Detector') of the digital TAA, connected to a total power receiver, is determined. The efficiency is defined as the ratio of the SNR in case of a digital TAA ($SNR^{d,tpd}$) and the SNR in case of an analog TAA ($SNR^{a,tpd}$):

$$\eta^{tpd} = \frac{SNR^{d,tpd}}{SNR^{a,tpd}} \tag{9.12}$$

A general expression for the SNR can be deduced using expression 7.2, giving the observed flux density and expression 7.4, giving the RMS flux density. We repeat these expressions:

$$S_0 = \frac{2kT_A}{A_e} \tag{7.2}$$

where

$$\begin{aligned} S_0 &= \text{fluxdensity (Watt per Hz per m}^2\text{)} \\ k &= \text{Boltzmann's constant (= 1.38 }10^{-23}\text{ Watt Hz}^{-1}\text{ K}^{-1}\text{)} \\ T_A &= \text{antenna temperature (K)} \\ A_e &= \text{effective antenna aperture (m}^2\text{)} \end{aligned}$$

$$\Delta S_{rms} = \frac{2kK_s}{\sqrt{\Delta\nu\, t\, N}} \cdot \frac{T_{sys}}{A_e} \tag{7.4}$$

where

$$\begin{aligned} \Delta S_{rms} &= \text{standard deviation(Watt per Hz per m}^2\text{)} \\ K_s &= \text{sensitivity constant (dimensionless)} \\ \Delta\nu &= \text{predetection bandwidth (Hz)} \\ t &= \text{postdetection integration time (s)} \\ N &= \text{number of records averaged (dimensionless)} \\ T_{sys} &= \text{system temperature, (K)} \end{aligned}$$

The SNR is then generally defined as:

$$SNR = \frac{S_0}{\Delta S_{rms}} = \frac{T_A}{T_{sys}} \frac{\sqrt{\Delta\nu\, N\, t}}{K_s} \tag{9.13}$$

The problem with a total power receiver is that it is not possible to determine T_A independently from T_{sys} because $T_{sys} = T_A + T_R$ (expression 7.3). For that reason, at least two measurements are used to determine the SNR. First, T_{sys} is determined by pointing all antennas within the Tied Array at a 'silent' part of the sky (off-source measurement). Ideally, in this case there is no correlation between the signals from individual telescopes. Refering to figure 9.3, this situation is modeled by $b = 0$. Because the off-source measurement corresponds with the case that $b = 0$, the corresponding temperature is denoted as $T_{sys}(0)$. Second, the telescopes are pointed towards the object of interest (on-source measurement). In general, the correlation between the signals from two telescopes is small which corresponds with small values of b. The measured temperature is denoted as $T_{sys}(b)$. The difference between the off-source and on-source measurements indicates the antenna temperature T_A. The resulting expression for the SNR becomes:

$$SNR = \frac{T_{sys}(b) - T_{sys}(0)}{T_{sys}(0)} \frac{\sqrt{\Delta\nu\, N\, t}}{K_s} \tag{9.14}$$

$T_{sys}(b) \cdot \sqrt{\Delta\nu}$ is proportional to the average value of the Square Law Detector of the total power receiver, also for $b = 0$. Another term for 'average value' is 'first order moment', so $T_{sys}(b) \cdot \sqrt{\Delta\nu}$ is proportional to the first order moment of the squared output of the TAA. In case of a digital TAA, this first order moment α_1^d is determined by the probability density $P_{Y_\Sigma^d}$:

$$\alpha_1^d(b) = \sum_{s=-\infty}^{\infty} s^2 \cdot P_{Y_\Sigma^d}(s) \tag{9.15}$$

The final expression for the SNR in case of a digital TAA becomes:

$$SNR^{d,tpd} = \frac{\alpha_1^d(b) - \alpha_1^d(0)}{\alpha_1^d(0)} \frac{\sqrt{N\, t}}{K_s} \tag{9.16}$$

This value ($SNR^{d,tpd}$) can be related to the SNR of an analog adding system $SNR^{a,tpd}$, where $SNR^{a,tpd}$ has been determined under the same conditions. In case of analog summation, the block diagram shown in figure 9.5 is used.

The difference between this block diagram and the block diagram for a digital TAA, presented in figure 9.1, is that there is no conversion from the analog domain into the digital domain before summation and no conversion from the digital domain into the analog domain after summation. Consequently, the summation is done completely in the analog domain. The task of the AGCs before summation is to guarantee that all signals have equal power before summation.

To determine the SNR of an analog adding system, the probability density of the signal Y_Σ^a (where the superscript a indicates the analog adding system) at the output of the analog adding system has to be determined. In total there are M telescopes where the signal from each telescope consists of two parts: a part which is correlated with other telescopes (the astronomical signal) and a part which is uncorrelated with other telescopes (receiver noise). The power of the correlated part equals b^2, the power of the receiver noise equals 1. In the analog TAA, the signals from M telescopes are summed where the *voltages* of the correlated signals and the *powers* of the uncorrelated signals are summed. The power at the output of the analog TAA therefore equals $M + (Mb)^2$. Because all signals have a Gaussian probability density, the probability density of the output of the analog

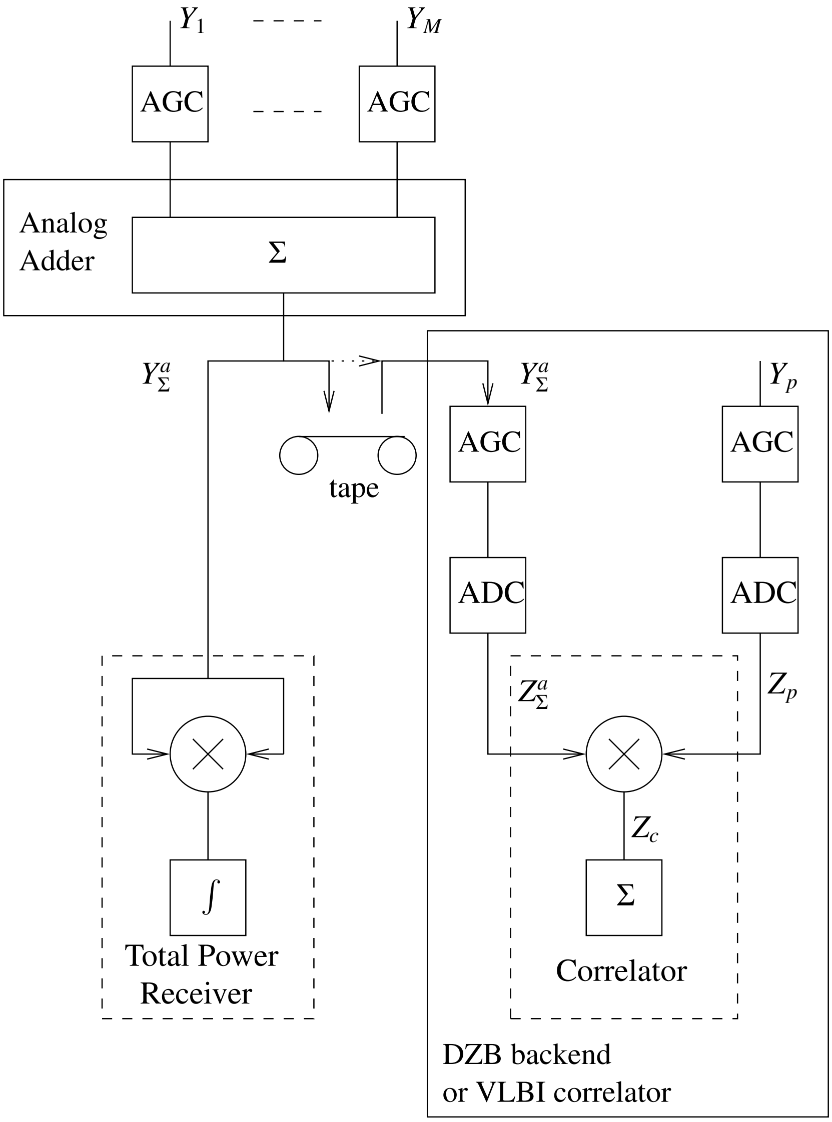


Figure 9.5: Block diagram of an Analog Tied Array system

Table 9.1: Total power receiver Efficiency η^{tpd}

Number of antennas	2-level	3-level	4-level
1	0.000	0.000	0.000
2	0.45	0.482	0.5
3	0.52	0.602	0.637
4	0.552	0.658	0.701
5	0.57	0.69	0.739
6	0.582	0.711	0.764
7	0.59	0.726	0.781
8	0.597	0.737	0.794
9	0.602	0.745	0.804
10	0.605	0.752	0.812
11	0.609	0.758	0.819
12	0.611	0.762	0.824
13	0.614	0.766	0.829
14	0.616	0.77	0.831

TAA is Gaussian as well. The resulting probability density becomes:

$$P_{Y^a_\Sigma}(s) = \frac{1}{\sqrt{2\pi(M+(Mb)^2)}} e^{-\frac{1}{2}\frac{s^2}{M+(Mb)^2}} \tag{9.17}$$

The moments and the SNR of the analog adding system are then defined by

$$\alpha^a_i(b) = \int_{s=-\infty}^{\infty} (s^2)^i \cdot P_{Y^a_\Sigma}(s)\, ds \tag{9.18}$$

$$SNR^{a,tpd} = \frac{\alpha^a_1(b) - \alpha^a_1(0)}{\alpha^a_1(0)} \frac{\sqrt{N\,t}}{K_s} \tag{9.19}$$

The ratio of the SNR in case of a digital TAA (expression 9.16) and the SNR in case of an analog TAA (expression 9.19) is defined as the efficiency of the digital adding system (expression 9.12). In table 9.1, the efficiency is given for 2-, 3- and 4-level quantization before summation using a power increase of 2.5 % (b = 0.025) and a small number of telescopes (up to 14 telescopes).
A graphical representation of these values is given in figure 9.6.

The efficiency of the digital TAA increases with an increasing number of telescopes. For the sum of a large number of telescopes, the efficiency appears to

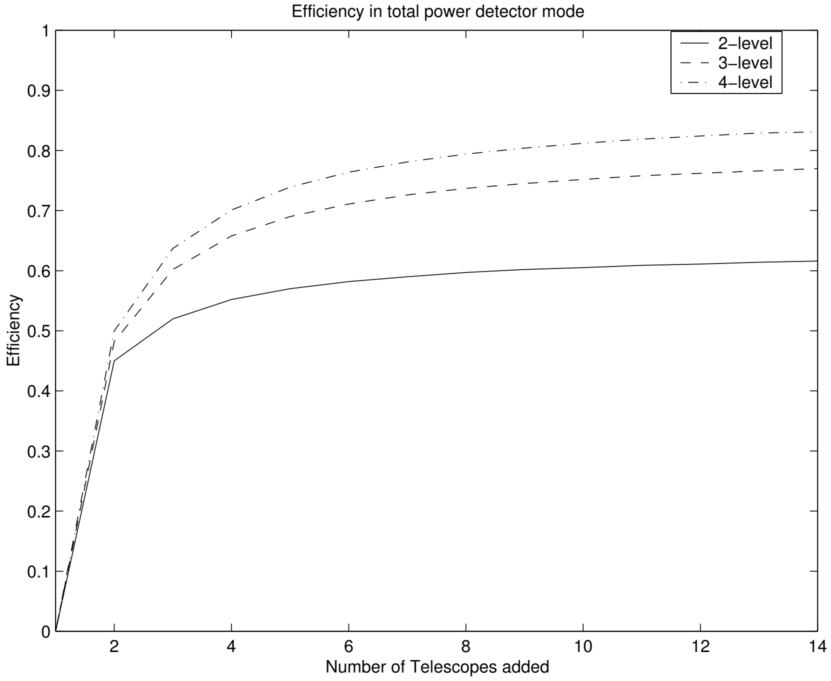


Figure 9.6: Total power receiver Efficiencies for 2-, 3-, and 4-level addition

approximate the well-known factors $2/\pi$, 0.81 and 0.88 for 2-, 3- and 4-level quantization respectively (see [86] and [87]).

The values presented may be useful for calculating the TPD effectivity in, for example, pulsar observations. If 14 antenna-signals are summed using 3-level quantizers, the effective antenna area can be calculated as $14 \cdot 0.77 = 10.78$ times the effective antenna area of a single telescope. This is valid in case of equal receiver temperatures. In case of unequal receiver temperatures the calculation of the effective signal temperature and antenna temperature is not straightforward and is elaborated in appendix G.

The general conclusion is that, because of coarse quantization, the efficiency of a digital TAA degrades significantly.

9.4 The correlation interferometer mode

After determining the efficiency of the digital TAA in combination with a total power receiver, we have to determine the degradation factor in case the TAA output is used by a correlation interferometer. The use of a TAA in a correlation interferometer mode is illustrated in figure 9.1. In the bottom right part it is shown that the digitally summed signal is correlated with another signal. This can be a signal from a VLBI partner but it can also be the output from another TAA (Compound Interferometry) or a signal from a single dish receiver, which is an element of the same multiple dish receiver. We will refer to this signal, however, as the signal from a VLBI partner. In expression 9.10 (together with expression 9.8), the conditional probability density $P_{Y_\Sigma^d}(s \mid n_0)$ of the sumsignal is given. To calculate the SNR at the output of a digital correlator, we have to determine the joint probability density function of the sumsignal Y_Σ^d and the signal Y_p from the VLBI partner (see also figure 9.1). To make use of the previous calculations concerning the sumsignal, we will determine the joint probability density function of the signal Y_p from the VLBI partner and the correlated signal, represented by noise source N_0 (see figure 9.3) . It is assumed that the receiver system of the VLBI partner is identical to the receiver system of a single element of the array. Since N_0 and Y_p are Gaussian noise signals, the joint probability density function $f_{N_0 Y_p}(n_0, y_p)$ is a bivariate normal distribution (see [14], pg. 147) where the correlation coefficient can be defined as r_p, so

$$f_{N_0 Y_p}(n_0, y_p) = \frac{1}{2\pi\sigma_{Y_p}\sigma_{N_0}\sqrt{1-r_p^2}} e^{-\frac{1}{2(1-r_p^2)}\left(\frac{n_0^2}{\sigma_{N_0}^2} - \frac{2r_p n_0 y_p}{\sigma_{Y_p}\sigma_{N_0}} + \frac{y_p^2}{\sigma_{Y_p}^2}\right)} \tag{9.20}$$

$$r_p = \frac{\mathrm{E}(N_0 \cdot Y_p)}{\sigma_{N_0}\sigma_{Y_p}} = \frac{b}{1 \cdot \sqrt{1+b^2}} = \sqrt{r} \tag{9.21}$$

where r equals the correlation coefficient between any pair of signals Y_i and Y_j ($i, j = 1, ..., M$, and $i \neq j$). If we now look at the joint probability density of Y_p and the sumsignal Y_Σ^d and realize that $\sigma_{N_0} = 1$, we see:

$$\begin{aligned} P_{Y_\Sigma^d Y_p}(s, y_p) &= \int_{-\infty}^{\infty} f_{N_0 Y_p}(n_0, y_p) \cdot P_{Y_\Sigma^d}(s \mid n_0)\, dn_0 \\ &= \int_{-\infty}^{\infty} \frac{1}{2\pi\sigma_{Y_p}\sqrt{1-r_p^2}} \cdot \\ &\quad e^{-\frac{1}{2(1-r_p^2)}\left(n_0^2 - \frac{2r_p n_0 y_p}{\sigma_{Y_p}} + \frac{y_p^2}{\sigma_{Y_p}^2}\right)} \cdot P_{Y_\Sigma^d}(s \mid n_0)\, dn_0 \end{aligned} \tag{9.22}$$

This joint probability density will be used in the next section to determine the SNR of a correlation interferometer, using the sumsignal of a digital TAA.

9.4.1 The Degradation Factor

In this section, the Degradation Factor in case of a correlation interferometer will be defined. For this, the SNRs in case of a digital TAA and in case of an analog

TAA are determined.

9.4.1.1 Signal to noise ratio in case of a digital TAA

The analog sumsignal Y_Σ^d is quantized before (VLBI-)correlation (see figure 9.1). The same ADCs as before the summation are assumed and their functionality is illustrated in figure 9.2. The SNR of a digital correlator has been studied by Cooper [86] and Bowers and Klingler [85]. In this document, the approach of Cooper is adopted.

The generalized SNR of a correlator using the sumsignal of a digital TAA is given as follows:

$$SNR^{d,int} = \frac{\mathrm{E}(Z_\Sigma^d \cdot Z_p)}{\sqrt{\mathrm{E}((Z_\Sigma^d \cdot Z_p)^2) - \mathrm{E}(Z_\Sigma^d \cdot Z_p)^2}} \frac{\sqrt{\Delta\nu\, t\, N}}{K_s} \tag{9.23}$$

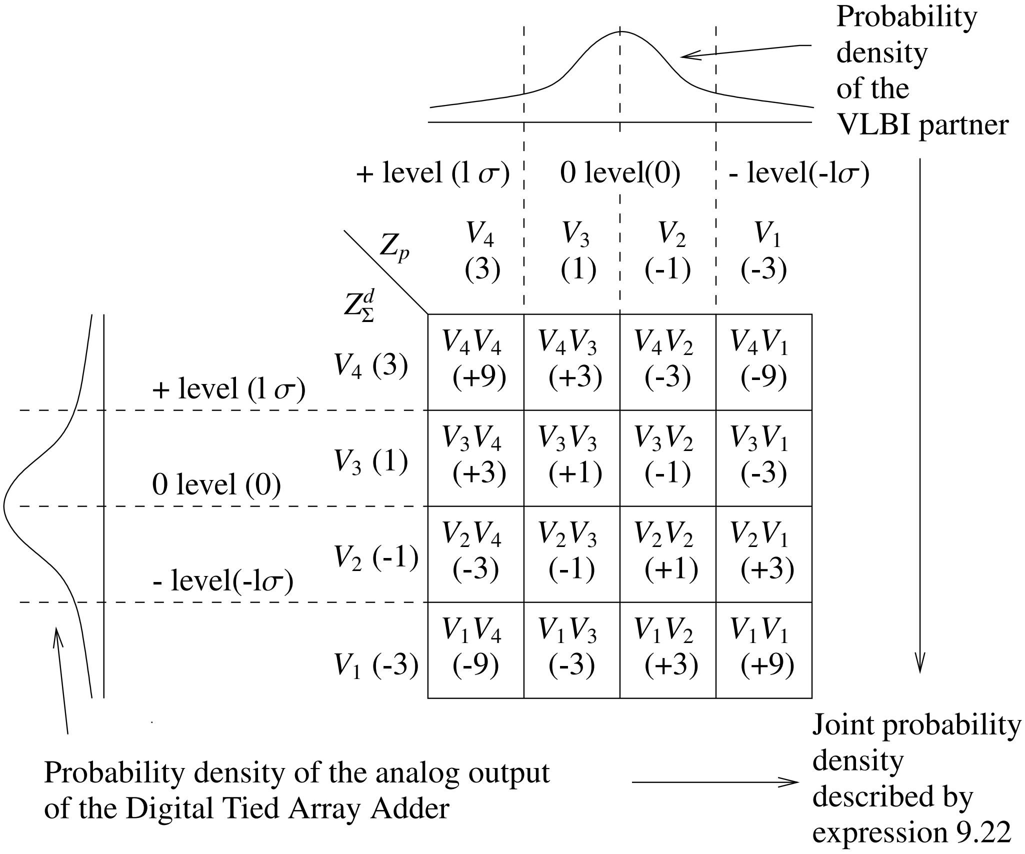
The superscript 'd, int' indicates digital summation and the correlation interferometer mode. We assume small correlation coefficients, which means that the correlation between the signals to be correlated, Z_Σ^d and Z_p, is close to zero, thus $\mathrm{E}(Z_\Sigma^d \cdot Z_p)$ is close to zero. We approximate the SNR as follows:

$$SNR^{d,int} \approx \frac{\mathrm{E}(Z_\Sigma^d \cdot Z_p)}{\sqrt{\mathrm{E}((Z_\Sigma^d \cdot Z_p)^2)}} \frac{\sqrt{\Delta\nu\, t\, N}}{K_s} \tag{9.24}$$

To determine the SNR, we have to determine $\mathrm{E}(Z_\Sigma^d \cdot Z_p)$ and $\mathrm{E}((Z_\Sigma^d \cdot Z_p)^2)$. Z_Σ^d and Z_p are Gaussian noise signals which are coarsely quantized. The quantization process is illustrated in figure 9.2. Each signal can have 4 possible output values. The maximum number of combinations of Z_Σ^d and Z_p therefore equals 16. We can construct a matrix, called the product matrix of the correlator, where the row index is determined by the digital value at one input (say the digitized output Z_Σ^d from the digital TAA) and the column index is determined by the digitized signal from the VLBI partner Z_p. An element within the matrix gives $Z_c = Z_\Sigma^d \cdot Z_p$, which equals the product of the two digital values used to select the row and column in which the element resides. The joint probability density of the inputs Z_Σ^d and Z_p is given in expression 9.22. The correlator product matrix, the input signals and their probability densities are schematically illustrated in figure 9.7.

The output of the multiplier is the product of the two inputs. Both inputs have only four possible values V_1, V_2, V_3 or V_4. Values that are often used for 2-bit correlation are -3, -1, 1 and 3 respectively. In this case, there are 6 possible output values ranging between -9 and 9.

To calculate the average value of the correlator output $\mathrm{E}(Z_c)$, we have to calculate, for every possible output value, the probability that this output will occur


Probability density of the VLBI partner
+ level (1 σ)
0 level(0)
- level(-lσ)
+ level (1 σ)
0 level (0)
- level(-lσ)
Probability density of the analog output of the Digital Tied Array Adder
Joint probability density described by expression 9.22

Z^d_Σ \ Z_p	V_4 (3)	V_3 (1)	V_2 (-1)	V_1 (-3)
V_4 (3)	V_4V_4 (+9)	V_4V_3 (+3)	V_4V_2 (-3)	V_4V_1 (-9)
V_3 (1)	V_3V_4 (+3)	V_3V_3 (+1)	V_3V_2 (-1)	V_3V_1 (-3)
V_2 (-1)	V_2V_4 (-3)	V_2V_3 (-1)	V_2V_2 (+1)	V_2V_1 (+3)
V_1 (-3)	V_1V_4 (-9)	V_1V_3 (-3)	V_1V_2 (+3)	V_1V_1 (+9)

Figure 9.7: Product matrix of the correlator

and use these values in the following way:

$$\mathrm{E}(Z_\Sigma^d \cdot Z_p) = \sum_{i=1}^{4} \sum_{j=1}^{4} P_{V_i V_j}^d \cdot V_i V_j \tag{9.25}$$

where $P_{V_i V_j}^d$ indicates the probability that the quantized input from the digital Tied Array equals V_i and the quantized input from the VLBI partner equals V_j. We will give expressions for $P_{V_i V_j}^d$ later on. First, we will further elaborate on how these values are used to calculate $SNR^{d,int}$.

If we assume that both the density of the output of the TAA and the density of the signal from the VLBI partner are symmetrical around the 0-level, we only have to evaluate either the upper-, lower-, left- or right half of the product matrix. We chose to evaluate the upper half:

$$\mathrm{E}(Z_\Sigma^d \cdot Z_p) = 2 \sum_{i=3}^{4} \sum_{j=1}^{4} P_{V_i V_j}^d \cdot V_i V_j \tag{9.26}$$

In a similar way, $\mathrm{E}((Z_\Sigma^d \cdot Z_p)^2)$ (=$\mathrm{E}(Z_c^2)$) can be calculated:

$$\mathrm{E}((Z_\Sigma^d \cdot Z_p)^2) = 2 \sum_{i=3}^{4} \sum_{j=1}^{4} P_{V_i V_j}^d \cdot (V_i V_j)^2 \tag{9.27}$$

The SNR, in case of a digital TAA used in the correlation interferometer mode, then equals:

$$SNR^{d,int} = \frac{\sum_{i=3}^{4} \sum_{j=1}^{4} P_{V_i V_j}^d \cdot V_i V_j}{\sqrt{\sum_{i=3}^{4} \sum_{j=1}^{4} P_{V_i V_j}^d \cdot (V_i V_j)^2}} \cdot \frac{\sqrt{\Delta\nu\, t\, N}}{K_s} \tag{9.28}$$

To evaluate $P_{V_i V_j}^d$, we have to use the joint probability density of Y_Σ^d and Y_p, given in expression 9.22. The following expressions describe the probabilities $P_{V_i V_j}^d$ for $i = 3, 4$ and $j = 1, 2, 3$ and 4:

$$
\begin{aligned}
P^d_{V_4V_4} &= \sum_{i=\text{ceil(‘+‘ level)}}^{\text{upper limit}} \int_l^{\infty} P_{Y^d_\Sigma Y_p}(i, y_p)\, dy_p && (9.29)\\
P^d_{V_4V_1} &= \sum_{i=\text{ceil(‘+‘ level)}}^{\text{upper limit}} \int_{-\infty}^{-l} P_{Y^d_\Sigma Y_p}(i, y_p)\, dy_p && (9.30)\\
P^d_{V_3V_4} &= \sum_{i=1}^{\text{floor(‘+‘ level)}} \int_l^{\infty} P_{Y^d_\Sigma Y_p}(i, y_p)\, dy_p && (9.31)\\
P^d_{V_4V_3} &= \sum_{i=\text{ceil(‘+‘ level)}}^{\text{upper limit}} \int_0^{l} P_{Y^d_\Sigma Y_p}(i, y_p)\, dy_p && (9.32)\\
P^d_{V_3V_1} &= \sum_{i=1}^{\text{floor(‘+‘ level)}} \int_{-\infty}^{-l} P_{Y^d_\Sigma Y_p}(i, y_p)\, dy_p && (9.33)\\
P^d_{V_4V_2} &= \sum_{i=\text{ceil(‘+‘ level)}}^{\text{upper limit}} \int_{-l}^{0} P_{Y^d_\Sigma Y_p}(i, y_p)\, dy_p && (9.34)\\
P^d_{V_3V_3} &= \sum_{i=1}^{\text{floor(‘+‘ level)}} \int_0^{l} P_{Y^d_\Sigma Y_p}(i, y_p)\, dy_p && (9.35)\\
P^d_{V_3V_2} &= \sum_{i=1}^{\text{floor(‘+‘ level)}} \int_{-l}^{0} P_{Y^d_\Sigma Y_p}(i, y_p)\, dy_p && (9.36)
\end{aligned}
$$

To understand the boundaries of the summations, we have to realize that the digital signal after summation is converted to an analog signal. We assume an ideal DAC which converts the decimal value linearly into a current or voltage, proportional to the decimal input value. This analog signal is quantized with a quantizer with three reference levels, the ’+’ level, the ’0’ level and the ’-’ level (see figure 9.2). Because we use the upper half of the product matrix of the correlator, we concentrate on the ’+’ and ’0’ levels and the maximum value of the sumsignal. The ’upper limit’ corresponds to the maximum decimal value at the output of the summation. For example, for 14 telescopes, the maximum decimal value equals $14 \times 3 = 42$. The ’+’ level corresponds to l times the standard deviation of the signal at the output of the summation. In general, l times the standard deviation will not equal a decimal value and we therefore use the ’ceil’ and ’floor’ operations to determine the summation ranges. The ceil operation selects the nearest decimal value which is larger than the given value. The floor operation selects the nearest decimal value, which is smaller than the given value. The ’0’ level corresponds to the average value of this signal. Because the outputs of the quantizers are chosen symmetrically around zero, the average value of the signal after summation will be zero as well.

Because of this, there is a finite probability that a decimal output of the summation exactly equals the ’0’ level. Note that this is only possible if the signals from an even number of telescopes are summed. If for example the signals from two telescopes are summed, the possible values after summation are: -6, -4, -2, 0, 2, 4 and 6. In that case, the values that equal 0 are not used because the summation starts at 1 (see expressions 9.31, 9.33, 9.35 and 9.36).

9.4.1.2 Signal to noise ratio in case of an analog TAA

For the derivation of the SNR in case of an analog TAA, we use the same approach as for the derivation in case of a digital TAA. We first derive the joint probability density of the output of the adder and the signal from the VLBI partner. This density is used, together with the product matrix of the correlator, to determine the SNR in case of an analog TAA, $SNR^{a,int}$. The superscript ’a, int’ indicates analog summation and the correlation interferometer mode.

The joint probability density function of the analog signal of the VLBI partner (Y_p) and the analog signal of one individual element of the array (Y_i) is known and the average of the product of these 2 signals equals: $\mathrm{E}(Y_p \cdot Y_i) = r \cdot \sigma_{Y_p} \cdot \sigma_{Y_i}$ assuming the delays will be corrected ideally. Of interest is the joint probability density function of Y_p and the analog signal Y^a_Σ at the output of the adder. We define r^a as the correlation between the signal Y_p from the VLBI partner and the output of the analog adder Y^a_Σ:

$$r^a = \frac{\mathrm{E}(Y_p \cdot Y^a_\Sigma)}{\sigma_{Y_p} \cdot \sigma_{Y^a_\Sigma}} \tag{9.37}$$

For the power at the output of the array ($\sigma^2_{Y^a_\Sigma}$) we find according to [14], [88] and [89]:

$$\sigma^2_{Y^a_\Sigma} = M\sigma^2_{Y_i}(1 + (M-1)r) \tag{9.38}$$

where M equals the number of telescopes within a subarray. $\mathrm{E}(Y_p \cdot Y^a_\Sigma)$ equals:

$$\mathrm{E}(Y_p \cdot Y^a_\Sigma) = M \cdot \mathrm{E}(Y_p \cdot Y_i) = M \cdot r \cdot \sigma_{Y_p} \cdot \sigma_{Y_i} \tag{9.39}$$

By using expressions 9.38 and 9.39 in expression 9.37 and assuming $\sigma_{Y_i} = \sigma_{Y_p}$

we obtain:

$$r^a = r\sqrt{\frac{M}{(1+(M-1)r)}} \tag{9.40}$$

The joint probability density function of Y_p and Y^a_Σ is just a bivariate normal distribution with correlation coefficient r^a:

$$P_{Y^a_\Sigma Y_p}(y^a_\Sigma, y_p) = \frac{1}{2\pi\sigma_{Y^a_\Sigma}\sigma_{Y_p}\sqrt{1-(r^a)^2}} e^{-\frac{1}{2(1-(r^a)^2)}\left(\frac{(y^a_\Sigma)^2}{\sigma^2_{Y^a_\Sigma}} - \frac{2r^a y^a_\Sigma y_p}{\sigma_{Y_p}\sigma_{Y^a_\Sigma}} + \frac{y_p^2}{\sigma^2_{Y_p}}\right)} \tag{9.41}$$

For small correlation coefficients, the analog SNR ($SNR^{a,int}$) can be defined as:

$$SNR^{a,int} \approx \frac{\mathrm{E}(Z^a_\Sigma \cdot Z_p)}{\sqrt{\mathrm{E}\Big((Z^a_\Sigma \cdot Z_p)^2\Big)}} \frac{\sqrt{\Delta\nu\, t\, N}}{K_s} \tag{9.42}$$

The expected values are determined in a way, similar to the digital case:

$$SNR^{a,int} = \frac{\sum_{i=3}^{4}\sum_{j=1}^{4} P^a_{V_iV_j} \cdot V_iV_j}{\sqrt{\sum_{i=1}^{4}\sum_{j=1}^{4} P^a_{V_iV_j} \cdot (V_iV_j)^2}} \cdot \frac{\sqrt{\Delta\nu\, t\, N}}{K_s} \tag{9.43}$$

The different probabilities are calculated as indicated below:

$$P^a_{V_4V_4} = \int_l^\infty \int_l^\infty P_{Y^a_\Sigma Y_p}(y^a_\Sigma, y_p)\, dy_p \cdot dy^a_\Sigma \tag{9.44}$$

$$P^a_{V_4V_1} = \int_l^\infty \int_{-\infty}^{-l} P_{Y^a_\Sigma Y_p}(y^a_\Sigma, y_p)\, dy_p \cdot dy^a_\Sigma \tag{9.45}$$

$$P^a_{V_3V_4} = \int_0^l \int_l^\infty P_{Y^a_\Sigma Y_p}(y^a_\Sigma, y_p)\, dy_p \cdot dy^a_\Sigma \tag{9.46}$$

$$P^a_{V_4V_3} = \int_l^\infty \int_0^l P_{Y^a_\Sigma Y_p}(y^a_\Sigma, y_p)\, dy_p \cdot dy^a_\Sigma \tag{9.47}$$

$$P^a_{V_3V_1} = \int_0^l \int_{-\infty}^{-l} P_{Y^a_\Sigma Y_p}(y^a_\Sigma, y_p)\, dy_p \cdot dy^a_\Sigma \tag{9.48}$$

$$P^a_{V_4V_2} = \int_l^\infty \int_{-l}^0 P_{Y^a_\Sigma Y_p}(y^a_\Sigma, y_p)\, dy_p \cdot dy^a_\Sigma \tag{9.49}$$

$$P^a_{V_3V_3} = \int_0^l \int_0^l P_{Y^a_\Sigma Y_p}(y^a_\Sigma, y_p)\, dy_p \cdot dy^a_\Sigma \tag{9.50}$$

$$P^a_{V_3V_2} = \int_0^l \int_{-l}^0 P_{Y^a_\Sigma Y_p}(y^a_\Sigma, y_p)\, dy_p \cdot dy^a_\Sigma \tag{9.51}$$

These expressions resemble the expressions for $P^d_{V_iV_j}$ in case of a digital TAA. The difference is that, in case of a digital TAA, the probability density of the output signal indicates a non-zero probability only for a limited number of discrete values. In case of an analog TAA, the probability density of the output signal is a Gaussian distribution, having a non-zero probability density for all values. For that reason, the summations in the expressions for $P^d_{V_iV_j}$, in case of a digital TAA, are replaced by corresponding integrals, in case of an analog TAA. The probabilities $P^a_{V_iV_j}$ are used in expression 9.43 to calculate $SNR^{a,int}$.

9.4.1.3 The Degradation Factor

We now have an expression for the SNR in case of a digital TAA (expression 9.28) and an expression for the SNR in case of an analog TAA (expression 9.43). We define the Degradation Factor η^{int} as follows:

$$\eta^{int} = \frac{SNR^{d,int}}{SNR^{a,int}} \tag{9.52}$$

Expression 9.52 is used to calculate Degradation Factors for different situations. For both one- and two-bit correlation interferometers, the Degradation Factors can be determined for 2-, 3- and 4-level quantization before summation. For one-bit correlation, the Degradation Factors are given in Table 9.2 and for two-bit

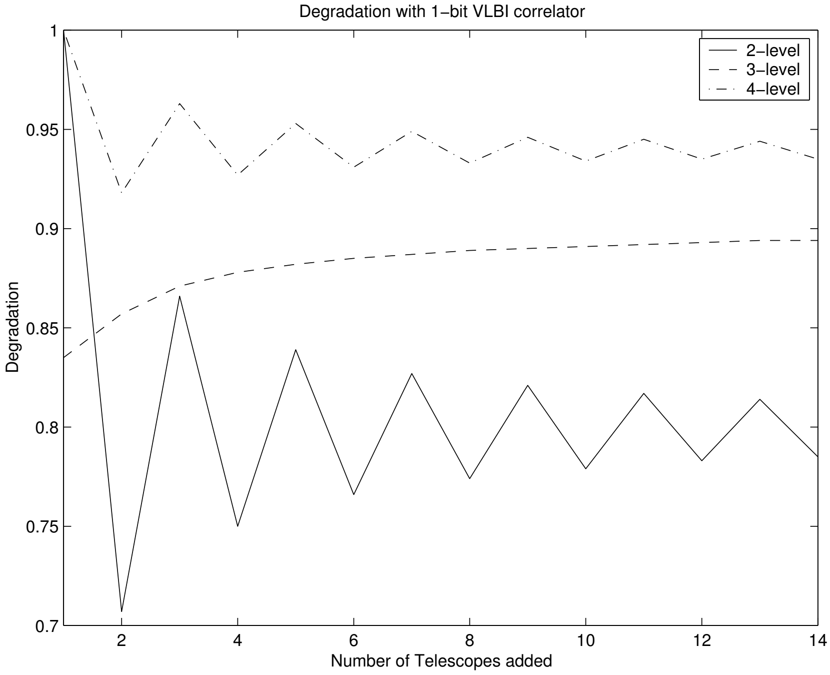


Figure 9.8: One-bit correlation Degradation Factors in case of 2-, 3- and 4-level addition

correlation, the factors are given in Table 9.3. A graphical representation of the same factors is given in figures 9.8 and 9.9.

For 2- and 4-level summation, an alternatively growing and falling degradation appears. This can be explained as the zero-redundancy effect due to the adding of an even number of digitized signals. In this case, there is a non-zero probability that the output of the digital TAA exactly equals the 0-level of the quantizer of the correlator. In our calculations, we ignore these samples (see expressions 9.31, 9.33, 9.35 and 9.36). In practice, the ADC of the correlator will randomly assign the value V_2 or V_3 (see figure 9.2) to these samples, giving the same effect: additional degradation. For 3-level summation, this zero-redundancy effect is present independent of the number of telescopes. Together with the receiver temperatures of the individual antennas, the Degradation Factor can be used to calculate the effective signal temperature and antenna temperature for the Tied Array used as a single dish in the correlation interferometer mode. This is elaborated in appendix H.

If we look at the definitions of $SNR^{d,int}$ and $SNR^{a,int}$ in expressions 9.24 and 9.42 respectively, and realize that the correlator variance using analog addition is nearly equal to the correlator variance using digital addition, $(\mathrm{E}\big((Z^a_\Sigma \cdot Z_p)^2\big) \approx$

Table 9.2: Degradation Factors η^{int} for one-bit correlation

Number of antennas	2-level	3-level	4-level
1	1	0.835	1
2	0.707	0.857	0.918
3	0.866	0.871	0.963
4	0.75	0.878	0.927
5	0.839	0.882	0.953
6	0.766	0.885	0.931
7	0.827	0.887	0.949
8	0.774	0.889	0.933
9	0.821	0.89	0.946
10	0.779	0.891	0.934
11	0.817	0.892	0.945
12	0.783	0.893	0.935
13	0.814	0.894	0.944
14	0.785	0.894	0.935

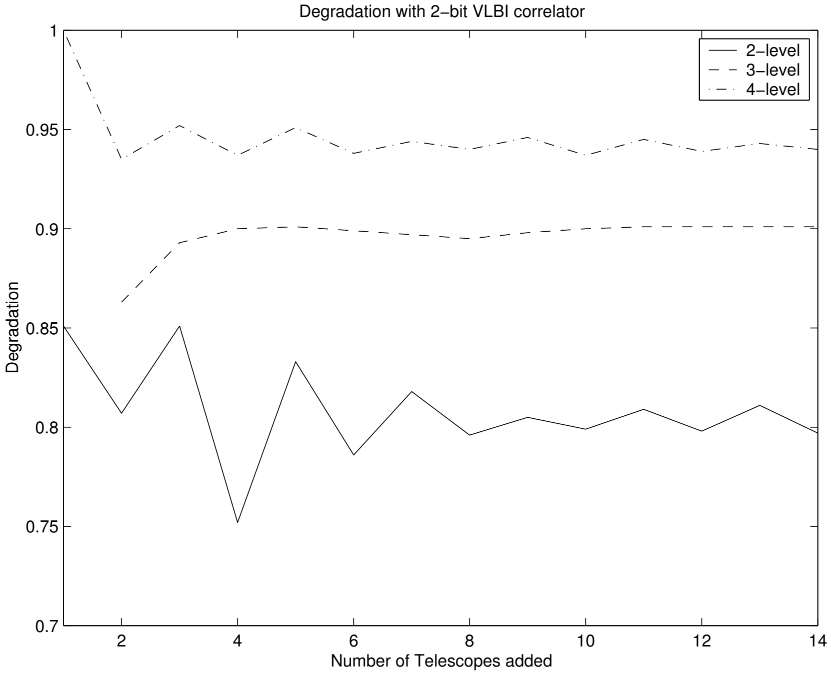


Figure 9.9: Two-bit correlation Degradation Factors in case of 2-, 3- and 4-level addition

Table 9.3: Degradation Factors η^{int} for two-bit correlation

Number of antennas	2-level	3-level	4-level
1	0.851		1
2	0.807	0.863	0.935
3	0.851	0.893	0.952
4	0.752	0.9	0.937
5	0.833	0.901	0.951
6	0.786	0.899	0.938
7	0.818	0.897	0.944
8	0.796	0.895	0.94
9	0.805	0.898	0.946
10	0.799	0.9	0.937
11	0.809	0.901	0.945
12	0.798	0.901	0.939
13	0.811	0.901	0.943
14	0.797	0.901	0.94

$\mathrm{E}\big((Z_\Sigma^d \cdot Z_p)^2\big)$), the following is valid:

$$\eta^{int} \approx \frac{\mathrm{E}(Z_\Sigma^d \cdot Z_p)}{\mathrm{E}(Z_\Sigma^a \cdot Z_p)} = \frac{\mathrm{E}(Z_\Sigma^d \cdot Z_p)/E_{\rho=1}}{\mathrm{E}(Z_\Sigma^a \cdot Z_p)/E_{\rho=1}} = \frac{\rho_d}{\rho_a} \tag{9.53}$$

where $E_{\rho=1}$ equals the correlator output if fully correlated signals are present at the inputs of the ADCs, ρ_d is the measured correlation coefficient using a digital correlator and ρ_a is the measured correlation coefficient using an analog correlator. Thus, calculated values for η^{int} also represent the quotient of ρ_d and ρ_a for small correlation coefficients.

9.5 Summary of the results

From the analysis, we see that the efficiency reduces in case the output of a digital TAA is connected to a total power receiver and that there is a degradation in performance if the output of the digital TAA is used in a correlation interferometer. Efficiency and degradation are related to the case where an analog TAA is used. Both the efficiency and degradation depend on the number of telescopes within a subarray. Generally, the efficiency in case of a total power receiver increases for an increasing number of telescopes. It appears that for the sum of an infinite number of telescopes, the efficiency approximates the well-known factors $\frac{2}{\pi}$, 0.81 and 0.88 for two-, three- and four-level quantization respectively. Furthermore, in case of a correlation interferometer using a digital TAA, the analysis shows that for two- and four-level summation of a small number of antennas, an alternatively growing and falling degradation appears. This can be explained as the zero-redundancy effect due to the addition of an even number of digitized signals in these cases. For three-level summation, this zero-redundancy effect is present independent of the number of telescopes.

9.6 Conclusion

The overall evaluation of the Analog-Digital Codesign of a TAA is given in table 9.4.

	analog Tied Array Adder	digital Tied Array Adder
Pathlength Compensation	Switching between waveguides	Done by ADC subsystem
Isolation	-30 to -40 dB	< -70 dB
Efficiency	1	> $2/\pi$ and < 0.88
Degradation	1	> 0.75 and < 1

Table 9.4: Comparison between the analog Tied Array Adder and the digital Tied Array Adder

Pathlength compensation will be very difficult to realize in an analog TAA. Switching between waveguides of different length is not an attractive option for baselines in the order of kilometers. A digital TAA for the WSRT can use the pathlength compensation system of the DZB backend. This pathlength compensation system combines two techniques to realize a delay. The sampling clock can be shifted and a digital memory is used to delay the signal in multiples of cycles of the sampling clock. Currently, this is an efficient way to compensate for pathlength differences.

When using an analog TAA, satisfying the WSRT requirements concerning isolation will be very difficult if not impossible. The requirement is that the signal power, at the output of the TAA, from a telescope that is not part of a subarray is 70 dB below the power of this signal at the input of the TAA. The suppression is called isolation. If analog power combiners are used, an isolation of -30 to -40 dB is feasible. This is not sufficient. In case of a digital TAA, the isolation is very large if the general design rules for digital design are followed.

Because of the coarse quantization, used in the ADC subsystem, quantization errors are introduced. Because of these quantization errors or quantization noise, the SNRs of the signals are reduced. If a digital TAA output is connected to a total power receiver, the efficiency of the overall system is reduced because of the coarse quantization. This reduction is related to the case where an analog TAA output is connected to a total power receiver. The efficiency is reduced to a level between 64 % and 88 %.

If the output of a digital TAA is connected to a correlation interferometer, the performance of the overall system degrades compared to the case where an analog TAA is used. This degradation factor ranges between 0.75 and 1.

Both the efficiency and the degradation factor have values closer to 1 if quantizers with a resolution larger that 2 bits are used. These situations can be analyzed using the approach presented above, starting with the derivation of the probability density of the sumsignal at the output of the TAA. However, since our aim was to use a part of the already existing ADC subsystem, we did not analyze the effects of ADCs with higher resolution.

Concluding, from an implementation point of view the digital TAA is preferred. Specifications are easier to satisfy and the costs are lower since a large part of the TAA (the pathlength difference compensation) is already present in another part of the system. From an observational point of view, the analog TAA is preferable because the digital TAA has a lower efficiency (total power receiver) and leads to degradation (correlation interferometer).

For the implementation of a TAA for the WSRT, it was decided to implement the digital TAA. The implementation aspects were given a higher priority than the lower efficiency and the degradation.

9.7 Reflection on the design process

In chapter 8, we have described the design of a Tied Array Adder using the Y-chart approach. We have gone through two iterations of the Y-chart approach. In the first iteration, an analog TAA was briefly designed. In the second iteration, a digital TAA using coarse quantization was described. In this chapter, we finalized the performance analysis of the second iteration. In this section, we will reflect on both design iterations.

9.7.1 The first iteration: the analog TAA

In the first iteration, we have gone through all stages of the Y-chart approach. These stages are: ’Architecture Specification’, ’Functional Specification’, ’Mapping’ and ’Performance Analysis’ (see figure 1.11). The most important aspects within the different stages are described below.

- Architecture Specification. The architecture consists of an analog processing platform.
- Functional Specification. We have basically specified the functionality of the TAA and the requirements concerning pathlength compensation and isolation.
- Mapping. Because we have not defined the structure of the analog processing platform, we have stated in the mapping stage that the pathlength compensation and the summation will be realized on the analog processing platform.
- Performance Analysis. Both the pathlength compensation and the summation of signals from different telescopes will be difficult to realize economically, using the analog processing platform. For that reason we have decided to make use of the feedback path from ’Performance Numbers’ to ’Architecture Specification’. The goal of the second iteration of the Y-chart approach is to modify the architecture to make realization of a TAA feasible.

9.7.2 The second iteration: the digital TAA

In the second iteration, we have gone through the ’Architecture Specification’, the ’Mapping’ and ’Performance Analysis’ stages of the Y-chart approach. The ’Application’ has not changed. The most important aspects of the different stages are described below.

- Architecture Specification. The basic idea is to realize the TAA in the digital domain. Because the signals needed within the TAA are already digitized in the ADC Sybsystem of the DZB, the ADCs within this subsystem are used to convert the signals from the analog into the digital domain. Compensation of pathlength differences and summation of signals from different telescopes is realized in the digital domain. The sumsignals are converted into the analog domain by DACs.
- Mapping. Besides the ADCs, pathlength compensation is also realized in the ADC subsystem. The pathlength compensation within the TAA is mapped onto this part of the ADC subsystem. The summation is mapped onto a digital processing part, dedicated to the TAA.
- Performance Analysis. The requirements concerning pathlength compensation and isolation can be satisfied by the digital TAA. However, the ADC

subsystem uses coarse quantization. The performance analysis is therefore focussed on the analysis of the effects of coarse quantization. The analog TAA acts as a reference and coarse quantization leads to loss of efficiency and degradation.

The result of the two iterations is that an analog TAA is not feasible and that the TAA will be implemented on a digital processing platform. When using parts of the ADC subsystem, the effects of coarse quantization lead to an update of the functional specifications because of the efficiency loss and degradation.

Chapter 10

Conclusion and future work

Many electronic systems can be characterized as mixed-signal systems; they consist of an analog part and a digital part. The design of a mixed-signal system is very often split into the design of the analog part and the design of the digital part in an early stage of the project. We have called this approach the sequential design methodology using the Specify-Explore-Refine-Verify paradigm. The interfaces between the analog- and digital part are defined and frozen in a very early stage and the analog- and digital design teams work independently and in parallel. Both design teams solve their own problems and if the interfaces are defined correctly, this leads to a workable design in a controlled way. This way of organizing the design of a mixed-signal system might lead to an optimal solution from an engineering-management point of view but it very often leads to suboptimal solutions from a realization- or implementation point of view.

An alternative Analog-Digital Codesign methodology is the iterative methodology using the Y-chart approach. Using this methodology, the role of the interface between the analog- and digital parts of the system differs from role of the interface in the sequential design approach. In the sequential design approach, the interface is more or less a foothold on which the digital- and analog designs are based. In Analog-Digital Codesign using the Y-chart approach, the interface is one of the elements that can be used to influence the design. It is an additional degree of freedom.

In mixed-signal systems, the Analog-Digital interface converts digital signals into analog signals and analog signals into digital signals. These conversion processes are basically specified by two parameters which have different interpretations in the two domains. In the digital domain, these parameters are sampling rate and resolution. In the analog domain these parameters are related to the bandwidth of the signals and the Signal-to-Noise Ratio (SNR). In general, the conversion from digital to analog is easier to implement than the conversion from analog to digital. Feasible sampling rates (bandwidth) and resolutions (SNR) are therefore often determined by the analog to digital conversion rather than the digital to analog conversion. The implementation of devices that realize the conversion from analog to digital (ADC) is generally considered as part of an analog design process.

In this thesis, we used the resolution (SNR) of the interface between the analog and digital part of a mixed-signal system as an element to influence the design of that system. By reducing the resolution without increasing the sampling rate, the complexity of the analog part of the system is reduced significantly. We concentrate on the resolution of the ADC in particular. We even reduce the resolution of the ADCs to an extreme extend (less than 5 bits). In this case, the quantization is called coarse quantization.

We have selected two cases to illustrate the effectiveness of coarse quantization using the Y-chart approach. In both cases, we go through the Analog-Digital Codesign loop iteratively where coarse quantization is used, leading to a reallocation of tasks and to alternative digital- and analog designs. The subject of the first case study is digital predistortion of power amplifiers (PAs). A PA is the last active stage in a (telecommunication) transmitter. The ideal PA is a linear device: the amplitude of the output signal is a constant value times the amplitude of the input signal. In practice, a PA exhibits non-linear behavior. The result is that the original signal is more difficult to detect at a receiving device and that signals are transmitted at frequencies where it is prohibited. To counteract these effects, digital predistortion is used. The general idea is to predistort the original signal in such way that the combination of the predistortion and the non-linear behavior of the PA results in linear amplification. Because the non-linear characteristics of a PA change in time, due to temperature changes and aging, the behavior of the PA is monitored continuously. The output of the PA is fed-back to a device which is controlling the predistortion. By comparing the signal at the input of the PA with the signal from the output of the amplifier, the non-linear characteristics of the PA can be extracted. Because the PA is an analog device and the predistortion is executed in the digital domain, the feedback path contains an ADC. In this case study we analyzed, during the first iteration of the Y-chart approach, the performance of an existing solution, the so-called Least Squares (LS-)predistorter. The LS predistorter gives significant reduction of the distortion that is introduced. Up to 16 dB of suppression of unwanted signals (Adjacent Channel Interference or ACI levels) can be obtained. The resolution of the ADC in the feedback path can be reduced to approximately 4 to 5 bits without significantly reducing the performance. For less bits, the performance degrades. A way to boost performance is to inject a dithering signal just before the ADC. The basic principle is that the quantization error and the original signal become uncorrelated because of the added dither signal. An important condition is that the dither signal should have sufficient power. In our simulations we used uniformly distributed noise with a range corresponding to one quantization step (1 LSB). However, the predistortion control that uses the data from the input and output of the PA, needs more data to estimate the non-linear behavior of the PA. In case of 5 bits resolution in the feedback path, 8k samples are needed to obtain a suppression of ACI levels of approximately 10 dB. If 2-bit quantization is used, 32k samples are needed to obtain a suppression of 6 dB. One characteristic of the LS predistorter is that the complexity of the digital processing that controls the predistortion, scales linearly with the number of samples used to estimate the non-linearity of the PA. This extends the time between the moment that the data is available and the time that the estimates of the non-linearity of the PA are available, significantly.

In the second iteration of the Y-chart approach of a digital predistorter for PAs, we tried to find an algorithm which first of all has a lower complexity than the LS predistorter. Second, the complexity of the algorithm should be independent of the number of samples needed to update the predistorter. This reduces the time between the availability of the data and the moment on which the predistorter can be updated if the same processing resources are used as for the LS predistorter. If the same timing as for the LS predistorter is allowed, less processing resources are sufficient. The resulting predistorter is called the Crosscorrelation predistorter. The general idea is to preprocess the signals from the input and output of the PA with very low complexity, resulting in a small condensed data set that contains all the relevant information. The Least Squares algorithm, as used in the LS predistorter, can then deal with this condensed data set. The preprocessing is realized by crosscorrelating the signals at the in- and output of the PA with a single-bit reference signal. The complexity of this operation is very low because no full precision multiplications are involved. The number of outputs of the crosscorrelation (the number of points of the crosscorrelation function) is fixed and, in general, much lower than the number of samples on which each point of the crosscorrelation function is based. In our simulations we used crosscorrelation functions consisting of 64 points. The Fourier Transforms of the crosscorrelation functions are calculated resulting in spectra and the Least Squares algorithm uses these spectra to extract the non-linear characteristics of the PA. By introducing the crosscorrelation, the number of samples used by the Least Squares algorithm has been reduced in typical cases from 8k points to 64 points.

By means of simulation, we have shown that the performance of the Crosscorrelation predistorter is better than, or equal to the performance of the LS predistorter under the condition that the resolution of the ADC in the feedback path is sufficient. However, the Crosscorrelation predistorter is more vulnerable to quantization effects. In general, if less than 10 bits are used in the feedback path, the performance of the Crosscorrelation predistorter is less than the performance of the LS predistorter using the same amount of data. The performance of the Crosscorrelation predistorter can be increased by adding a dither signal, similar to the dither signal for the LS predistorter, and by increasing the number of samples used to calculate the crosscorrelation function. This increase of data does not affect the number of full precision multiplications that have to be executed. So, in general, the complexity is not increased, only the update rate of the predistorter will be affected.

The effects of coarse quantization and the addition of a dither signal have been analyzed quantitatively. Because of quantization, quantization noise is added to the original signal. In case of coarse quantization, the quantization noise is correlated to the original signal and both the LS- and Crosscorrelation predistorter introduce systematic errors in the estimate of the non-linear behavior of the PA. By adding a dither signal, the correlation between the quantization noise and the input signal is reduced, improving the performance of both types of predistorters.

In the second case study presented in this thesis, the Analog-Digital Codesign of a part of a Radio Telescope is described. A Radio Telescope is an instrument used for astronomical observations at Radio Frequencies (30 MHz to 300 GHz). Two different types of telescopes are used: single-dish and multiple-dish. In this case study, we concentrated on the design of a Tied Array Adder (TAA), a part of the Westerbork Synthesis Radio Telescope (WSRT) which is a multiple-dish telescope. Within a TAA, the signals from different dishes are first aligned in time (pathlength compensation) and then summed. This enables the multiple-dish telescope to be operated as a single dish. The Analog-Digital Codesign of a TAA for the WSRT basically consists of two iterations of the Y-chart approach. In the first iteration a completely analog realization is analyzed, in the second iteration a completely digital realization is analyzed.

The starting point of the first iteration is the functional specification. The most important specifications concern the pathlength compensation and the isolation. Isolation is defined as the ratio of the power of a signal at the output and the power of a signal at the input of a TAA in case the signal should not participate in the summation. In the ideal case, this should be less than -70 dB. In the performance analysis of the analog design, it becomes clear that the implementation of pathlength compensation in the analog domain is cumbersome and that the realization of sufficient isolation is impossible without changing the functional specifications. Since it is not desirable to change these parts of the functional specification, it was chosen to develop an alternative architecture.

In the second iteration of the Analog-Digital Codesign of the TAA using the Y-chart approach, the pathlength compensation and summation are mapped onto the digital part of the system. This eases the realization of the pathlength compensation considerably. Pathlength compensation had already been realized in another part of the WSRT, the DZB backend. In this backend, telescope signals are coarsely quantized (2 bits) before pathlength compensation. Pathlength compensation is realized by means of digital memories and a mechanism to change the phase of the sampling clock. Because of the use of digital signals, the digital summation gives more than sufficient isolation. However, coarse quantization introduced by the DZB backend, adds quantization noise to the telescope signals. In the performance analysis of the second iteration, the effects of coarse quantization in the TAA application are analyzed. The effects differ for different types of observations. We considered two types. In the first type of observation, a total power receiver is used to measure the power at the output of the TAA. In the second type of observation, the output of the TAA is connected to a correlation interferometer where the TAA output is crosscorrelated with another signal. This can be a signal originating from a single dish telescope or from another TAA. For both cases, the SNR in case a digital TAA is used (second iteration of the Analog-Digital Codesign of a TAA), is compared with the SNR in case an analog TAA is used (first iteration). In both the total power receiver and the correlation interferometer, the digital TAA results in a lower SNR. In case of a total power receiver, the lower SNR attributes to a factor called 'efficiency'. The efficiency depends on the number of telescopes of which the signals are summed and ranges between 64 % for one-bit quantization before summation and 88 % for two-bit quantization before summation. In case of a correlation interferometer using a TAA, the ratio between the SNR in case of a digital TAA and an analog TAA is called 'degradation'. The degradation is also dependent on the number of telescopes that participate in the summation and ranges between 75 % (one-bit summation) and 95 % (two-bit summation).

From both iterations, it is clear that the functional specification is affected. The analog design results in a mechanism for pathlength compensation which is hardly feasible and the specification concerning isolation can not be satisfied. However, it does not lead to efficiency-loss or degradation. The digital design results in an elegant and economically interesting design for pathlength compensation while satisfying the isolation requirement. However, efficiency-loss and degradation have to be accepted.

Both case studies, digital predistortion of PAs and a TAA for a Radio Telescope, show the advantages of Analog-Digital Codesign via the Y-chart approach using coarse quantization. First, the Y-chart approach facilitates and encourages the search for alternative, more optimal solutions. The explicit performance analysis leads to a critical reflection of the analog- and digital design via explicit feedback paths to functional specification, mapping and architecture specification. Second, the Analog-Digital Codesign principle using the Y-chart approach, leads to a more structured design flow where decisions at different levels are taken explicitly. This design flow can also serve as a vehicle for the documentation process. By assigning documents to the different iterations and phases of the Y-chart approach, a well-organized documentation structure is introduced.

In this thesis, coarse quantization is used as an instrument to influence the design of mixed-signal systems. Reducing the resolution of primarily the ADC is an effective way to reduce the complexity of the analog part of the system. The analysis of the effects of coarse quantization is, however, awkward. For that reason, the use of coarse quantization is very often not investigated and therefore its potential advantages remain undiscovered. In Radio Astronomy, however, there is a tradition concerning coarse quantization. In Radio Telescopes, the signals that are received are Gaussian signals and the effects of coarse quantization of Gaussian signals have been investigated thoroughly. In modern telecommunication standards, the signals from multiple users are multiplexed onto one single communication channel. According to the Central Limit Theorem, the signals in the transmission- and receive paths of the telecommunication equipment therefore can approximately be regarded as Gaussian signals as well. For that reason, the analysis of coarse quantization, developed within the Radio Astronomy application, is useful for telecommunication applications as well. It is important to note that coarse quantization (without oversampling) is not useful if the instantaneous values of signals are of interest, e.g. if the received signal is used to reconstruct a speech signal sent by a telecommunication transmitter. It is only useful if certain characteristics of signals have to be measured. If these signal characteristics are known, instrumental effects e.g. PA non-linearity, can be deduced.

10.1 Main contributions of this thesis

The main contributions of this thesis are:

- The use of the Y-chart approach in two case studies where coarse quantization is used as an instrument to influence the design.

- The development of the Crosscorrelation predistorter using a single-bit representation of the predistorted signal.

- Extension of the existing algorithm to invert a polynomial relation so it can be used in situations where a PA (model) introduces severe memory effects and distortion.

- The introduction of a single general framework to analyze digital predistorters. The single framework is used to describe four different types of predistorters: Indirect Learning - normal polynomials, Indirect Learning - orthogonal polynomials, Direct Learning - normal polynomials and Direct Learning - orthogonal polynomials.

- A simulator to simulate the four different schemes for the two predistorters: the LS- and the Crosscorrelation predistorter. This simulator was used to generate the simulation results.

- A new way to analyze the performance of digital predistorters. The traditional way to analyze the performance is to generate a single signal and use this signal repetitively to determine the predistorter settings. After a few, often not specified number of iterations, the predistorter settings are fixed and the spectrum at the output of the PA is determined. This spectrum is used to calculate an Adjacent Channel Interference (ACI-) level. This level is specific for the specific signal and only for a single power level at the input of the PA. This does not give a realistic view of the performance of a digital predistorter once operational. In our simulator we continuously generate new signals which are fed to the predistorter. We also continuously monitor the output of the PA and judge the performance by two parameters. We look at the average ACI-levels over all the iterations and we look at the compression, introduced by the combination of the predistorter and the PA. We repeat this for different power levels at the input of the predistorter. We also analyzed the effects of (pre)distortion at several power levels at the input instead of a single power level.

- The analysis of the effects of coarse quantization for both the LS- and Crosscorrelation predistorter and the presentation of simulation results when using coarse quantization.

- A general analysis of correlation of quantised noise and periodic signals.

- Presentation of the effects of coarse quantization in a TAA in a structured way. An analysis of the effects was available in [89]. This analysis was structured and extended (see publication [8]) and placed in the context of Analog-Digital Codesign using the Y-chart approach.

10.2 Future work

The research, described in this thesis, can be continued in various ways. First, the implementation aspects of both the LS- and Crosscorrelation predistorters are important. Until now, the analysis of the implementation aspects has been limited to indicating the order of magnitude of the number of full precision multiplications and very briefly elaborating the algorithms for one specific processor. To obtain a realistic view of the feasibility of the predistorters, implementation aspects for different platforms have to be analyzed. These platforms can range from integrated circuits, fine-grain reconfigurable hardware, coarse-grain reconfigurable hardware, Digital Signal Processors (DSPs) to general-purpose microprocessors. One step further is to analyze the implementation on a reconfigurable heterogeneous platform. Second, a testbed is necessary to analyze the effectiveness of predistortion in a real system. This testbed should incorporate the digital processing part, the conversion stages from baseband to radio frequencies (RF) and from RF to baseband, and a PA. Preferably, the testbed should be made modular in such way that different PAs can be tested. Third, it is interesting to investigate the applicability of predistortion to compensate other effects within the analog part of a telecommunication system. Known effects are, for example, I/Q mismatch and LO leakage. The usefulness of predistortion can be investigated in general and the use of the LS- and Crosscorrelation predistorter in particular. Fourth, the concept of digital IF is emerging rapidly. Digital predistortion at IF frequencies can not be analyzed using the framework presented in this thesis. An extension is necessary to fully explore the benefits of digital IF. The fifth subject for future work is to investigate the effects of coarse quantization of digital-IF signals. Sixth, the concept of Analog-Digital Codesign using the Y-chart approach can be used within many other applications. One of the options is to investigate the benefits of this approach for the design of an inkjet for a copier. In future generation copiers, the status of the inkjet will constantly be monitored. When applying the predistortion concept, the status information can be used to control the the ink-flow in such way that the effects of irregularities, introduced by the inkjet, can be reduced. It is interesting to investigate whether the concepts, developed in this thesis, apply to the inkjet application as well.

Appendix A

Elaboration of predistorter options

In this appendix, the options presented in section 4.2.2.4, are worked out resulting in four sets of expressions. These sets are presented in table A.1.

	Normal polynomial	**Orthogonal polynomial**
Direct Learning	$\mathbf{y} = [y(1), ..., y(T)]^T$ $\mathbf{y}_{k\tau} = [\phi_k(x_p(1-\tau)), ..., \phi_k(x_p(T-\tau))]^T$ $\mathbf{\Gamma} = [\mathbf{y}_{10}, ..., \mathbf{y}_{K0}, ..., \mathbf{y}_{1\tau_{max}-1}, ..., \mathbf{y}_{K\tau_{max}-1}]$ $\hat{\mathbf{a}} = (\mathbf{\Gamma}^H\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^H\mathbf{y}$	$\mathbf{y} = [y(1), ..., y(T)]^T$ $\mathbf{y}_{k\tau} = [\psi_k(x_p(1-\tau)), ..., \psi_k(x_p(T-\tau))]^T$ $\mathbf{\Gamma} = [\mathbf{y}_{10}, ..., \mathbf{y}_{K0}, ..., \mathbf{y}_{1\tau_{max}-1}, ..., \mathbf{y}_{K\tau_{max}-1}]$ $\hat{\mathbf{a}} = (\mathbf{\Gamma}^H\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^H\mathbf{y}$
Indirect Learning	$\mathbf{x}_p = [x_p(1), ..., x_p(T)]^T$ $\mathbf{y}_{k\tau} = [\phi_k(y(1-\tau)), ..., \phi_k(y(T-\tau))]^T$ $\mathbf{\Gamma} = [\mathbf{y}_{10}, ..., \mathbf{y}_{K0}, ..., \mathbf{y}_{1\tau_{max}-1}, ..., \mathbf{y}_{K\tau_{max}-1}]$ $\hat{\mathbf{a}} = (\mathbf{\Gamma}^H\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^H\mathbf{x}_p$	$\mathbf{x}_p = [x_p(1), ..., x_p(T)]^T$ $\mathbf{y}_{k\tau} = [\psi_k(y(1-\tau)), ..., \psi_k(y(T-\tau))]^T$ $\mathbf{\Gamma} = [\mathbf{y}_{10}, ..., \mathbf{y}_{K0}, ..., \mathbf{y}_{1\tau_{max}-1}, ..., \mathbf{y}_{K\tau_{max}-1}]$ $\hat{\mathbf{a}} = (\mathbf{\Gamma}^H\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^H\mathbf{x}_p$

Table A.1: Expressions for different predistorter configurations

Appendix B

PA models and simulation results

Within this appendix, different Power Amplifier models (PA models) are introduced and results of simulations are presented. For the different amplifiers, Adjacent Channel Interference (ACI) levels and Compression values are given with and without predistortion for different standard deviations at the input of the predistorter. Two different digital predistortion techniques are used: LS predistortion and Crosscorrelation predistortion. For both techniques, 4 results are given. Each result represents one predistortion scheme. The 4 predistortion schemes are (see also section 4.2.2.4):

1. Indirect Learning, normal polynomials.
2. Indirect Learning, orthogonal polynomials.
3. Direct Learning, normal polynomials.
4. Direct Learning, orthogonal polynomials.

B.1 PA model 1

PA model 1 is a parallel Wiener model (see [64]):

$$y(n) = \sum_{i=1}^{3} F_i(H_i(x(n))) \tag{B.1}$$

$$H_1(z) = 1, \quad H_2(z) = \frac{1 + 0.3z^{-1}}{1 - 0.1z^{-1}}, \quad H_3(z) = \frac{1 - 0.2z^{-1}}{1 - 0.4z^{-1}} \tag{B.2}$$

$$F_i(x) = \sum_{k=1}^{K} a_{ki} x \mid x \mid^{k-1} \tag{B.3}$$

The values for the polynomial distortion coefficients are:

$$\begin{aligned}
a_{11} &= 1.0108 + 0.0858j, & a_{31} &= 0.0879 - 0.1583j,\\
a_{51} &= -1.0992 - 0.8891j, & a_{12} &= 0.1179 + 0.0004j,\\
a_{32} &= -0.1818 + 0.0391j, & a_{52} &= 0.1684 + 0.0034j,\\
a_{13} &= 0.0473 - 0.0058j, & a_{33} &= 0.0395 + 0.0283j,\\
a_{53} &= -0.1015 - 0.0196j, & a_{ki} &= 0, \text{otherwise}
\end{aligned} \tag{B.4}$$

The ACI-levels are given in figure B.1 for both the LS- and Crosscorrelation predistorter and the compression is given in figure B.2. The dashed curves show the values for the LS predistorter.

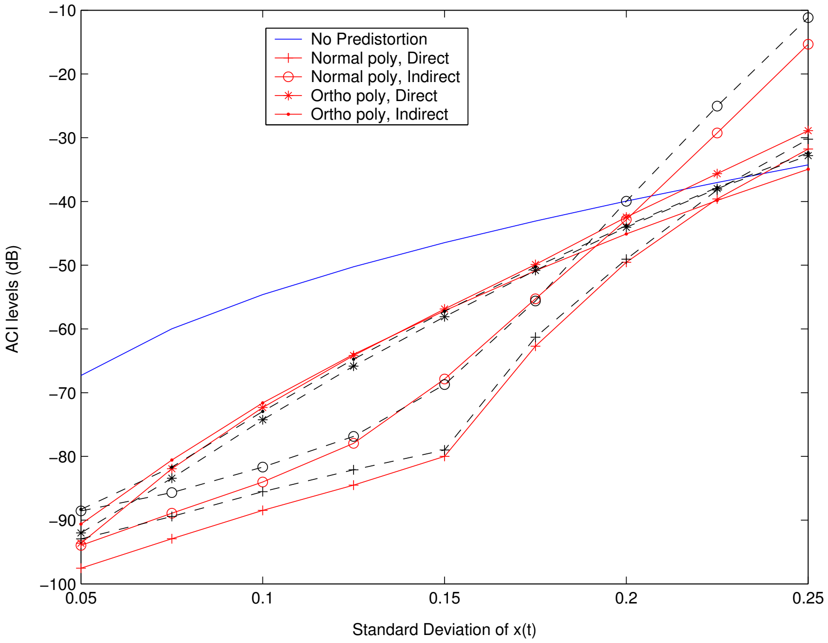


Figure B.1: Performances PA model 1

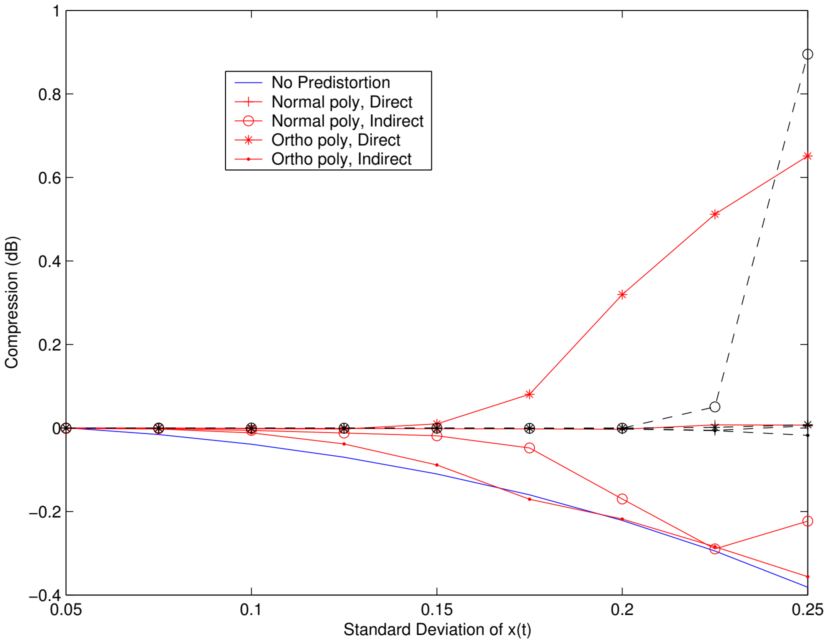


Figure B.2: Compression PA model 1

B.2 PA model 2

PA model 2 is a memory polynomial model also from [64]:

$$y(n) = \sum_{k=1}^{K} \sum_{q=0}^{Q} c_{kq} x(n-q)|x(n-q)|^{k-1} \tag{B.5}$$

The values for the polynomial distortion coefficients are:

$$\begin{aligned}
c_{10} &= 1.0513 + 0.0904j, & c_{30} &= -0.0542 - 0.2900j, \\
c_{50} &= -0.9657 - 0.7028j, & c_{11} &= -0.0680 - 0.0023j, \\
c_{31} &= 0.2234 + 0.2317j, & c_{51} &= -0.2451 - 0.3735j, \\
c_{12} &= 0.0289 - 0.0054j, & c_{32} &= -0.0621 - 0.0932j, \\
c_{52} &= 0.1229 + 0.1508j, & c_{kq} &= 0, \text{otherwise}
\end{aligned} \tag{B.6}$$

The ACI-levels are given in figure B.3 for both the LS- and Crosscorrelation predistorter and the compression is given in figure B.4. The dashed curves show the values for the LS predistorter.

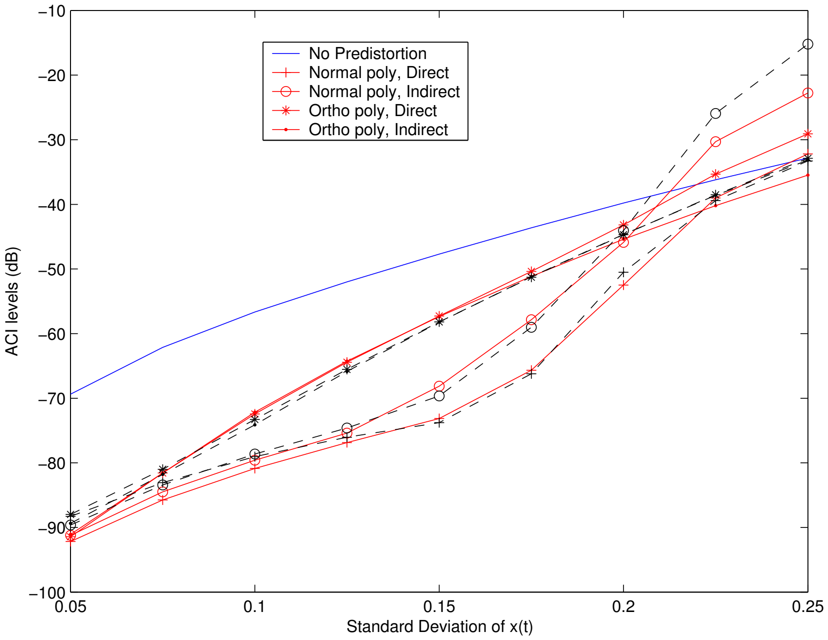


Figure B.3: Performances PA model 2

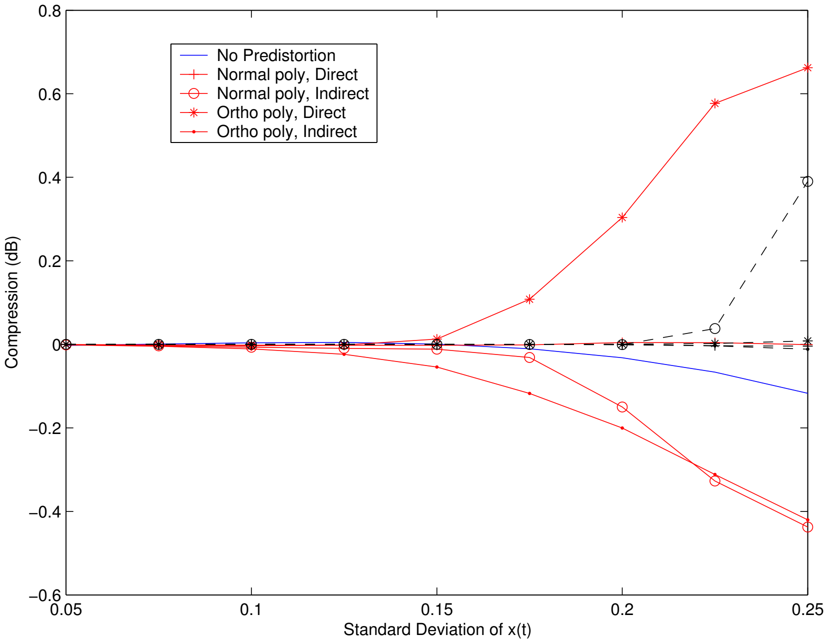


Figure B.4: Compression PA model 2

B.3 PA model 3

PA model 3 is an arctan model from [65]:

$$y(n) = (\alpha_1 \tan^{-1}(\beta_1|x(n)|) + \alpha_2 \tan^{-1}(\beta_2|x(n)|))e^{i\angle x(n)} \qquad \text{(B.7)}$$

where $\alpha_1 = 8.0034 - j4.6116$, $\alpha_2 = -3.7717 + j12.0376$, $\beta_1 = 2.2690$ and $\beta_2 = 0.8234$.

The ACI-levels are given in figure B.5 for both the LS- and Crosscorrelation predistorter and the compression is given in figure B.6. The dashed curves show the values for the LS predistorter.

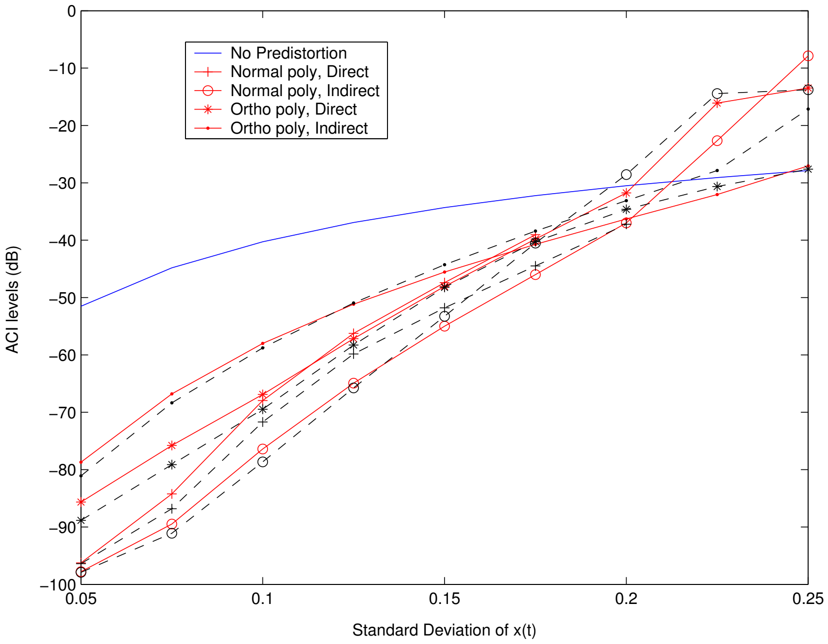


Figure B.5: Performances PA model 3

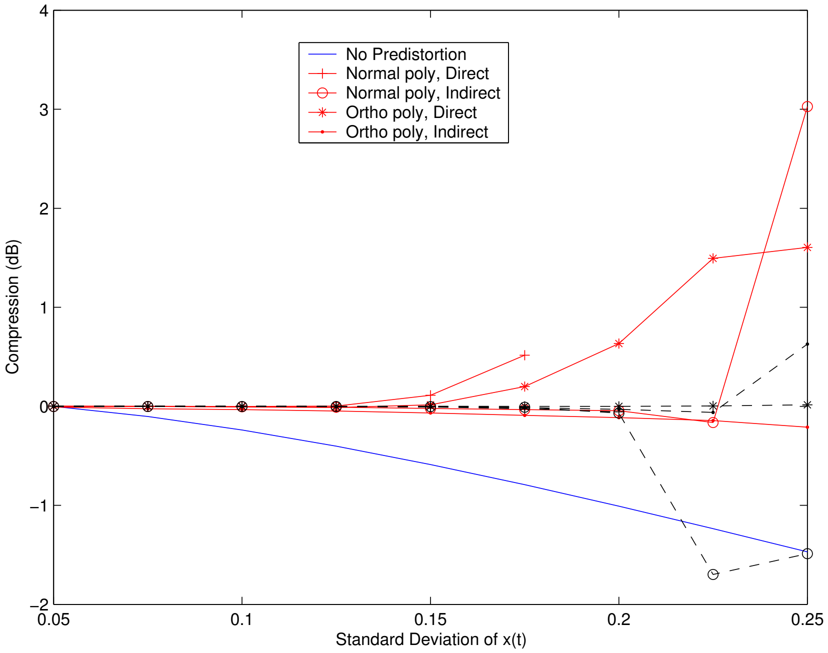


Figure B.6: Compression PA model 3

Appendix C

System timing analysis

In this appendix, the computational effort for the predistortion control of the LS- and Crosscorrelation predistorters is estimated when executed on a Texas Instruments C67x Floating Point Digital Signal Processor. We assume a typical but representative application: UMTS with sufficient resolution in the feedback path. The number of samples on which an estimate of the PA non-linearity is based equals 8k samples. The sampling frequency equals 49 MHz.

C.1 Computations LS predistorter

The task that has to be executed by the DSP is to calculate an estimate ($\hat{\mathbf{a}}$) of the PA non-linearity, based on 8k input samples. This activity is described as (expression 4.37): $\hat{\mathbf{a}} = (\mathbf{\Gamma}^H\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^H\mathbf{x}_2$. Since we use 8k samples, a memory depth of two ($\tau = 0, 1$) and only odd polynomials up to the fifth degree ($k = 1, 3, 5$), $\mathbf{\Gamma}$ is an 8k × 6 matrix and $\mathbf{x}_2$ is an 8k-element vector. The result is the 6-element vector $\hat{\mathbf{a}}$.

The execution time is basically determined by two operations: the matrix product $\mathbf{\Gamma}^H\mathbf{\Gamma}$ and the matrix-vector product $\mathbf{\Gamma}^H\mathbf{x}_2$. We ignore the matrix inversion because the dimensions of the resulting square matrix are small compared to the dimensions of the original matrices. The dimensions of the matrices and the vector within these operations are respectively $(6 \times 8\text{K}) \times (8\text{K} \times 6)$ and $(6 \times 8\text{K}) \times (8\text{K} \times 1)$. The resulting dimensions are respectively (6×6) and (6×1). The execution times of operations on the resulting matrix and vector can be neglected compared to the execution times of the first two operations because of the small dimensions. The number of cycles needed for both operations are determined, using the Texas Instruments C67x DSP Benchmarks. For calculating the number of processor cycles for both the $\mathbf{\Gamma}^H\mathbf{\Gamma}$ and $\mathbf{\Gamma}^H\mathbf{x}_2$ operations we use the 'SP Complex Matrix Multiplication' benchmark ([76]). The multiplication of a $(r1{\times}c1){\times}(c1{\times}c2)$ matrix requires $2 \times r_1 \times c_1 \times c_2 + 33$ cycles. The $\mathbf{\Gamma}^H\mathbf{\Gamma}$ operation requires $2 \times 6 \times 8\text{K} \times 6 + 33$ cycles. The $\mathbf{\Gamma}^H\mathbf{x}_2$ operation requires $2 \times 6 \times 8\text{K} \times 1 + 33$ cycles. The results are displayed in table C.1.

Operation	Number of cycles
$\mathbf{\Gamma}^H\mathbf{\Gamma}$	589857
$\mathbf{\Gamma}^H\mathbf{x}_2$	98337
total	688194

Table C.1: Number of TI C67x DSP cycles for the LS predistorter

For a processor, running at 300 MHz (3.33 ns cycle time) the execution of the LS predistortion algorithm takes approximately 2.3 ms.

C.2 Computations Crosscorrelation predistorter

For the Crosscorrelation predistorter, the tasks that have to be executed by the Texas Instruments C67x Floating Point Digital Signal Processor are:

- Applying the Hanning taper (expressions 5.17 and 5.18).
- Applying the Fast Fourier transform (also expressions 5.17 and 5.18).
- Normalizing the spectra (expressions 5.21 and 5.24).
- Calculating the least squares solution $\hat{\mathbf{a}}$ (expression 5.26).

For all four tasks, we will estimate the computational effort separately.

- Applying the Hanning taper. To estimate the computational effort to apply the Hanning taper on the 7 crosscorrelation functions, we use the 'SP Vector Multiplication' benchmark ([76]). This benchmark assumes real vectors. The Hanning taper values are real but the crosscorrelation functions contain complex values. We therefore use two times the benchmark value. The total processing effort becomes: $7{\times}(2{\times}(2{\times}\text{floor}(\frac{n-1}{2}))+18)$ cycles. In our application, $n = 64$ leading to $7 \times (2 \times (2 \times 31) + 18) = 994$ cycles.

- The 7 tapered crosscorrelation functions are transformed into the frequency domain by means of a 64-point Fast Fourier Transform (FFT). We use the 'SP Complex DIT FFT (radix2)' benchmark ([76]):

  $7 \times ((2 \times n \times {}^2\log n) + 42)$ cycles which leads to $7 \times ((2 \times 64 \times 6) + 42) =$ 5670 cycles.

- The normalization of the spectra is realized by multiplication of the spectra with the element-wise reciprocals of a vector. So, we have to calculate the reciprocals and then we have to apply them. The calculation of the reciprocals of a complex value involves 4 real multiplications and the calculation of the reciprocal of a real value. For the real multiplications, we again use the 'SP vector Multiplication' benchmark and for the reciprocal of the real values, we use the 'SP Vector Reciprocal' benchmark ([76]). The total required number of cycles becomes:

  $(4 \times (2 \times \text{floor}(\frac{n-1}{2}) + 18) + (8 \times \text{floor}(\frac{n-1}{4}) + 53) = 4 \times (2 \times 31 + 18) + (8 \times 15 + 53) = 493$ cycles

  Applying the complex reciprocals to a complex spectrum involves 4 'SP Vector Multiplications'. This is repeated for 6 spectra:

  $6 \times 4 \times (2 \times \text{floor}(\frac{n-1}{2}) + 18) = 6 \times 4 \times (2 \times 31 + 18) = 1920$ cycles.

  The total number of cycles for the normalization equals 2413 cycles.

- The non-linearity of the PA is determined by calculating the least squares solution: $\hat{\mathbf{a}} = (\mathbf{G}^H\mathbf{G})^{-1}\mathbf{G}^H\mathbf{g}$ (expression 5.26). As for the LS predistorter, we will only consider the products $\mathbf{G}^H\mathbf{G}$ and $\mathbf{G}^H\mathbf{g}$ in our calculations since, after these multiplications, the dimensions of the resulting matrix and vector are reduced to such an extend that we can neglect the computational effort for the remaining calculations. The calculations involved are a $(6 \times 64) \times (64 \times 6)$ matrix multiplication and an $(6 \times 64) \times (64 \times 1)$ matrix-vector multiplication. For both products, we use the 'SP Complex Matrix Multiplication benchmark' ([76]): the multiplication of a $(r1 \times c1)$ matrix with a $(c1 \times c2)$ matrix requires $2 \times r_1 \times c_1 \times c_2 + 33$ cycles. The $\mathbf{G}^H\mathbf{G}$ operation requires $2 \times 6 \times 64 \times 6 + 33 = 4641$ cycles. The $\mathbf{G}^H\mathbf{g}_2$ operation requires $2 \times 6 \times 64 \times 1 + 33 = 801$ cycles. The total number of cycles for determining the least squares solution equals 5442 cycles.

An overview of number of cycles required by the Texas Instruments C67x Floating Point Digital Signal Processor for the different tasks within the predistortion control block of the Crosscorrelation predistorter, is given in table C.2.

Operation	Number of cycles
Hanning taper	994
FFT	5670
Normalization	2413
$\mathbf{G}^H\mathbf{G}$ and $\mathbf{G}^H\mathbf{g}$	5442
total	14519

Table C.2: Number of TI C67x DSP cycles for the Crosscorrelation predistorter

For a DSP, running at 300 MHz (3.33 ns cycle time) this amounts to 48 μs.

Appendix D

Quantization of noise and periodic signals

A model for determining the crosscorrelation function of partially correlated noise is presented[1]. In this model a strong interferer, represented by a periodic signal common to both channels of a correlator, is included. A general expression for the correlation function is deduced. This expression is checked against known results in case only noise is present at the Analog-to-Digital Converter inputs and in case only a sinusoidal signal is present. Furthermore, an expression is deduced in case two quantizers with different resolutions are used to sample noise. Estimates of the errors, introduced by quantization, are given and an example of a power spectrum is calculated and compared with the result of a simulation.
The results presented in this appendix can be used to analyze quantization effects in crosscorrelators. Crosscorrelators are applied in a diversity of applications ranging from Radio Astronomy Receivers to Crosscorrelation predistorters for Power Amplifiers. Expressions are derived keeping the Radio Astronomy application in mind. The expressions, used to analyze the effects of quantization in Crosscorrelation predistorters are simplifications of the expressions for the Radio Astronomy application.

D.1 Introduction

Within Radio Astronomy, the radio window of the earth's atmosphere (roughly from 30 MHz to 300 GHz) is used for observations of celestial objects. In

[1] Large parts of this appendix have been published in publication [9]

general, the power of the radiation received from these objects is below the power of the noise produced in the analog parts of a Radio Astronomy Receiver (RA Receiver). A technique to reduce the effects of the noise is to crosscorrelate signals within a multiple-telescope RA Receiver. This way, signals up to 70 dB below the noise level can be detected.

Interference, common to all telescopes, remains visible after crosscorrelation. As a result of the non-linearity of the Analog-to-Digital Converter (ADC), harmonics of the interference are present as well, contaminating the whole spectrum. RA Receivers have been designed to have sufficient selectivity in the analog paths to the ADCs to remove interference. Because of the increasing bandwidth of the ADCs and the growing use of the electro magnetic spectrum, the condition that the signals at the input of the ADCs do not contain interference is no longer satisfied. For the design of modern RA Receivers, it is therefore necessary to be able to analyze the effect of narrowband interference for different resolutions of the ADC.

The effects of quantization for the case no interference is present is studied in [85] and [86]. This work is strongly related to the analysis of quantizing systems using nonsubtractive dither [75], where the measured correlation coefficients equal the nonsubtractive dithered second order moments.

The crosscorrelation of noise and periodic signals has been studied in [92] but only in case of extreme clipping and a small Signal-to-Noise Ratio (SNR). Multi-bit quantization and arbitrary SNRs are not considered. Multi-bit quantization of sinusoidal signals has been studied intensively (see [93] and [94]) but the developed theory is not applicable to correlation configurations.

Within this appendix, a generic model of a crosscorrelator for different types of input signals, is given. A novel general expression for the measured crosscorrelation function in case of multi-bit quantization of a signal containing noise and a periodic signal, is found. This expression is elaborated in case the periodic signal is a sinusoidal signal. Only quantization effects and no sampling effects are considered. We check the expression against known results and present estimates of the errors introduced in the general case. We conclude with an example of a calculated and simulated spectrum.

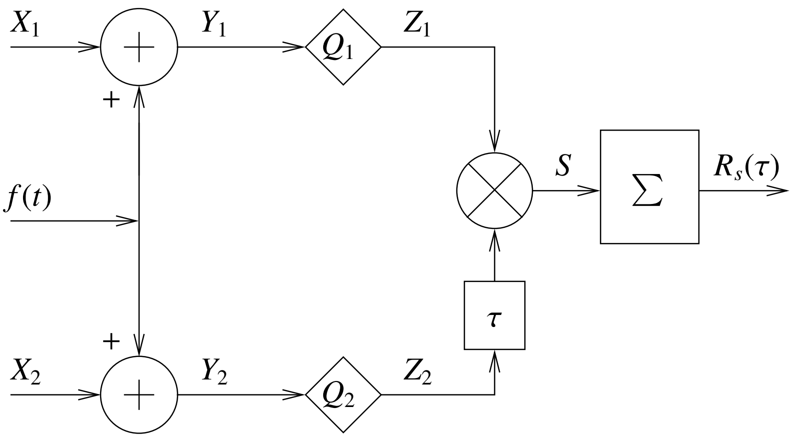


Figure D.1: A general correlator model

D.2 A general correlator model

D.2.1 The correlation function

The correlation function of a sine wave plus noise after extreme clipping has been determined in [92]. Based on that analysis, a general correlator model is presented in figure D.1.

There are 2 wide-sense stationary noise signals X_1 and X_2. A periodic signal $f(t)$ is added to both X_1 and X_2. The resulting signals Y_1 and Y_2 are quantized by Q_1 and Q_2 respectively and crosscorrelated.

Because X_1 and X_2 are wide-sense stationary, the noise inputs can be described with a bivariate normal distribution. Zero mean ($\mu = 0$) and equal power are assumed. Without loss of generality, it is assumed that the power for both X_1 and X_2 equals one ($\sigma^2 = 1$). If we consider the signals X_1 and X_2 at the respective times t and $t - \tau$, the correlation equals $\rho(\tau)$. Signals Y_1 and Y_2 are quantized by a b-bit mid-riser quantizer with unit step size:

$$Q(x) = \left(\sum_{i=-2^{b-1}+1}^{2^{b-1}-1} \Phi(gx - i) \right) - 2^{b-1} + \frac{1}{2} \tag{D.1}$$

Φ is Heaviside's step function (see expression 3.6) and g determines the granularity of the quantization process. After quantization, the signals Z_1

and Z_2 are multiplied and the product is averaged. We define the time-dependent correlation function $R(t, \tau)$ as the ensemble average of $Z_1(t)$ and $Z_2(t - \tau)$:

$$R(t, \tau) \;=\; < S > \;=\; < Z_1(t) \cdot Z_2(t - \tau) > \tag{D.2}$$

The correlation function $R_s(\tau)$ equals the time average over one cycle of the periodic signal with period T_p (see [92]). If the signal after multiplication is sampled and integrated, $R_s(\tau)$ is described by:

$$R_s(\tau) = \frac{1}{T_p} \sum_{t=0}^{T_p - 1} R(t, \tau) \tag{D.3}$$

D.2.2 Splitting the noise sources

As is done in [88], the two signals X_1 and X_2 (see figure D.1) can be modeled by 3 uncorrelated Gaussian noise sources N_0, N_1 and N_2, all having zero mean and $\sigma = 1$:

$$X_1 \;=\; \sqrt{1 - \mid \rho(\tau) \mid} N_1 + \sqrt{\mid \rho(\tau) \mid} N_0 \tag{D.4}$$

$$X_2 \;=\; \sqrt{1 - \mid \rho(\tau) \mid} N_2 + \text{sign}(\rho(\tau)) \sqrt{\mid \rho(\tau) \mid} N_0 \tag{D.5}$$

The sign(ρ) operator is defined as:

$$\text{sign}(\rho) = \begin{cases} -1 & \text{if } \rho < 0 \\ 1 & \text{otherwise} \end{cases} \tag{D.6}$$

Because of the sign operator in expression D.5, negative correlations between X_1 and X_2 are allowed as well. The periodic signal $f(t)$ is added to both X_1 and X_2. The components, due to N_0 and f, are defined as displacements:

$$\Delta_1(n_0, t, \tau) \;=\; \sqrt{\mid \rho(\tau) \mid} n_0 + f(t) \tag{D.7}$$

$$\Delta_2(n_0, t, \tau) \;=\; \text{sign}(\rho(\tau)) \sqrt{\mid \rho(\tau) \mid} n_0 + f(t - \tau) \tag{D.8}$$

This leads to the following expressions for Y_1 and Y_2:

$$\begin{aligned} Y_1 &= \sqrt{1- \mid \rho(\tau) \mid} N_1 + \Delta_1(n_0, t, \tau) && \text{(D.9)} \\ Y_2 &= \sqrt{1- \mid \rho(\tau) \mid} N_2 + \Delta_2(n_0, t, \tau) && \text{(D.10)} \end{aligned}$$

For both quantizers, the input is the sum of two stochastic signals. Signals due to N_1 and N_2 are uncorrelated and have power $1- \mid \rho(\tau) \mid$ while the displacements Δ_1 and Δ_2 depend on noise source N_0. The power of N_0 equals one. The time-dependent correlation function $R(t, \tau)$ can be written as the ensemble average over the three noise sources N_0, N_1 and N_2:

$$\begin{aligned} R(t, \tau) = &\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{N_0}(n_0) f_N(n_1) f_N(n_2) \\ &\cdot Q(n_1 + \Delta_1(n_0, t, \tau)) \cdot Q(n_2 + \Delta_2(n_0, t, \tau)) \\ &\cdot dn_2 dn_1 dn_0 \end{aligned} \tag{D.11}$$

where

$$\begin{aligned} f_{N_0}(n) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}n^2} && \text{(D.12)} \\ f_N(n) &= \frac{1}{\sqrt{2\pi(1- \mid \rho(\tau) \mid)}} e^{-\frac{1}{2}\left(\frac{n}{\sqrt{1-|\rho(\tau)|}}\right)^2} && \text{(D.13)} \end{aligned}$$

By rearranging expression D.11, two gain functions G_1 and G_2 can be defined:

$$R(t, \tau) = \int_{-\infty}^{\infty} f_{N_0}(n_0).G_1(n_0, t, \tau).G_2(n_0, t, \tau) \cdot dn_0 \tag{D.14}$$

where

$$G_1(n_0, t, \tau) = \int_{-\infty}^{\infty} f_N(n_1).Q\big(n_1 + \Delta_1(n_0, t, \tau)\big) \cdot dn_1 \tag{D.15}$$

$$G_2(n_0, t, \tau) = \int_{-\infty}^{\infty} f_N(n_2).Q\big(n_2 + \Delta_2(n_0, t, \tau)\big) \cdot dn_2 \tag{D.16}$$

D.2.3 Elaborating G_1

The quantization staircase consists of an ideal transfer function and an error (see [94]):

$$Q(x) = gx + \sum_{m=1}^{\infty} \frac{1}{m\pi} \sin(2m\pi gx) \tag{D.17}$$

Using this in (D.15), using standard rules for integration and trigonometry [95] and inserting (D.7), the expression for G_1 can be written as

$$\begin{aligned} G_1(n_0, t, \tau) =& gf(t) + g\sqrt{|\,\rho(\tau)\,|}n_0 \\ &+ \sum_{m=1}^{\infty} \frac{1}{m\pi} e^{-2\pi^2 m^2 g^2 (1-|\rho(\tau)|)} \\ &\cdot \left[\left(\cos\left(2\pi mg\sqrt{|\,\rho(\tau)\,|}n_0\right) \cdot \sin\left(2\pi mgf(t)\right) \right) \right. \\ &+ \left. \left(\sin\left(2\pi mg\sqrt{|\,\rho(\tau)\,|}n_0\right) \cdot \cos\left(2\pi mgf(t)\right) \right) \right] \end{aligned} \tag{D.18}$$

A similar expression can be found for G_2 where ' $\sqrt{|\,\rho(\tau)\,|}$' has to be replaced by 'sign($\rho(\tau)$) $\sqrt{|\,\rho(\tau)\,|}$' and '$f(t)$' by '$f(t-\tau)$'.

D.2.4 Final expression

Using the expressions for G_1 and G_2, $R(t, \tau)$ (see D.14) can be calculated:

$$\begin{aligned}
R(t,\tau) = \quad & g^2 f(t) f(t-\tau) + \quad g^2\rho(\tau) \\
+ \quad & g f(t) \sum_{m_2=1}^{\infty} \frac{1}{m_2\pi} e^{-2\pi^2 g^2 m_2^2} \cdot \sin(2\pi m_2 g f(t-\tau)) \\
+ \quad & g f(t-\tau) \sum_{m_1=1}^{\infty} \frac{1}{m_1\pi} e^{-2\pi^2 g^2 m_1^2} \cdot \sin(2\pi m_1 g f(t)) \\
+ \quad & \sum_{m_2=1}^{\infty} 2g^2\rho(\tau) e^{-2\pi^2 m_2^2 g^2} \cdot \cos(2\pi m_2 g f(t-\tau)) \\
+ \quad & \sum_{m_1=1}^{\infty} 2g^2\rho(\tau) e^{-2\pi^2 m_1^2 g^2} \cdot \cos(2\pi m_1 g f(t)) \\
+ \quad & \sum_{m_1=1}^{\infty} \sum_{m_2=1}^{\infty} \frac{1}{m_1 m_2 \pi^2} e^{-2\pi^2 (m_1^2+m_2^2) g^2} \cdot \\
& \big[\sin(2\pi m_1 g f(t)) \sin(2\pi m_2 g f(t-\tau)) \cdot \\
& \cosh(4\pi^2 g^2 \rho(\tau) m_1 m_2) + \\
& \cos(2\pi m_1 g f(t)) \cos(2\pi m_2 g f(t-\tau)) \cdot \\
& \sinh(4\pi^2 g^2 \rho(\tau) m_1 m_2)\big]
\end{aligned} \tag{D.19}$$

If the input signal is defined as $f(t) = a\sin(\Psi(t))$, the following expressions can be used to rewrite expression D.19:

$$\sin(z\sin(\Psi)) = 2 \sum_{p=1,odd}^{\infty} J_p(z)\sin(p\Psi) \tag{D.20}$$

$$\cos(z\sin(\Psi)) = J_0(z)\cos(\Psi) + 2 \sum_{p=2,even}^{\infty} J_p(z)\cos(p\Psi) \tag{D.21}$$

$J_p(z)$ is the p-th order Bessel function of the first kind. $R_s(\tau)$ can then be expressed as:

$$\begin{aligned} R_s(\tau) =& (1+\epsilon_\rho)g^2\rho(\tau) + (1+\epsilon_f)\frac{g^2a^2}{2}\cos(\Psi(\tau))+ \\ & \frac{1}{2}\epsilon^e_{p=0} + \sum_{p=1,odd}^{\infty} \epsilon^o_p \cos(p\Psi(\tau))+ \\ & \sum_{p=2,even}^{\infty} \epsilon^e_p \cos(p\Psi(\tau)) \end{aligned} \tag{D.22}$$

where

$$\begin{aligned} \epsilon_\rho =& 4\sum_{m=1}^{\infty} J_0(2\pi mga)e^{-2\pi^2m^2g^2} \\ \epsilon_f =& \frac{4}{ga}\sum_{m=1}^{\infty}\frac{1}{m\pi}J_1(2\pi mga)e^{-2\pi^2m^2g^2} \\ \epsilon^e_p =& \sum_{m_1=1}^{\infty}\sum_{m_2=1}^{\infty}\frac{2}{m_1m_2\pi^2}J_p(2\pi m_1ga)J_p(2\pi m_2ga) \\ & e^{-2\pi^2(m_1^2+m_2^2)g^2}.\sinh(4\pi^2g^2m_1m_2\rho(\tau)), p=0,2,4.... \\ \epsilon^o_p =& \sum_{m_1=1}^{\infty}\sum_{m_2=1}^{\infty}\frac{2}{m_1m_2\pi^2}J_p(2\pi m_1ga)J_p(2\pi m_2ga) \\ & e^{-2\pi^2(m_1^2+m_2^2)g^2}.\cosh(4\pi^2g^2m_1m_2\rho(\tau)), p=1,3,5.... \end{aligned}$$

An interpretation of the distortion coefficients ϵ is that ϵ_ρ and $\epsilon^e_{p=0}$ represent the distortion of the correlated noise part. ϵ_f represents the distortion of the sinusoidal signal. ϵ^e_p for $p \neq 0$ and ϵ^o_p represent a combined effect.

An alternative approach to calculate the crosscorrelation function in case no interference is present ($f(t) = 0, \forall t$), is to use the second-order joint-moment given as:

$$E[Z_1, Z_2] = (1 + \delta(\mu_1, \mu_2))E[Y_1, Y_2] + \Delta\mu_{12}(\mu_1, \mu_2) \tag{D.23}$$

For the elaborated version of this formula we refer to [96]. The gain of the system is influenced by δ and the error is reflected by $\Delta\mu_{12}$. Both δ and $\Delta\mu_{12}$ are defined in [96] and depend on μ_1 and μ_2, the expected values of Y_1 and Y_2 respectively. The more intuitive analysis presented above, equals the analysis of the second-order joint-moments when $\mu_1 = \mu_2 = 0$.

D.3 Case studies

With the developed model, different cases can be analyzed. The first case is single-bit quantization of noise. In the second case, an expression for the correlation function is given in case a noise signal is quantized by quantizers with different resolutions. The third case is multi-bit quantization of a sinusoidal signal without noise. In the fourth case, estimates of the quantization errors ϵ are given and at the end an example of a calculated spectrum is compared with a simulated result.

D.3.1 Single-bit quantization of noise

Single-bit quantization of noise occurs if no sinusoidal signal is present ($a = 0$) and the scaling factor before quantization is small ($g << 1$). For small g, the RMS value of the noise signal at the input of the ADC is much less than 1 LSB so, as an approximation, only the least significant bit contains information about the input signal. The expression for $R_s(\tau)$ then becomes:

$$R_s(\tau) = (1 + \epsilon_\rho) g^2 \rho(\tau) + \frac{1}{2}\epsilon^e_{p=0} \qquad \text{(D.24)}$$

with

$$\begin{aligned}\epsilon_\rho =& 4 \sum_{m=1}^{\infty} e^{-2\pi^2 m^2 g^2} \\ \epsilon^e_{p=0} =& \sum_{m_1=1}^{\infty}\sum_{m_2=1}^{\infty} \frac{2}{m_1 m_2 \pi^2} e^{-2\pi^2(m_1^2+m_2^2)g^2} \\ & \sinh(4\pi^2 g^2 m_1 m_2 \rho(\tau))\end{aligned}$$

Because, in this situation, there is a one-to-one relation between the correlation of the signals before quantization ($\rho(\tau)$) and the correlation function after quantization ($R_s(\tau)$), we could plot R_s as a function of ρ. R_s is approximated by $R_s^{'}$ where the summations over m, m_1 and m_2 run from 1 to 50. Because the output values of the ADC are $-\frac{1}{2}$ and $\frac{1}{2}$, $R_s^{'}$ ranges from 0 to 0.25 while ρ ranges from 0 to 1. It is more convenient to normalize $R_s^{'}$:

$$\rho_s(\rho) = 4R_s^{'}(\rho) \qquad \text{(D.25)}$$

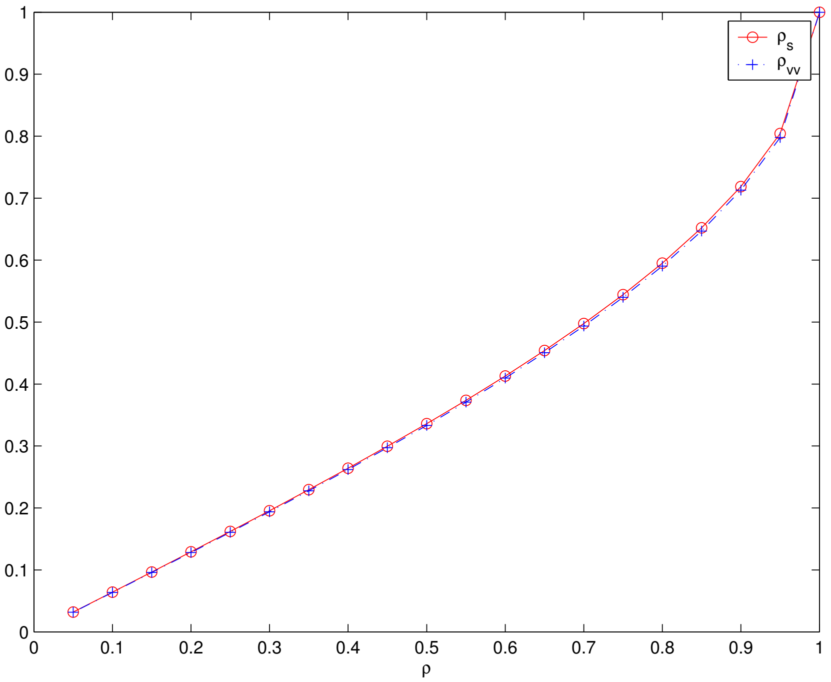


Figure D.2: Approximated and theoretical Correlation Coefficients (ρ_s and ρ_{vv} resp.) versus real Correlation Coefficient (ρ)

In [15] and [97], the correlation coefficient in case of single-bit quantizers, is determined analytically. This correlation coefficient ρ_{vv} equals:

$$\rho_{vv}(\rho) = \frac{2}{\pi}\arcsin(\rho) \tag{D.26}$$

This 'van Vleck relation' should equal expression D.24 (via D.25). However, expression D.25 can not be converted into expression D.26 in a straightforward way. We therefore calculate $\rho_s(\rho)$ and $\rho_{vv}(\rho)$ separately and compare the results. In figure D.2, both the approximated correlation after quantization ($\rho_s(\rho)$) and the theoretical correlation after quantization ($\rho_{vv}(\rho)$) are plotted against the correlation before quantization (ρ) where $g = 0.1$.

We see in figure D.2 that ρ_s is a good approximation of the theoretical correlation ρ_{vv}. If we compare ρ_s with ρ_{vv}, the error is largest for fully correlated signals ($\rho = 1$). The error then equals $1.6.10^{-2}$ and can be reduced by increasing the maximum values for m, m_1 and m_2. For small correlation coefficients ($\rho < \frac{1}{2}$), the error is smaller than 10^{-10}.

D.3.2 Quantization of noise with different resolutions

When sampling noise, we can derive an expression for the crosscorrelation function in case the quantizers Q_1 and Q_2 have different resolutions. Suppose the resolution of Q_1 is determined by g_1 and the resolution of Q_2 by g_2. The resulting expression for the time dependent correlation function $R(t, \tau)$ then equals:

$$R(t,\tau) = \int_{-\infty}^{\infty} f_{N_0}(n_0).G_1(n_0,t,\tau).G_2(n_0,t,\tau) \cdot dn_0 \tag{D.27}$$

where

$$\begin{aligned} G_1(n_0,t,\tau) =& g_1 \sqrt{|\,\rho(\tau)\,|} n_0 \\ &+ \sum_{m=1}^{\infty} \frac{1}{m\pi} e^{-2\pi^2 m^2 g_1^2 (1-|\rho(\tau)|)}. \\ &\sin\left(2\pi m g_1 \sqrt{|\,\rho(\tau)\,|}\, n_0\right) \\ G_2(n_0,t,\tau) =& g_2 \cdot \mathrm{sign}(\rho(\tau)) \sqrt{|\,\rho(\tau)\,|} n_0 \\ &+ \sum_{m=1}^{\infty} \frac{1}{m\pi} e^{-2\pi^2 m^2 g_2^2 (1-|\rho(\tau)|)}. \\ &\sin\left(2\pi m g_2 \cdot \mathrm{sign}(\rho(\tau)) \sqrt{|\,\rho(\tau)\,|}\, n_0\right) \end{aligned}$$

Using these expressions, $R_s(\tau)$ can then be written as:

$$R_s(\rho(\tau)) = (1 + \epsilon_1) g_1 g_2 \rho(\tau) + \frac{1}{2}\epsilon_2 \tag{D.28}$$

where

$$\begin{aligned} \epsilon_1 =& \sum_{m=1}^{\infty} 2(e^{-2\pi^2 m^2 g_1^2} + e^{-2\pi^2 m^2 g_2^2}) \\ \epsilon_2 =& \sum_{m_1=1}^{\infty} \sum_{m_2=1}^{\infty} \frac{2}{m_1 m_2 \pi^2} e^{-2\pi^2 (m_1^2 g_1^2 + m_2^2 g_2^2)}. \\ &\sinh(4\pi^2 g_1 g_2 m_1 m_2 \rho(\tau)) \end{aligned}$$

This expression resembles expression D.24 to a large extend; the g-terms are 'distributed equally' over g_1 and g_2.

D.3.3 Multi-bit quantization of a sinusoidal signal without noise

This situation occurs if the noise powers before quantization (and after scaling with g) are negligibly small. This means that g is small, and the power of the sinusoidal signal ($\frac{1}{2}g^2a^2$) is non-zero. The measured correlation function then becomes:

$$R_s(\tau) \approx (1+\epsilon_f)\frac{g^2a^2}{2}\cos(\Psi(\tau)) + \sum_{p=1,odd}^{\infty} \epsilon_p^o \cos(p\Psi(\tau)) \tag{D.29}$$

where

$$\begin{aligned}\epsilon_f &\approx \frac{4}{ga}\sum_{m=1}^{\infty}\frac{1}{m\pi}J_1(2\pi mga)\\ \epsilon_p^o &\approx \sum_{m_1=1}^{\infty}\sum_{m_2=1}^{\infty}\frac{2}{m_1m_2\pi^2}J_p(2\pi m_1ga)J_p(2\pi m_2ga) = \left(\sum_{m=1}^{\infty}\frac{\sqrt{2}}{m\pi}J_p(2\pi mga)\right)^2\end{aligned}$$

The amplitude of the fundamental of the sinusoidal signal $\cos(\Psi(\tau))$ becomes:

$$\begin{aligned}&\frac{1}{2}\left[g^2a^2 + 2ga\sum_{m=1}^{\infty}\frac{2}{m\pi}J_1(2\pi mga) + \left(\sum_{m=1}^{\infty}\frac{2}{m\pi}J_1(2\pi mga)\right)^2\right] =\\ &\frac{1}{2}\left(ga + \sum_{m=1}^{\infty}\frac{2}{m\pi}J_1(2\pi mga)\right)^2\end{aligned} \tag{D.30}$$

For the harmonic signals ($p > 1$), the amplitudes of $\cos(p\Psi(\tau))$ equal ϵ_p^o. The amplitudes found this way equal half the squared gain factors found in [94]. This is to be expected since in [94], the output amplitude of a single ADC as a function of time is considered while in this appendix we are interested in the correlation function of this signal, giving power.

D.3.4 Multi-bit quantization of a sinusoidal signal with partially correlated noise

D.3.4.1 Estimation of the errors ϵ_ρ, ϵ_f, ϵ^e and ϵ^o

In this section, estimates of ϵ_ρ, ϵ_f, ϵ^e and ϵ^o are determined.
In the expressions for ϵ_ρ and ϵ_f, $e^{-2\pi^2m^2g^2}$ is an important factor (see expres-

sion D.22). For m^2g^2 large, the ϵ-values become very small: for $m^2g^2 = 2$, $e^{-2\pi^2 m^2 g^2}$ equals $7.2 \cdot 10^{-18}$ which can be neglected. We define:

$$m_{max} = \lceil \left(\frac{2}{g} \right) \tag{D.31}$$

$\lceil(x)$ delivers the smallest integer larger than x. If the summations for ϵ_ρ and ϵ_f are executed until m_{max}, $\epsilon_\rho{}'$ and $\epsilon_f{}'$ are estimates of ϵ_ρ and ϵ_f with a relatively small error.

To analyze ϵ^e and ϵ^o, all m's should be brought into the exponent except the m's within J_p:

$$\begin{aligned}\epsilon_p^e = &\sum_{m_1=1}^{\infty}\sum_{m_2=1}^{\infty} \frac{1}{\pi^2} J_p(2\pi m_1 ga).J_p(2\pi m_2 ga).\\ &\Big(e^{-\ln(m_1)-\ln(m_2)-2\pi^2(m_1^2+m_2^2)g^2+4\pi^2g^2m_1m_2\rho(\tau)}\\ &- e^{-\ln(m_1)-\ln(m_2)-2\pi^2(m_1^2+m_2^2)g^2-4\pi^2g^2m_1m_2\rho(\tau)}\Big)\end{aligned} \tag{D.32}$$

$$\begin{aligned}\epsilon_p^o = &\sum_{m_1=1}^{\infty}\sum_{m_2=1}^{\infty} \frac{1}{\pi^2} J_p(2\pi m_1 ga).J_p(2\pi m_2 ga).\\ &\Big(e^{-\ln(m_1)-\ln(m_2)-2\pi^2(m_1^2+m_2^2)g^2+4\pi^2g^2m_1m_2\rho(\tau)}\\ &+ e^{-\ln(m_1)-\ln(m_2)-2\pi^2(m_1^2+m_2^2)g^2-4\pi^2g^2m_1m_2\rho(\tau)}\Big)\end{aligned} \tag{D.33}$$

For radio astronomical applications, it is realistic to assume that the power of the astronomical signal is less than the noise power generated in the analog part of the receiver. This means that we can assume that $\rho(\tau) \leq 1/2$.

If m_{max} is the upper limit for both m_1 and m_2, the error in the estimates for ϵ_p^e and ϵ_p^o is of the same order as for ϵ_ρ and ϵ_f.

According to [94]:

$$J_p(z) \approx \sqrt{\frac{2}{\pi z}} \cos(z - \frac{p\pi}{2} - \frac{\pi}{4}) \tag{D.34}$$

for $z >>\mid p \mid +1$. If the power of the sinusoidal signal ($\frac{1}{2}g^2a^2$) is large:

$$J_p(2\pi mga) \leq \sqrt{\frac{1}{\pi^2 mga}} \tag{D.35}$$

Note that the approximation of $J_p(2\pi mga)$ cannot have a maximum for all p simultaneously. Nevertheless the approximation is useful since it gives an estimate of the maximum value. The estimates, ϵ'_ρ, ϵ'_f, $\epsilon^{e'}$ and $\epsilon^{o'}$ of the error terms then become:

$$\epsilon'_\rho = 4\sum_{m=1}^{m_{max}} \frac{1}{\sqrt{\pi^2 mga}} e^{-2\pi^2 m^2 g^2} \tag{D.36}$$

$$\epsilon'_f = \frac{4}{ga}\sum_{m=1}^{m_{max}} \frac{1}{m\pi}\frac{1}{\sqrt{\pi^2 mga}} e^{-2\pi^2 m^2 g^2} \tag{D.37}$$

$$\begin{aligned}\epsilon^{e'} =& \sum_{m_1=1}^{m_{max}}\sum_{m_2=1}^{m_{max}} \frac{1}{\pi^2}\cdot\frac{1}{\pi^2 ga}\cdot \\ & \left(e^{-\ln(m_1)-\ln(m_2)-2\pi^2(m_1^2+m_2^2)g^2+4\pi^2g^2m_1m_2}\right. \\ & \left.- e^{-\ln(m_1)-\ln(m_2)-2\pi^2(m_1^2+m_2^2)g^2-4\pi^2g^2m_1m_2}\right)\end{aligned} \tag{D.38}$$

$$\begin{aligned}\epsilon^{o'} =& \sum_{m_1=1}^{m_{max}}\sum_{m_2=1}^{m_{max}} \frac{1}{\pi^2}\cdot\frac{1}{\pi^2 ga}\cdot \\ & \left(e^{-\ln(m_1)-\ln(m_2)-2\pi^2(m_1^2+m_2^2)g^2+4\pi^2g^2m_1m_2}\right. \\ & \left.+ e^{-\ln(m_1)-\ln(m_2)-2\pi^2(m_1^2+m_2^2)g^2-4\pi^2g^2m_1m_2}\right) \\ \approx & \; \epsilon^{e'}\end{aligned} \tag{D.39}$$

D.3.4.2 Evaluation of the errors

In the previous paragraph, approximations of the errors ϵ_ρ, ϵ_f, ϵ^e and ϵ^o are determined. ϵ'_ρ is an estimate of the relative error in the measured correlated power, ϵ'_f is an estimate of the relative error in the measured power of the sinusoidal signal. For ϵ^e and ϵ^o, the estimates are almost equal so $\epsilon^{e'}$ can act as an estimate of the absolute distortion power introduced by the quantization. ϵ'_ρ and ϵ'_f are relative errors (see expression D.22), while $\epsilon^{e'}$ is an absolute power. In order to compare the different ϵ-values, $\epsilon^{e'}$ must be normalized with either g^2 or $\frac{1}{2}g^2a^2$. We choose to normalize with g^2, the scaling factor of the correlation function. In figure D.3, ϵ'_ρ, ϵ'_f and $\epsilon^{e'}/g^2$ are plotted as functions of the RMS noise level at the quantization staircase where $\rho \leq \frac{1}{2}$. An 8 bit sinusoidal signal is present at the input of the ADC ($ga = 128$).

From this figure it can be concluded that the errors, introduced by quantization, rapidly decrease by increasing noise levels at the input of the ADCs.

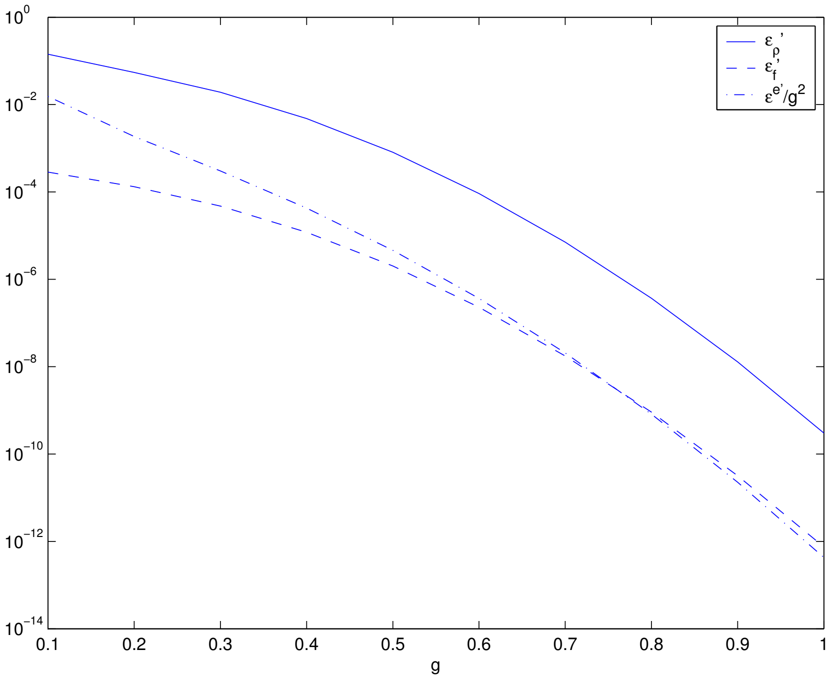


Figure D.3: Errors for $ga = 128$ and RMS noise levels < 1

If an 8 bit sinusoidal signal and 1 LSB noise is present, it is possible to estimate the correlation coefficient and the power of the sinusoidal signal with an accuracy better than 10^{-9} and 10^{-12} respectively and the spectral distortion will be less than 10^{-12}.

D.3.5 An example of a spectrum

As an example, we *calculated* the crosscorrelation of a 2.5 LSB amplitude sinusoidal signal with 1/4 LSB RMS noise at the quantization staircases. The noise at the two ADCs is partially correlated with $\rho(0) = \frac{1}{2}$. We also *simulated* the set-up where we used pseudo-random noise sources to model N_0, N_1 and N_2. The crosscorrelation function is used to determine the cross-power spectral density. The results of both the calculation and the simulation are given in figure D.4.

The bandwidth of the correlated noise signal (N_0) is $\frac{1}{32}$nd of the sampling frequency and the frequency of the sinusoidal signal is $\frac{1}{16}$th of the sampling frequency. It is chosen this way to illustrate the effect that the noise spectrum is repeated around the even harmonics of the sinusoidal signal. The

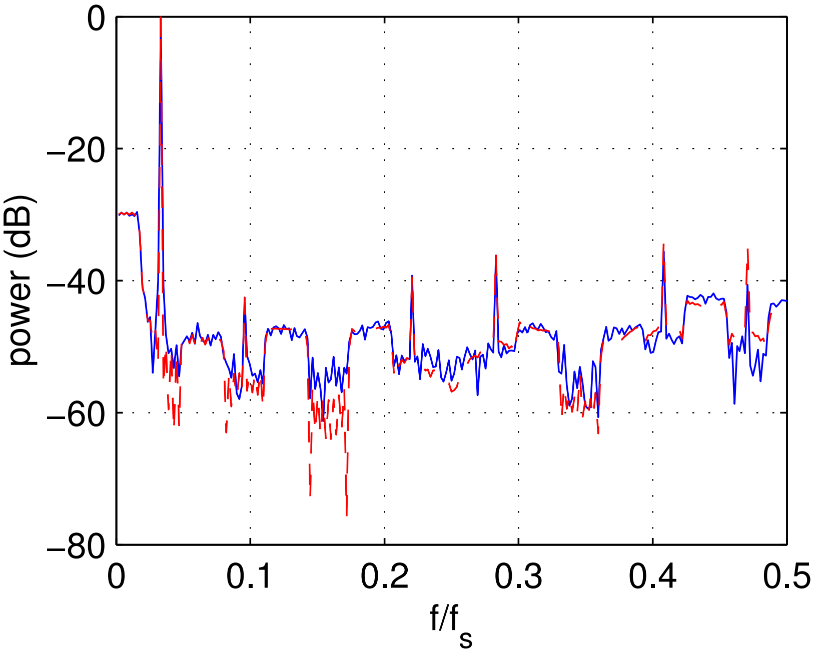


Figure D.4: Normalized Spectrum of a 2.5 LSB amplitude sinusoid plus a $\frac{1}{4}$ LSB noise with a correlation coefficient of 0.5. The solid curve shows the spectrum obtained with a simulation model and the dashed curve shows the calculated spectrum.

bandwidth of the uncorrelated noise sources N_1 and N_2 is half the sampling frequency.

A small difference between the simulated and calculated spectrum can be observed. This is due to the fact that, for practical reasons, sampling *before* multiplication is used in the simulation model. This sampling causes aliasing on both inputs of the multiplier. Amplitudes of sinusoidal signals with frequencies which do not overlap before sampling but which do overlap after sampling, are summed. The derived model assumes sampling only *after* the multiplication. After integration and Fourier transformation, values for frequencies which overlap after sampling, are summed. These values represent power. So, in case of the simulated spectrum, the amplitude of the sinusoidal signals with overlapping frequencies after sampling are summed while in case of the calculated spectrum only the power of these signals is summed. This explains the small difference. By calculating the spectra in cases more noise is added to the channels of the crosscorrelator, the average distortion is decreased which is also known as the 'dithering' effect (see for example [98]).

D.4 Conclusion

A general expression for the correlation function, in case of crosscorrelating multi-bit quantized signals, is determined for input signals consisting of 3 components: noise which is equal for both channels of the correlator (correlated noise), noise which is produced within each of the channels of the correlator (uncorrelated noise) and a common periodic signal. Two extreme cases are: the input signal consists of only noise and the input signal consists of a sinusoidal signal plus uncorrelated noise. For these cases the results were already known and match with the developed expression. All other combinations of the 3 types of signals can also be analyzed using the general expression of the correlation function. In developing the model, the quantization staircase has been decomposed into a linear transfer function and an error-sawtooth. The effects of the error-sawtooth on the periodic signal are analyzed. The effects of the correlated- and uncorrelated noise have been included. The well-known fact that noise smoothes the quantization staircase of an ADC is demonstrated with the above model. Not included in the model is the effect of the uncorrelated noise and the sinusoidal signal on the SNR of the measured correlation function and the effects of the linearity errors of the ADCs. These are subjects for further study.

Appendix E

Distortion plots of PA models

In this appendix, we present the AM-AM and AM-PM distortion plots of:

- Sampled stimulus- and response data of a PA. The data was obtained through measurements on a PA by Philips Semiconductors.
- A PA model constructed by Philips Semiconductors, based on the data mentioned above.
- A memory polynomial constructed by the algorithm used in the LS predistorter, using the stimulus- and response data provided by Philips. The LS predistorter uses orthogonal polynomials and the Direct Learning scheme.
- A memory polynomial constructed by the algorithm used in the Crosscorrelation predistorter, using the stimulus- and response data provided by Philips. The LS predistorter uses orthogonal polynomials and the Direct Learning scheme.

In figure E.1, the AM-AM distortion for the original data and the model provided by Philips are presented.

In figure E.2, the AM-AM distortion of the memory polynomials, determined by the LS- and Crosscorrelation predistorter, are given.

In figure E.3, the AM-PM distortion for the original data and the model provided by Philips are presented.

In figure E.4, the AM-AM distortion of the memory polynomials, determined by the LS- and Crosscorrelation predistorter, are given.

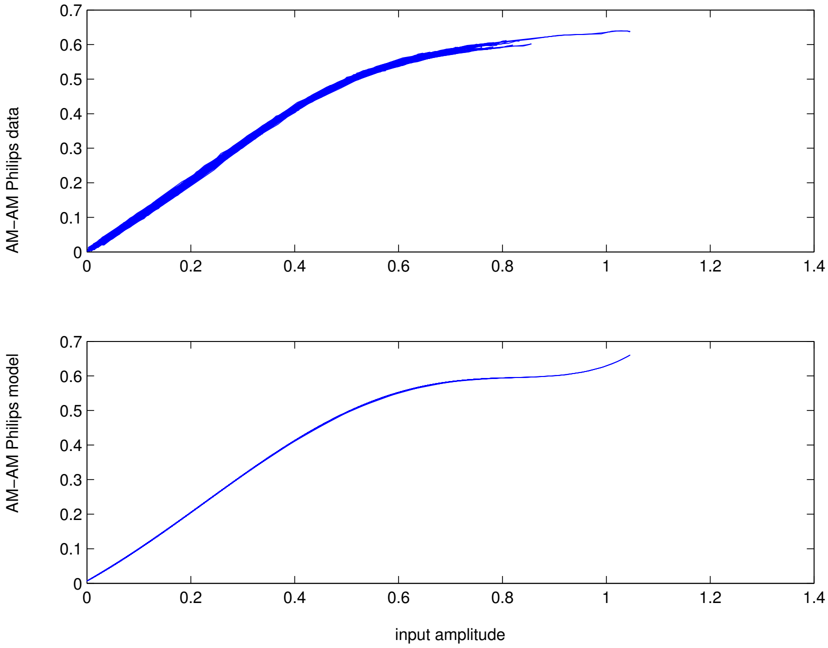


Figure E.1: AM-AM distortion of the Philips data and the Philips model

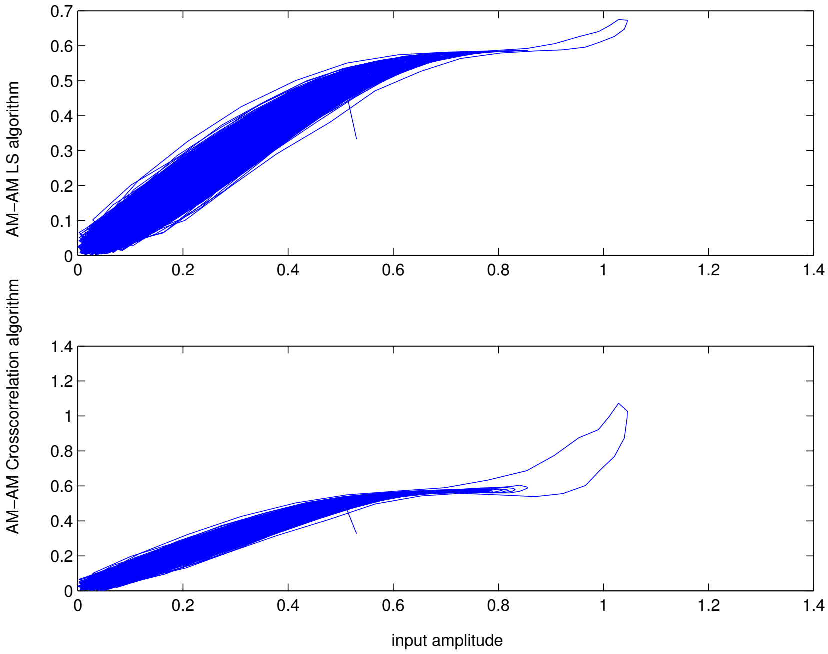


Figure E.2: AM-AM for the PA models based on the LS- and Crosscorrelation predistorters

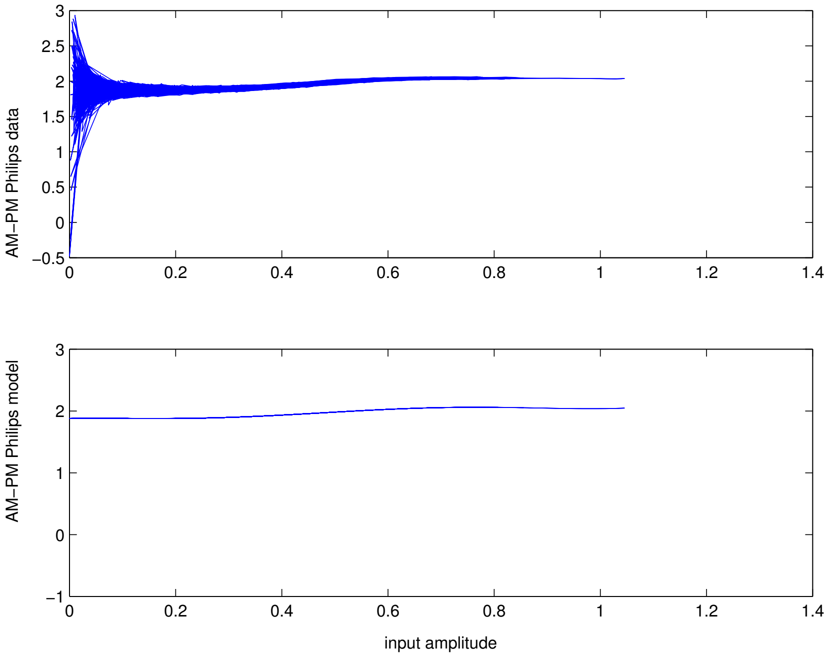


Figure E.3: AM-PM distortion of the Philips data and the Philips model

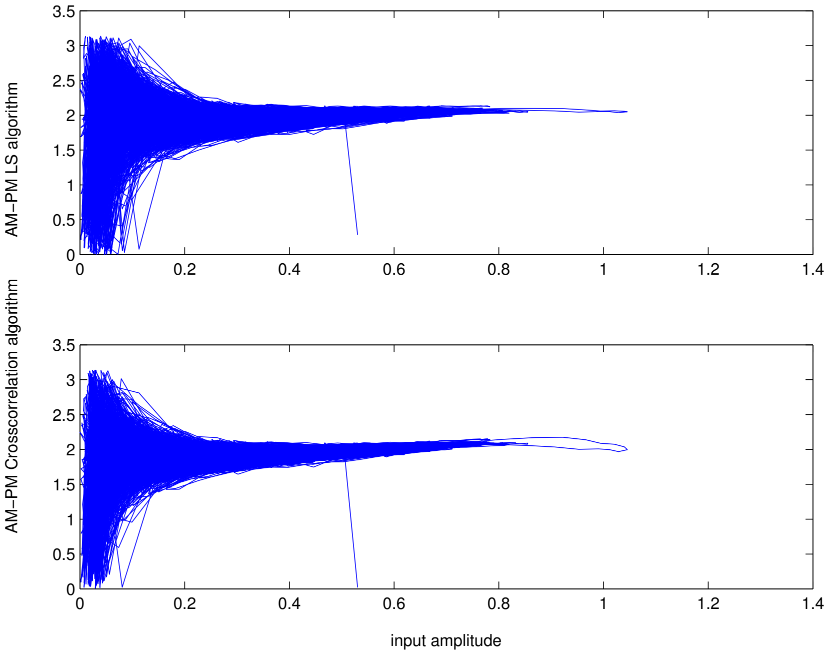


Figure E.4: AM-PM for the PA models based on the LS- and Crosscorrelation predistorters

Appendix F

Spectra within a back-end

In this paragraph, we present spectra at the different points in the back-end of a Radio Astronomy receiver. This appendix is based on [14], section 12.2. We use the block diagram as it was presented in figure 7.4. For convenience it is repeated in this appendix in figure F.1.

Our basic interest concerns the spectra (Power Spectral Densities) at the input of the Square Law Detector (SLD), at the output of the SLD and at the output of the Lowpass Filter. We assume that the signal at the input of the SLD is a Gaussian noise signal with a flat Power Spectral Density and that the spectrum ranges between $-\Delta\nu$ and $\Delta\nu$. $\Delta\nu$ is generally called the predetection bandwidth and in practice, it will be determined by the bandwidth of the last stage before the SLD. For convenience, the signal at the input of the SLD is supposed to be centered around DC (0 Hz). There is however no fundamental difference with systems where power is detected at IF frequencies. The total power of the signal equals σ^2. The different spectra within a receiver have also been discussed in [99], pp. 337 to 340. We present and discuss the results of this analysis.

The Power Spectral Density (spectrum) at the input of the SLD is presented in figure F.2.

The total power of the signal equals the area between the x-axis and the Power Spectral Density curve.

The most important characteristic of the SLD is that the instantaneous value of the output is proportional to the instantaneous power of the input. One operation which satisfies this characteristic is the 'square' operation. In this case, the input signal is multiplied with itself. The resulting spectrum at the output of the SLD is then indicated in figure F.3 (from [99], p. 338).

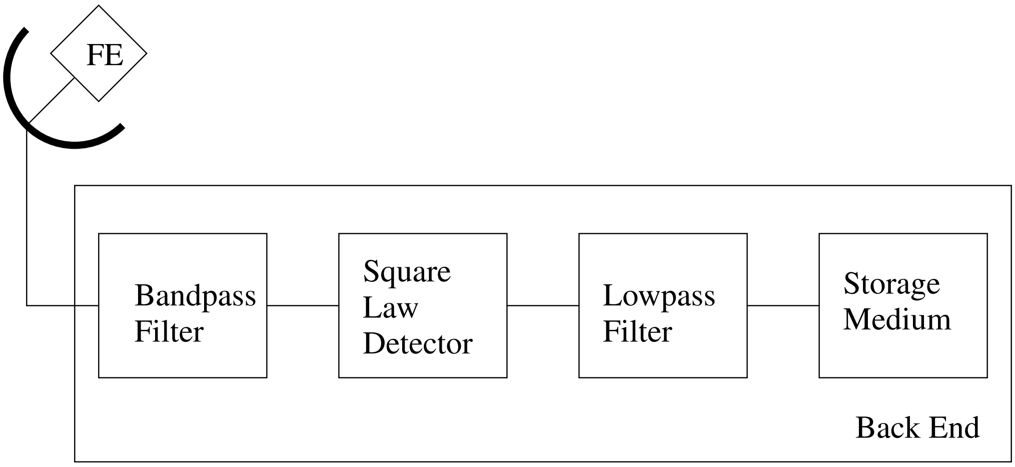


Figure F.1: Block diagram of a back-end for a single-dish telescope

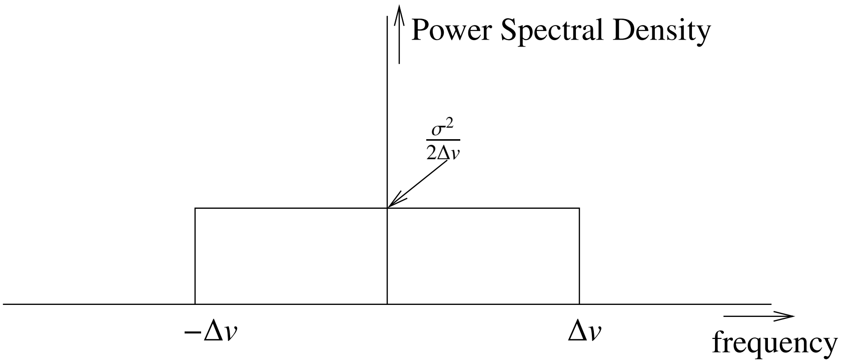


Figure F.2: Spectrum at the input of the Square Law Detector

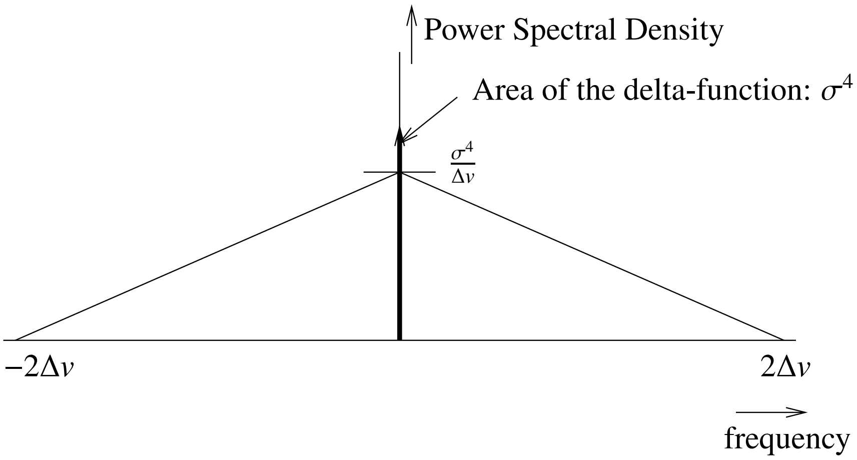


Figure F.3: Spectrum at the output of the Square Law Detector

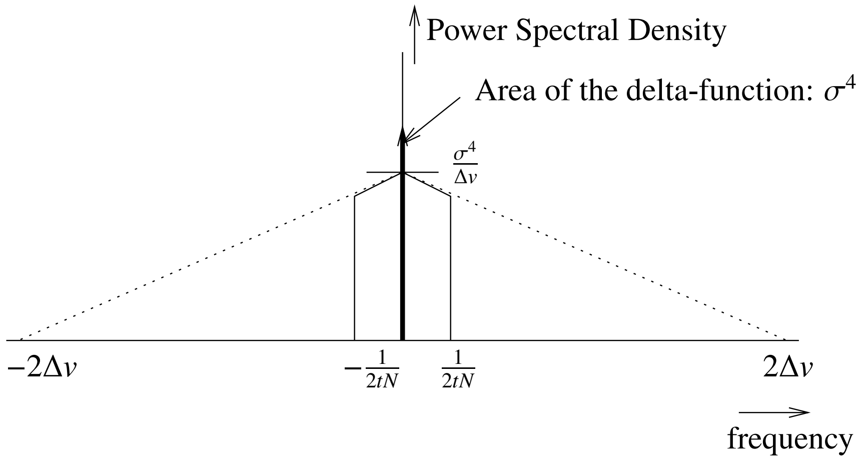


Figure F.4: Spectrum at the output of the Lowpass Filter

We see that the spectrum has broadened from $-\Delta v ... \Delta v$ to $-2\Delta v ... 2\Delta v$. Furthermore, the Power Spectral Density is decreasing for frequencies further from DC. The overall AC power at the output of the SLD equals the area between the x-axis and the Power Spectral Density and equals $2\sigma^4$. Note that the RMS value equals $\sqrt{2}\sigma^2$. Furthermore, there is a DC component, indicated by a delta function at 0 Hz, with an area equal to σ^4. This means that the power of the DC component equals σ^4 and that its RMS value equals σ^2. This signal is filtered. The purpose of the filtering is to reduce the area between the x-axis and the Power Spectral Density curve while passing through the DC value. This can be achieved by a Low Pass Filter. We assume that the Low Pass Filter is implemented in the digital domain and averages $N \cdot t$ samples. The effect of this 'filter' is approximately that the Power Spectral Density is restricted to the frequencies ranging from $-\frac{1}{2tN}$ to $\frac{1}{2tN}$. This is indicated in figure F.4.

We see that the output consists of a DC component, representing the power of the original signal and a residual AC component. The power of the DC component equals σ^4 and the power of the AC component is approximated by $\frac{\sigma^4}{\Delta v t N}$.

If the power at the input of the SLD is constant, two effects are clearly visible. First, if the number of samples tN, used to average the output of the SLD, is increased, the variation on the signal at the output of the filter is decreased. This is caused by the fact that the filter becomes more narrow (see figure F.4). Another effect is that if the predetection bandwidth Δv is increased, the variation on the signal at the output of the filter is decreased. This is due to the fact the the spectrum at the output of the SLD is widened as well, but the total power is still the same. The maximum value, at DC, is therefore lower. If the bandwidth of the Lowpass Filter is kept constant, the variation at the output of the Lowpass Filter is decreased.

Appendix G

Application of the Efficiency η^{tpd}

The total power receiver efficiency (η^{tpd}) equals the ratio of the SNR in case of digital addition in a TAA and in case of analog addition, see expression 9.12 in section 9.3. Writing this expression in a different form gives:

$$SNR^{d,tpd} = \eta^{tpd} SNR^{a,tpd} \tag{G.1}$$

The SNR in case of analog addition, $SNR^{a,tpd}$, is determined in section 9.3, for a hypothetical situation where all antennas have equal receiver noise power and the step in correlated power is known. In practice however, receiver noise power differs for different telescopes. In this paragraph, we will determine the quality of an analog adding system with unequal receiver noise powers for individual antenna's and use expression G.1 and Table 9.1 to estimate the quality of a digital adding system.
The factor η^{tpd} can also be used to estimate the effective antenna area in case of summation of digital signals.

G.1 Determination of $SNR^{a,tpd}$

Temperature, multiplied with two times Boltzmann's constant, is a measure of power. Because of this simple linear relation, temperature is very often used as a measure of power. In case of an analog TAA, an alternative definition of $SNR^{a,tpd}$ is:

$$SNR^{a,tpd} = \frac{T_{\Sigma,on} - T_{\Sigma,off}}{\Delta T_{rms}} \tag{G.2}$$

where $T_{\Sigma,on}$ equals the on-source temperature and $T_{\Sigma,off}$ equals the off-source temperature. ΔT_{rms} is the standard deviation given in a similar way as the source flux standard deviation is given (see expressions 7.4 and 7.7):

$$\Delta T_{rms} = T_{\Sigma,on} \frac{K_s}{\sqrt{\Delta\nu\, t\, N}} \tag{G.3}$$

As mentioned in section 9.1, the ADCs are equipped with Automatic Gain Control circuits which weigh the analog inputs to have reference levels for A/D conversion which are at a fixed fraction of the RMS level of the input signal. The weight coefficients w_k can be determined as:

$$w_k = \frac{T_L}{T_{R,k} + T_{A,k}} \tag{G.4}$$

where T_L is a measure of the power level at the outputs of the AGC systems (which is equal for all AGCs), $T_{R,k}$ is the receiver temperature and $T_{A,k}$ is the source antenna temperature of the k-th radiotelescope. In most applications $T_{A,k} << T_{R,k}$, so

$$w_k \approx \frac{T_L}{T_{R,k}} \tag{G.5}$$

The effective receiver temperature for the analog TAA as a single dish $T^a_{R,\Sigma}$ equals:

$$T^a_{R,\Sigma} = \sum_{k=1}^{M} w_k T_{R,k} \tag{G.6}$$

The effective antenna temperature of the analog TAA as a single dish $T^a_{A,\Sigma}$ equals:

$$T^a_{A,\Sigma} = \left(\sum_{k=1}^{M} \sqrt{w_k T_{A,k}} \right)^2 \tag{G.7}$$

The sumsignal $T_{\Sigma,on}$ at the output of the total power receiver, when M radiotelescopes are pointed on-source, is given by:

$$T_{\Sigma,on} = T^a_{R,\Sigma} + T^a_{A,\Sigma} = \sum_{k=1}^{M} w_k T_{R,k} + \left(\sum_{k=1}^{M} \sqrt{w_k T_{A,k}} \right)^2 \tag{G.8}$$

The sum signal $T_{\Sigma,off}$ in the absence of a source signal equals:

$$T_{\Sigma,off} = \sum_{k=1}^{M} w_k T_{R,k} \tag{G.9}$$

Using the values for $T_{\Sigma,on}$ (G.8), $T_{\Sigma,off}$ (G.9) and ΔT_{rms} (G.3) in expression G.2 and realizing that $T_{A,k} << T_{R,k}$, the following expression for the SNR in case of an analog TAA can be given:

$$SNR^{a,tpd} \approx \frac{\left(\sum_{k=1}^{M} \sqrt{w_k T_{A,k}}\right)^2}{\left(\sum_{k=1}^{M} w_k T_{R,k}\right)} \frac{\sqrt{\Delta\nu\, t\, N}}{K_s} \tag{G.10}$$

$$\approx \frac{\left(\sum_{k=1}^{M} \sqrt{w_k T_{A,k}}\right)^2}{MT_L} \frac{\sqrt{\Delta\nu\, t\, N}}{K_s} \tag{G.11}$$

So, using expression G.1, the digital SNR ($SNR^{d,tpd}$) becomes:

$$SNR^{d,tpd} \approx \eta^{tpd} \frac{\left(\sum_{k=1}^{M} \sqrt{w_k T_{A,k}}\right)^2}{MT_L} \frac{\sqrt{\Delta\nu\, t\, N}}{K_s} \tag{G.12}$$

We can define the effective antenna temperature for the tied array as a single dish, as $T_{A,\Sigma}^{d,tpd}$ and the effective receiver temperature as $T_{R,\Sigma}^{d}$ such that

$$SNR^{d,tpd} = \frac{T_{A,\Sigma}^{d,tpd}}{T_{R,\Sigma}^{d}} \frac{\sqrt{\Delta\nu\, t\, N}}{K_s} \tag{G.13}$$

To define the effective antenna temperature $T_{A,\Sigma}^{d,tpd}$ meaningfully, we will first define the effective receiver temperature $T_{R,\Sigma}^{d}$ and then equate the right parts of expressions G.12 and G.13 in order to define $T_{A,\Sigma}^{d,tpd}$.

In [48] it is suggested to define the effective receiver temperature of the tied array as described below. Because of the AGC, the power (temperature) at the output of the TAA always equals MT_L: for every specific set of values $T_{R,k}$, the weights w_k ($k = 1, ..., M$) are set in such way that the output equals MT_L. Suppose we freeze the settings of the AGCs by freezing w_k and suppose the receiver temperatures $T_{R,k}$ become 1 K for all receivers. The received power then equals $\sum_{k=1}^{M} w_k$. The effective receiver temperature of the TAA is then defined as follows:

$$T^{d}_{R,\Sigma} = \frac{MT_L}{\sum_{k=1}^{M} w_k} \tag{G.14}$$

This value can be interpreted as the real receiver temperature at the output of the adder normalized with the temperature in case all antennas have a receiver temperature of 1 K. Equating the right parts of G.12 and G.13 and using definition G.14 gives:

$$\eta^{tpd} \frac{\left(\sum_{k=1}^{M} \sqrt{w_k T_{A,k}} \right)^2}{MT_L} \frac{\sqrt{\Delta\nu\, t\, N}}{K_s} = \frac{T^{d,tpd}_{A,\Sigma}}{MT_L} \sum_{k=1}^{M} w_k \frac{\sqrt{\Delta\nu\, t\, N}}{K_s} \tag{G.15}$$

Then the effective antenna temperature $T^{d,tpd}_{A,\Sigma}$ for the tied array becomes:

$$T^{d,tpd}_{A,\Sigma} = \eta^{tpd} \frac{\left(\sum_{k=1}^{M} \sqrt{w_k T_{A,k}} \right)^2}{\sum_{k=1}^{M} w_k} \tag{G.16}$$

This formula can then be used to calculate the effective area A^{tpd}_{eff} of the Tied Array. Referring to expression 7.2, we define A^{tpd}_{eff} via:

$$S_0 = \frac{2kT^{d,tpd}_{A,\Sigma}}{A^{tpd}_{eff,\Sigma}} \tag{G.17}$$

When S_0 and $T^{d,tpd}_{A,\Sigma}$ are known, we can calculate the effective area of the Tied Array:

$$A^{tpd}_{eff,\Sigma} = \frac{2kT^{d,tpd}_{A,\Sigma}}{S_0} \tag{G.18}$$

where $A^{tpd}_{eff,\Sigma}$ is the effective area of the Tied Array, k is Boltzmann's constant, S_0 is the source flux in Watt per Hz per m^2 or Jansky (Jy).

The above mentioned formulas are used in the following example:

Example

The Westerbork Synthesis Radio Telescope works in a Tied Array mode using 10 antennas and 3-level quantization. The system temperatures are taken after a calibration observation: $T_{R,1} = 170, T_{R,2} = 151, T_{R,3} = 142, T_{R,4} = 164, T_{R,5} = 159, T_{R,6} = 177, T_{R,7} = 204, T_{R,8} = 130, T_{R,9} = 192, T_{R,10} = 157$. The source flux $F = 15$ Jansky (Watt Hz^{-1} m^{-2}) and the expected antenna temperature is $T_{A,k} = 1.5$ for all k. The effective area of a single telescope equals 276 m^2. From Table 9.1 we see that $\eta^{tpd} = 0.752$ for 3-level quantization and 10 antennas. Using expressions G.5 and G.14, we can calculate the effective receiver temperature $T^d_{R,\Sigma}$. Using G.5 and G.16, the effective antenna temperature $T^{d,tpd}_{A,\Sigma}$ can be calculated. The effective area of the Tied Array can then be calculated using expression G.18. The effective number of telescopes M_{eff} is calculated by dividing the effective area of the TAA with the effective area of a single telescope. The results of the calculations are:

$$\begin{aligned}
T^d_{R,\Sigma} &= 163.45\ ^\circ\text{K} \\
T^d_{A,\Sigma} &= 11.24\ ^\circ\text{K} \\
A^{tpd}_{eff,\Sigma} &= 2076\text{m}^2 \\
M_{eff} &= A^{tpd}_{eff,\Sigma}/276 = 7.522
\end{aligned}$$

Thus, because of the quantization, using 10 telescopes does not lead to an increase in effective area of 10 times the effective area of a single telescope but leads to an increase of approximately 7.5 times the effective area of a single telescope. This decrease in effective area is due to the effects of quantization.

Appendix H

Application of the Degradation η^{int}

The Degradation Factor η^{int} in case of a TAA, used in a correlation interferometer, is determined in expression 9.52 and rewritten in expression 9.53, section 9.4. Similar to the application of the total power receiver Efficiency in appendix G, the Degradation Factor is used to calculate $SNR^{d,int}$ (ρ_d) in case $SNR^{a,int}$ (ρ_a) is known. Also the effective area of the Tied Array can be calculated when used in VLBI mode.

H.1 Determination of $SNR^{d,int}$

Expression 9.52 can be rewritten as:

$$SNR^{d,int} = \eta^{int} SNR^{a,int} \tag{H.1}$$

Furthermore, the analog signal to noise ratio $SNR^{a,int}$ can be written as (referring to figure 9.5):

$$\begin{aligned} SNR^{a,int} &\approx \frac{\mathrm{E}(Z^a_\Sigma \cdot Z_p)}{\sqrt{\mathrm{E}\Big((Z^a_\Sigma \cdot Z_p)^2\Big)}} \frac{\sqrt{\Delta\nu\, t\, N}}{K_s} \\ &\approx k_{deg} \frac{\mathrm{E}(Y^a_\Sigma \cdot Y_p)}{\sqrt{\mathrm{E}\Big((Y^a_\Sigma \cdot Y_p)^2\Big)}} \frac{\sqrt{\Delta\nu\, t\, N}}{K_s} \end{aligned} \tag{H.2}$$

where k_{deg} is the degradation caused by the quantizers in the correlator. These degradation factors are well known ([85] and [86]): for one-bit correlation, k_{deg} equals $2/\pi$, for two-bit k_{deg} equals 0.88. Both Y^a_Σ and Y_p consist of a common signal N_0 and uncorrelated noise indicated with N^a_Σ and N_p. All three components N_0, N^a_Σ and N_p have zero mean. The power of the correlated component N_0 is not equal for Y^a_Σ and Y_p. We therefore define the gains a^a_Σ and a_p, which determine the power of the correlated signal. Thus:

$$Y^a_\Sigma = a_\Sigma N_0 + N^a_\Sigma \tag{H.3}$$

$$Y_p = a_p N_0 + N_p \tag{H.4}$$

Using these definitions, expression H.2 can be evaluated. The numerator of the middle part of the last approximation can be written as:

$$\begin{aligned} \mathrm{E}(Y^a_\Sigma \cdot Y_p) &= \mathrm{E}(a_\Sigma N_0 \cdot a_p N_0) + \mathrm{E}(a_\Sigma N_0 \cdot N_p) + \mathrm{E}(N^a_\Sigma \cdot a_p N_0) + \mathrm{E}(N^a_\Sigma \cdot N_p) \\ &= \mathrm{E}(a_\Sigma N_0 \cdot a_p N_0) \\ &= a_\Sigma a_p \mathrm{E}(N_0^2) \end{aligned} \tag{H.5}$$

The expected value of the product of a noise signal (N^a_Σ and N_p) with the correlated component N_0 and the expected value of the product of the two noise signals equal zero because all these components are mutually independent. The value $a_\Sigma a_p \mathrm{E}(N_0^2)$ can be rewritten into $a_\Sigma a_p \sigma^2_{N_0} = a_\Sigma \sigma_{N_0} \cdot a_p \sigma_{N_0}$. It is more convenient to use the temperature language so we define the effective antenna temperatures $T^a_{A,\Sigma}$ and T_{A,Y_p}:

$$T^a_{A,\Sigma} = a^2_\Sigma \sigma^2_{N_0} \tag{H.6}$$

$$T_{A,Y_p} = a^2_p \sigma^2_{N_0} \tag{H.7}$$

$$\tag{H.8}$$

The resulting expression of the numerator of the middle part of expression H.2 becomes:

$$\mathrm{E}(Y^a_\Sigma \cdot Y_p) = \sqrt{T^a_{A,\Sigma}}\sqrt{T_{A,Y_p}} \tag{H.9}$$

where we know from appendix G:

$$T^a_{A,\Sigma} = \left(\sum_{k=1}^{M} \sqrt{w_k T_{A,k}}\right)^2 \tag{G.7}$$

To approximate the denominator of the middle part of expression H.2, we assume that the power of the correlated component N_0 can be ignored compared to the noise power. This means that a_Σ and a_p equal zero. The denominator of the middle part then becomes:

$$\begin{aligned}\sqrt{\mathrm{E}\Big((Y^a_\Sigma \cdot Y_p)^2\Big)} &\approx \sqrt{\mathrm{E}\Big((N^a_\Sigma \cdot N_p)^2\Big)} \\ &\approx \sqrt{\mathrm{E}\Big((N^a_\Sigma)^2\Big) \cdot \mathrm{E}\Big((N_p)^2\Big)}\end{aligned} \tag{H.10}$$

The last step is allowed because N^a_Σ and N_p are uncorrelated. The expected value $\mathrm{E}((N^a_\Sigma)^2)$ can be written as $T^a_{R,\Sigma}$ and $\mathrm{E}((N_p)^2)$ as T_{R,Y_p}. Then, the denominator becomes:

$$\sqrt{\mathrm{E}\Big((Y^a_\Sigma \cdot Y_p)^2\Big)} = \sqrt{T^a_{R,\Sigma}}\sqrt{T_{R,Y_p}} \tag{H.11}$$

where we know from appendix G:

$$T^a_{R,\Sigma} = \sum_{k=1}^{M} w_k T_{R,k} \tag{G.6}$$

The SNR in case of an Analog Tied Array Adder (expression H.2) can then be written as:

$$\begin{aligned}SNR^a &\approx k_{deg}\frac{\sqrt{T^a_{A,\Sigma}}}{\sqrt{T^a_{R,\Sigma}}}\frac{\sqrt{T_{A,Y_p}}}{\sqrt{T_{R,Y_p}}}\frac{\sqrt{\Delta\nu\ t\ N}}{K_s} \\ &\approx k_{deg}\frac{\sum_{k=1}^{M}\sqrt{w_k T_{A,k}}}{\sqrt{\sum_{k=1}^{M} w_k T_{R,k}}}\frac{\sqrt{T_{A,Y_p}}}{\sqrt{T_{R,Y_p}}}\frac{\sqrt{\Delta\nu\ t\ N}}{K_s} \\ &\approx k_{deg}\frac{\sum_{k=1}^{M}\sqrt{w_k T_{A,k}}}{\sqrt{MT_L}}\frac{\sqrt{T_{A,Y_p}}}{\sqrt{T_{R,Y_p}}}\frac{\sqrt{\Delta\nu\ t\ N}}{K_s}\end{aligned} \tag{H.12}$$

Using $SNR^{d,int} = \eta^{int} SNR^{a,int}$, expression H.12 becomes:

$$SNR^{d,int} = k_{deg}\frac{\eta^{int}\sum_{k=1}^{M}\sqrt{w_k T_{A,k}}}{\sqrt{MT_L}}\frac{\sqrt{T_{A,Y_p}}}{\sqrt{T_{R,Y_p}}}\frac{\sqrt{\Delta\nu\ t\ N}}{K_s} \tag{H.13}$$

Using this expression, we are now able to determine the SNR at the output of a digital correlator using a Digital Tied Array Adder. To calculate this SNR we need to know the power of the correlated signal at the individual antennas ($T_{A,k}$) the power of the correlated signal at the VLBI partner (T_{A,Y_p}), the power level at the outputs of the AGC (T_L), the weight factors w_k and the power of the noise added by the VLBI partner (T_{R,Y_p}).

A general expression for the SNR of a digital correlator is:

$$SNR^{d,int} = k_{deg}\sqrt{SNR_1}\sqrt{SNR_2}\frac{\sqrt{\Delta\nu\ t\ N}}{K_s} \tag{H.14}$$

If we take a closer look at expression H.13 we see a ratio $\sqrt{T_{A,Y_p}}/\sqrt{T_{R,Y_p}}$. This actually is the square root of the SNR of the signal Y_p from the VLBI partner. It actually equals $\sqrt{SNR_2}$ in expression H.14. The remaining fraction to the left in expression H.13 actually acts as $\sqrt{SNR_1}$ in expression H.14 and represents the square root of the SNR of the digital sumsignal Σ:

$$\sqrt{SNR_1} = \frac{\eta^{int}\sum_{k=1}^{M}\sqrt{w_k T_{A,k}}}{\sqrt{MT_L}} \tag{H.15}$$

We can define the effective receiver- and antenna temperatures of the Tied Array using SNR_1 similar to the way T^a_{R,Y_p} and T_{A,Y_p} are used to define SNR_2:

$$\sqrt{SNR_1} = \sqrt{\frac{T^{d,int}_{A,\Sigma}}{T^d_{R,\Sigma}}} = \frac{\eta^{int}\sum_{k=1}^{M}\sqrt{w_k T_{A,k}}}{\sqrt{MT_L}} \tag{H.16}$$

We use the same definition for the effective receiver temperature $T^d_{R,\Sigma}$ as in the total power receiver case. We therefore repeat (expression G.14):

$$T^d_{R,\Sigma} = \frac{MT_L}{\sum_{k=1}^{M} w_k} \tag{G.14}$$

Then the effective antenna temperature of the array in case of correlation ($T^{d,int}_{A,\Sigma}$) can be determined using expressions H.16 and G.14:

$$T^{d,int}_{A,\Sigma} = (\eta^{int})^2\frac{\left(\sum_{k=1}^{M}\sqrt{w_k T_{A,k}}\right)^2}{\sum_{k=1}^{M} w_k} \tag{H.17}$$

This expression very much resembles expression G.16 which gives the effective antenna temperature in case of a total power receiver where η^{tpd} is replaced by $(\eta^{int})^2$.

The expression can be used to calculate the effective area in case a digital Tied Array Adder is connected to a correlation interferometer which is illustrated in the following example.

Example

If the configuration of the example in appendix G is used in the configuration where the output of the Digital Tied Array Adder is connected to a correlation interferometer, the effective area can simply be found by means of the effective area as calculated in the total power receiver mode.

$$T_{A,\Sigma}^{d,int} = \frac{(\eta^{int})^2}{\eta^{tpd}} T_{A,\Sigma}^{d,tpd} \tag{H.18}$$

$$A_{eff,\Sigma}^{int} = \frac{2T_{A,\Sigma}^{d,int} k}{S_0} \tag{H.19}$$

where k is Boltzmann's constant and F is the source flux in Jansky.

In the example in appendix G, the signals from 10 antenna's are summed digitally and η^{tpd} was found to be 0.752. For one-bit correlation, η^{int} equals 0.891 and for two-bit correlation, η^{int} equals 0.9. Using a calculated effective area in the total power receiver mode of 2076, the effective areas for one- and two-bit correlation equal 2016 m^2 and 2057 m^2 respectively. The effective number of telescopes equals 7.304 for 1-bit correlation and 7.453 for 2-bit correlation (7.522 in TPD mode).

Bibliography

[1] G.E. Moore, “Cramming more components onto integrated circuits,” *Electronics*, vol. 38, no. 8, Apr. 1965, pp. 114-117.

[2] W Stallings, *Computer Organization & Architecture*, sixth edition, Prentice Hall, 2003, ISBN 0-13-049307-4.

[3] R.H. Walden, “Performance Trends for Analog-to-Digital Converters,” *IEEE Commun. Mag.*, Feb. 1999, pp. 96-101.

[4] B. Murmann and B. Boser, “Digitally Assisted Analog Integrated Circuits,” *ACM queue DSP*, vol. 2, no. 1, Mar. 2004, pp. 65-71.

[5] R. van de Plassche, *Integrated Analog-to-Digital and Digital-to-Analog Converters*, Kluwer Academic Publishers, 1994, ISBN 0-7923-9436-4.

[6] J.E. Gunn, K.S. Barron, W. Ruczczyk, “A Low-Power DSP Core-Based Software Radio Architecture,” *IEEE J. Select. Areas Commun.*, vol. 17, no. 4, Apr. 1999, pp. 574-590.

[7] D.B. Chester, “Digital IF Filter Technology for 3G Systems: An Introduction,” *IEEE Commun. Mag.*, Feb. 1999, pp. 102-107.

[8] H. Nyquist, “Certain topics in telegraph transmission theory,” *Trans. AIEE*, vol. 47, Feb. 1928, pp. 617-644.

[9] C.E. Shannon, “Communication in the presence of noise,” *Proc. Institute of Radio Engineers*, vol. 37, no. 1, Jan. 1949, pp. 10-21.

[10] 3rd Generation Partnership Project; Technical Specification Group GSM/EDGE Radio Access Network; Radio transmission and reception, *3GPP TS 45.005, V6.3.0 (2003-08).*

[11] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Base Station (BS) radio transmission and reception (FDD), *3GPP TS 25.104, V6.3.0 (2003-09).*

[12] IEEE Std 802.11b-1999 (Supplement to ANSI/IEEE Std802.11, 1999 Edition), *Part 11: Wireless LAN Medium Access Control (MAC) and*

Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band.

[13] IEEE Std 802.11g-2003 (Amendment to IEEE Std802.11, 1999 Edition), *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band*.

[14] W.B. Davenport and W.L. Root, *An introduction to the theory of random signals and noise*, IEEE PRESS edition of a book published by McGraw-Hill Book Company in 1958 under the same title, 1987, ISBN 0-87942-235-1.

[15] J.H. van Vleck and David Middleton, "The Spectrum of Clipped Noise," *Proc. IEEE*, vol. 54, no.1, Jan. 1966, pp 2-19.

[16] E.Y. Chou and B. Sheu, "Nanometer Mixed-Signal System-on-a-chip Design," *IEEE Circuits Devices Mag.*, July 2002, pp. 7-17.

[17] W. Wolf, "A Decade of Hardware/Software Codesign," *Computer*, vol. 36, no. 4, Apr. 2003, pp. 38-43.

[18] R. Dömer, D. Gajski, J. Zhu, "Specification and design of Embedded Systems," *it+ti Magazine*, no. 3, Jun. 1998.

[19] D.D. Gajski, F. Vahid, S. Narayan, and J.Gong, "SpecSyn: An Environment Supporting the Specify-Explore-Refine Paradigm for Hardware/Software System Design," *IEEE Trans. VLSI Syst.*, vol. 6, no. 1, Mar. 1998, pp. 84-100.

[20] F.G. Wolff, M.J. Knieser, D.J. Weyer, C.A. Papachristou, "Using Codesign Techniques to support Analog Functionality," *Proc. CODES 1999*, 1999, pp. 79-83.

[21] P. Oehler, C. Grimm, and K. Waldschmidt, "A methodology for System-Level Synthesis of Mixed-Signal Applications," *IEEE Trans. VLSI Syst.*, vol. 10, no. 6, Dec. 2002, pp. 935-942.

[22] Ch. Grimm, Ch. Meise, W. Heupke, K. Waldschmidt, "Refinement of Mixed-Signal Systems with SystemC," *Design and Test in Europe 2003 (DATE 2003)*, Paris, France, 2003, pp. 11170-11171.

[23] A. Vachoux, C. Grimm, K. Einwich, "Towards Analog and Mixed-Signal Soc Design with SystemC-AMS," *Proc. Second IEEE International Workshop on Electronic Design, Test and Applications (DELTA 2004)*, Perth, Australia, Jan. 2004, pp. 97-102.

[24] T.A.C.M. Claasen, "System on a chip: Changing IC Design Today and in the Future," *IEEE micro*, May-June 2003, pp. 20-26.

[25] B. Kienhuis, E. Deprettere, K. Vissers, P. van der Wolf, "An Approach for Quantitative Analysis of Application-Specific Dataflow Architectures," *Proceedings of ASAP '97*, 1997, pp. 338-349.

[26] P. Lieverse, P. van der Wolf, E. Deprettere, K. Vissers, "A Methodology for Architecture Exploration of Heterogeneous Signal Processing Systems," *Journal of VLSI Signal Processing*, vol. 29, 2001, pp. 197-206.

[27] L. Ding, G.T. Zhou, Z. Ma, D.R. Morgan, J.S. Kenney, J. Kim and C.R. Giardina, "A Robust Digital Baseband Predistorter Constructed Using Memory Polynomials," *IEEE Trans. Commun.*, vol. 52, no. 1, Jan. 2004, pp. 159-165.

[28] H. Besbes, T. Le-Ngoc and H. Lin, "A Fast Adaptive Polynomial Predistorter for Power Amplifiers," Globecom 2001, vol. 1, 25-29 november 2001, pp. 659-663.

[29] World Wide Web: `www.itu.int/home`

[30] N. Pothcary, *Feedforward linear power amplifiers*, Artech House, 1999, ISBN 1580530222.

[31] World Wide Web: `www.naic.edu`

[32] World Wide Web: `www.mpifr-bonn.mpg.de/div /effelsberg`

[33] World Wide Web: `www.parkes.atnf.csiro.au`

[34] World Wide Web: `www.gb.nrao.edu/GBT`

[35] World Wide Web: `www.astron.nl`

[36] World Wide Web: `www.vla.nrao.edu`

[37] P.J. Napier, A.R. Thompson and R.D. Ekers, "The Very Large Array - Design and performance of a modern synthesis radio telescope," *Proc. IEEE*, vol. 71, Nov. 1983, pp. 1295-1320.

[38] World Wide Web: `www.narrabri.atnf.csiro.au`

[39] World Wide Web: `www.gmrt.ncra.tifr.res.in`

[40] A.R. Thompson, J.M. Moran and G.W. Swenson, *Interferometry and Synthesis in Radio Astronomy*, John Wiley and Sons, 1986, ISBN 0471806145.

[41] World Wide Web: `www.evlbi.org`

[42] World Wide Web: `www.jb.man.ac.uk/merlin`

[43] World Wide Web: `www.aoc.nrao.edu/vlba`

[44] M. Kouwenhoven, *Pulsar observations with the Westerbork Synthesis Radio Telescope*, PhD. Thesis, Utrecht, 2000, ISBN 90-393-2503-0.

[45] J.A. Hogbom and W.N. Brouw, “The Synthesis Radio Telescope at Westerbork. Principles of Operation, Performance and Data Reduction,” *Astron. Astrophys.*, vol. 33, 1974, pp. 289-301.

[46] J.W.M. Baars and B.G. Hooghoudt, “The Synthesis Radio Telescope at Westerbork. General Lay-Out and Mechanical Aspects,” *Astron. Astrophys.*, vol. 31, 1974, pp. 323-331.

[47] A. Bos, *Design consideration for the DZB: A new backend for the WSRT*, NFRA Internal Technical Report 200.

[48] A. van Ardenne, *Verlies in gevoeligheid met Westerbork als “digitale” tied-array. Een eerste verkenning*, NFRA Technical Note 289, Feb. 1979.

[49] J. Vankka, M. Kosunen, I. Sanchis and K. Halonen, “A Multicarrier QAM Modulator,” *IEEE Trans. Circuits Syst. II*, vol. 47, no. 1, Jan. 2000, pp. 1-10.

[50] S. Müller, R. Bäuml, R. Fischer and J. Huber, “OFDM with Reduced Peak-to-Average Power Ratio by Multiple Signal Representation,” *Annals of Telecommunications*, vol. 52, no. 1-2, feb. 1997, pp. 58-67.

[51] World Wide Web: `www.etsi.org`

[52] World Wide Web: `www.fcc.gov`

[53] D.C. Cox, “Linear amplification by sampling techniques, a new application for delta coders,” *IEEE Trans. Comm.*, vol. 23, no. 8, Aug. 1975, pp. 793-798.

[54] A. Bateman, “The combined analogue locked loop universal modulator (CALLUM),” *Proc. 42nd IEEE Veh. Technol. Conf.*, Sept. 1992, pp. 759-763.

[55] C.G. Rey, “Predistorter linearizes CDMA power amplifiers,” *Microwaves RF*, Oct. 1998, pp. 114-123.

[56] F. Zavosh, M. Thomas, C. Thron, T. Hall, D. Artusi, D. Anderson, D. Ngo, and D. Runton, “Digital predistortion techniques for RF power amplifiers with CDMA applications,” *Microwave Journal*, Oct. 1999.

[57] Y. Nagata, “Linear amplification technique for digital mobile communications,” *Proc. 39th IEEE Veh. Technol. Conf.*, May 1989, pp. 159-164.

[58] J. K. Cavers, “Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements,” *IEEE Trans. Veh. Technol.*, vol. 39, no. 4, Nov. 1990, pp. 374-383.

[59] M. Faulkner, T. Mattsson and W. Yates, “Adaptive linearisation using predistortion,” *Proc. 40th IEEE Veh. Technol. Conf.*, May 1990, pp. 35-40.

[60] A. S. Wright and W. G. Durtler, “Experimental performance of an adaptive digital linearized power amplifier,” *IEEE Trans. Veh. Technol.*, vol. 41, no.4, Dec. 1992, pp. 395-400.

[61] J. Kim and K. Konstantinou, “Digital predistortion of wideband signals based on power amplifier model with memory,” *Electronics Letters*, vol. 37, no. 23, Nov. 2001, pp. 1417-1418.

[62] C. Eun and E.J. Powers, “A predistorter design for a memory-less nonlinearity preceded by a dynamic linear system,” *Proc. GLOBECOM 1995*, 1995, pp. 152-156.

[63] L. Ding, R. Raich, and G.T. Zhou, “A Hammerstein predistortion linearization design based on the indirect learning architecture,” *Proc. ICASSP 2002*, vol. 3, May 13-17, 2002, pp. III-2689 - III-2692.

[64] L. Ding, G.T. Zhou, D.R. Morgan, Z. Ma, J.S. Kenney, J. Kim, and C.R. Giardina, “A robust digital baseband predistorter constructed using memory polynomials,” *IEEE Trans. on Communications*, vol. 52, no. 1, Jan 2004, pp. 159-165.

[65] R. Raich, H. Qian, and G.T. Zhou, “Digital baseband predistortion of nonlinear power amplifiers using orthogonal polynomials,” *Proc. ICASSP2003*, vol. 6, Apr. 6-10, 2003, pp. VI-689 - VI-692.

[66] R. Raich, H. Qian, and G.T. Zhou, “Orthogonal Polynomials for Power Amplifier Modeling and Predistorter Design,” *IEEE Trans. Veh. Technol.*, vol. 53, no.5, Sept. 2004, pp. 1468-1479.

[67] H. Pretl, L. Maurer, W. Schelmbauer, R. Weigel, B. Adler and J. Fenk, “Linearity Considerations of W-CDMA Front-Ends for UMTS,” *Proc. 2000 IEEE MTT-S Microwave Symposium*, 2000, pp. 433-436.

[68] Stephen A. Maas, *Nonlinear Microwave Circuits*, Artech House Inc., 1988, ISBN 0-89006-251-X.

[69] V. John Mathews, Giovanni L. Sicuranza, *Polynomial Signal Processing*, John Wiley & Sons, Inc., 2000, ISBN 0-471-03414-2.

[70] Sergio Benedetto and Ezio Biglieri, *Principles of Digital Transmission With Wireless Applications*, Kluwer Academic/Plenum Publishers, 1999, ISBN 0-306-45753-9.

[71] Hugo D. Wasaff, *Adaptive Pre-Distortion for Nonlinear High Power Amplifiers in OFDM Systems*, PhD. Thesis. Barcelona 2004, ISBN 84-688-9269-6.

[72] M. Kendall and A. Stuart, *The advanced theory of statistics*, 4th ed., vol. 2, Charles Griffin & Company, 1977, ISBN 0-85264-242-3.

[73] William H. Press, Brian P. Flannery, Saul A. Teukolsky, William T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing, 2nd Edition*, Cambridge University Press, 1992, ISBN 0-521-43108-5.

[74] L. Sundström, M. Faulkner and M. Johansson, “Quantization analysis and design of a digital linearizer for RF power amplifiers,” *IEEE Trans. Veh. Technol.*, vol. 45, no. 4, Nov. 1996, pp. 707-719.

[75] R.A. Wannamaker, S.P. Lipshitz, J. Vanderkooy, and J.Nelson Wright, “A Theory of Nonsubtractive Dither,” *IEEE Trans. Signal Processing*, vol. 48, no. 2, Feb. 2000, pp. 499-516.

[76] World Wide Web: `dspvillage.ti.com`

[77] G.R. Kirchhoff, “ Über den zusammenhang zwischen Emission un Absorption van Licht und Wärme,” *Monatsber. Akad. Wiss. Berlin*, December, 1859, pp. 783-787.

[78] K.G. Jansky, “Electrical Disturbances Apparently of Extraterrestrial Origin,” *Proc. IRE*, vol. 21, 1933, pp. 1387-1398.

[79] S. Weinreb, “Digital Radiometer,” *Proc. IEEE*, vol. 49, no. 6, 1961, p. 1099.

[80] J.L. Casse, C.A. Muller, “The Synthesis Radio Telescope at Westerbork. The 21 cm Continuum Receiver System,” *Astron. & Astrophys.*, vol. 31, 1974, pp. 333-338.

[81] J.D. Kraus, *Radio Astronomy*, McGraw-Hill, 1966, ISBN 0070353921.

[82] J.M. Moran, in M. Felli and R.E. Spencer(eds.), *Very Long Baseline Interferometry Techniques and Applications*, Kluwer, Dordrecht, 1989, ISBN 0-7923-0376-8.

[83] J.M. Moran, in J.A. Zensus, P.J. Diamond, P.J. Napier(eds.), *Very Long Baseline Interferometry and VLBA*, Astronomical Society of the Pacific, 1995, ISBN 1886733023.

[84] R.J. Dewey, “The effects of correlated noise in phased-array observations of radio sources,” *The Astronomical Journal*, vol. 108, 1994, pp. 337.

[85] F. K. Bowers and R.J. Klingler, "Quantization noise of correlation spectrometers," *Astron. Astrophys. Suppl.*, Vol. 15, 1974, pp. 373-380.

[86] B.F.C. Cooper, "Correlators with Two-Bit Quantization," *Australian Journal of Physics*, Vol. 23, 1970, pp. 521-527.

[87] J. B. Hagen and D.T. Farley, "Digital-correlation techniques in radio science," *Radio Science*, Vol. 8, Numbers 8,9, 1973, pp. 775-784.

[88] A. Stuart and J.K. Ord, *Kendall's advanced theory of statistics*, 5th ed., vol. 1, Charles Griffin & Company, 1987, ISBN 0-85264-285-7.

[89] A. van Ardenne, *Calibrating the tied array in the case of analog summation of analog signals*, NFRA Technical Note 315, Februari 1980.

[90] Robert M. Gray and Thomas G. Stockham, "Dithered quantizers," *IEEE Trans. Inform. Theory*, vol. 39, no. 3, May 1993, pp. 805-812.

[91] B. Widrow, "Statistical analysis of amplitude-quantized sampled-data systems," *AIEE Trans. Apll. Industry*, vol. 79, pt. 2, Jan. 1961, pp. 555-568.

[92] J.A. McFadden, "The correlation function of a sine wave plus noise after extreme clippings," *IRE Trans. on Inform. Theory*, Jun. 1956, pp. 82-83.

[93] W.R. Bennett, "Spectra of quantized signals," *Bell Syst. Tech. J.*, vol. 27, Jul. 1948, pp. 446-472.

[94] Nelson M. Blachman, "The inter-modulation and distortion due to quantization of sinusoids," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 6, Dec. 1985, pp. 1417-1426.

[95] I.S. Gradshteyn and I.M. Ryzhik, *Table of integrals, series and products*, 4th ed., Academic Press, 1965.

[96] L. Cheded and P.A. Payne, "The exact impact of amplitude quantization on multi-dimensional, high-order moments estimation," *Signal Processing*, vol. 39, no. 3, Sep. 1994, pp. 293-315.

[97] L. Cheded and P.A. Payne, "A new and natural proof to both the arcsine law and a related result," *Int. J. Electronics*, vol. 76, no. 6, 1994, pp. 1099-1114.

[98] A. W. Gunst, *The use of multi-bit A/D converters in radio astronomy*, University of Twente, Enschede, Report Code 060.2721, Jun. 1999.

[99] R. N. Bracewell, *The Fourier Transform and its Applications*, 2nd edition, revised, McGraw Hill, 1986, ISBN 0-07-066454-4.

Publications

[1] S. Alliot, E.F. Deprettere, and A.B.J. Kokkeler. A modular approach for a large scalable embedded signal processing system. In *Proceedings of the Progress 2000 Workshop on Embedded Systems*, October 2000.

[2] A.J. Boonstra, A. Leshem, A.J. van der Veen, A.B.J. Kokkeler, and G. Schoonderbeek. The effect of blanking of tdma interference on radio-astronomical observations: experimental results. In *Proceedings of the ICASSP 2000 conference*, June 2000.

[3] A.W. Gunst, A.B.J. Kokkeler, and G.W. Kant. A/d converter research for ska. In *Proceedings NFRA SKA Symposium; Technologies for Large Antenna Arrays*, April 1999.

[4] A.B.J. Kokkeler. A digital crosscorrelation predistorter for power amplifiers. In *Proceedings of the Prorisc workshop on Circuits, Systems and Signal Processing*, November 2003.

[5] A.B.J. Kokkeler. Analysis of power amplifier modeling schemes for crosscorrelation predistorters. In *Proceedings of the Prorisc workshop on Circuits, Systems and Signal Processing*, November 2004.

[6] A.B.J. Kokkeler. A crosscorrelation predistorter using memory polynomials. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, May 2004.

[7] A.B.J. Kokkeler. Performance analysis of crosscorrelation predistorters. In *Proceedings of the European Conference on Wireless Technology*, October 2004.

[8] A.B.J. Kokkeler, P. Fridman, and A. van Ardenne. Degradation due to quantization noise in radio astronomy phased arrays. *Experimental Astronomy*, 11:33–56, 2001.

[9] A.B.J. Kokkeler and A.W. Gunst. Modeling correlation of quantized noise and periodic signals. *IEEE Signal Processing Letters*, 11(10), October 2004.

[10] A. Leshem, A.J. van der Veen, A.J. Boonstra, A.B.J. Kokkeler, and G. Schoonderbeek. Blanking of tdma interference and its effects on radio-astronomical correlation measurement: experimental results. In *Proceedings of the Prorisc workshop on Circuits, Systems and Signal Processing*, pages 24–25, November 1999.

[11] L.T. Smit, G.J.M. Smit, J.L Hurink, and A.B.J. Kokkeler. Soft output bit error rate estimation for wcdma. In *Proceedings of Personal Wireless Communication 2003 conference*, September 2003. ISBN: 3-540-20123-8.

[12] A.B. Smolders, G.W. Kant, A.B.J. Kokkeler, and A.W. Gunst. Receiver architectures of the thousand element array (thea). In *Proceedings of the SPIE 2000 conference*, pages 27–31, March 2000.