

TECHNISCHE UNIVERSITÄT
MÜNCHEN

**Sampling ADC Based Data Acquisition
for
Positron Emission Tomography**

Dissertation

von

Alexander Bernd Mann



TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Datenverarbeitung

Lehrstuhl für Experimentalphysik E18

Sampling ADC Based Data Acquisition
for
Positron Emission Tomography

Alexander Bernd Mann

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Th. Eibert

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. K. Diepold

2. Univ.-Prof. Dr. rer. nat. St. Paul

Die Dissertation wurde am 02.12.2009 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 13.04.2010 angenommen.

Abstract

This work addresses the requirements of a data acquisition system for the Munich Avalanche Diode Positron Emission Tomograph-II (MADPET-II), a PET scanner system for radiopharmaceutical studies with small animals. The unique detector design, with single channel readout of scintillator crystals using avalanche photodiodes, allows high count rates and improved image reconstruction performance. The data acquisition system is therefore continuously digitizing all 1152 detector channels and has to extract a precise energy and time information for each detected signal in real-time. The current front-end specification is thus used as a starting point to evaluate different feature extraction algorithms by simulation. The simulated results are then compared to real measurements with the MADPET-I and MADPET-II systems. Further measurements are targeting the data acquisition with novel silicon photomultiplier detectors, especially focusing on the challenging multimodal combination of PET and magnetic resonance imaging (MRI). In addition, the work addresses also the offline calibration of the scanner system, using the recorded data. The data acquisition system is realized with FPGA based electronic modules, already used in high energy physics, utilizing the possible synergies between both research fields. This leads to the development of a new mezzanine sampling analog-to-digital converter card which can be the basic building block of an updated readout system for MADPET-II.

Kurzfassung

Die vorliegende Arbeit beschäftigt sich mit den Anforderungen an ein Datenerfassungssystem für den Munich Avalanche Diode Positron Emission Tomograph-II (MADPET-II), einem PET Scanner zur Durchführung von radiopharmazeutischen Studien an kleinen Tieren. Dessen spezielles Detektordesign, basierend auf Einzelkanalauslese der Szintillatorkristalle durch Lawinenphotodioden, erlaubt hohe Zählraten sowie eine verbesserte Bildrekonstruktion. Das Datenerfassungssystem digitalisiert dazu kontinuierlich alle 1152 Detektorkanäle und muss danach in Echtzeit zu jedem detektierten Ereignis eine präzise Energie- und Zeitinformation bestimmen. Hierzu wird die derzeitige Detektorspezifikation als Ausgangspunkt verwendet, um verschiedene Algorithmen zur Merkmalsextraktion mittels Simulationen zu bewerten. Die Simulationsergebnisse werden danach mit echten Messdaten von MADPET-I und MADPET-II verglichen. Weitergehende Messungen betrachten die Datenerfassung mit Silizium-Photomultiplier Detektoren, insbesondere im Hinblick auf die anspruchsvolle, multimodale Kombination von PET und Magnetresonanztomographie (MRT). Weiterhin geht die Arbeit auch auf die Offline-Kalibrierung des Scannersystems mittels der aufgezeichneten Daten ein. Die Datenerfassung ist mit FPGA basierten Elektronikmodulen realisiert, die bereits in der Hochenergiephysik Verwendung finden, wodurch sich Synergieeffekte zwischen den beiden Fachgebieten nutzen lassen. Dies führt schließlich zu der Entwicklung einer neuen Mezzanine Sampling ADC Karte als Grundbaustein für ein aktualisiertes Auslesesystem des MADPET-II Scanners.

'Where shall I begin, please your Majesty?' he asked.

'Begin at the beginning,' the King said gravely, 'and go on till you come to the end: then stop.'

Lewis Carroll, Alice's Adventures in Wonderland

The beginning is the most important part of the work.

Plato, The Republic

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Overview	3
2	Introduction to Positron Emission Tomography	5
2.1	Positron Decay and Tracers	5
2.2	PET Detectors	8
2.2.1	Requirements	8
2.2.2	Photon Detection	9
2.2.3	Detector Systems	14
2.3	Image Reconstruction	16
2.4	Multimodality Imaging	19
3	The MADPET-II Small Animal PET System	21
3.1	Clinic versus Small Animal PET Imaging	21
3.2	The MADPET-II Detector Concept	22
3.2.1	APD Readout Electronics	24
3.2.2	Data Acquisition	25
3.3	Detector Calibration Procedures	26
3.3.1	Energy Calibration	27
3.3.2	Existing Time Calibration	28
3.3.3	Analytic Time Calibration	32
4	Data Acquisition Based on Sampling ADCs	45
4.1	Signal Sampling	46
4.1.1	Detector Signal Model	47
4.1.2	ADC Requirements	49
4.1.3	Effects due to Sampling	50
4.2	Trigger Decision	52
4.2.1	Signal Averaging	53

CONTENTS

4.3	Amplitude Detection	56
4.3.1	Pile-Up	57
4.3.2	Signal Conditioning	58
4.3.3	Amplitude Measurement	61
4.3.4	Sensitivity to Noise	76
4.3.5	Summary of the Amplitude Detection	81
4.4	Time Reconstruction	83
4.4.1	Leading Edge Timing	83
4.4.2	Constant Fraction Timing	84
4.4.3	Least Squares Timing	95
4.4.4	Summary of the Time Reconstruction	104
4.5	Pulse Shape Optimization	104
5	Sampling ADC Based Electronics	109
5.1	The COMPASS Sampling ADC System	109
5.1.1	The Sampling ADC Module (SADC)	110
5.1.2	The Multiplexer Module (MUX8)	112
5.1.3	The Clock Distribution	113
5.1.4	The PCI Interface/Buffer Module	114
5.1.5	SADC USB Interface	115
5.2	Introduction to Field Programmable Gate Arrays (FPGAs)	116
5.2.1	Internal Structure	116
5.2.2	Design Process	117
5.3	Firmware Development	118
5.3.1	SADC Firmware	118
5.3.2	MUX8 Firmware	127
5.4	ATCA Based Electronics	127
5.4.1	The Mezzanine Sampling ADC	128
5.4.2	MSADC USB Interface	132
6	Sampling ADC Measurements	133
6.1	MADPET-II Measurement Setup	133
6.2	Data Analysis	134
6.2.1	Amplitude Extraction	134
6.2.2	Time Extraction	136
6.3	MADPET-I Measurement Setup	141
6.3.1	MADPET-I Hardware Overview	141
6.3.2	Data Analysis	142

6.4	Silicon Photomultiplier Measurement Setup	144
6.4.1	Hardware Overview	144
6.4.2	Measurement Results	146
7	Summary and Outlook	151
Appendix		
A	List of Abbreviations	155
B	Mathematical Notation	159
C	Transformations Overview	161
C.1	Fourier Transform	161
C.2	Laplace Transform	162
C.3	z-Transform	163
	List of Figures	165
	List of Tables	171
	Bibliography	173
	Acknowledgments	181

Chapter 1

Introduction

Since the first medical X-ray image by Wilhelm Conrad Röntgen in 1895, the field of medical imaging has evolved into a powerful tool for everyday medical diagnosis. Besides the “classical” X-ray radiography, new modalities like X-ray computed tomography (CT), ultrasonography and magnetic resonance imaging (MRI) enable today a non-invasive and high resolution anatomic view into the body. With nuclear medicine imaging techniques, like single photon emission computed tomography (SPECT) and positron emission tomography (PET), even functional processes in organs and cells can be visualized in a non-invasive way. Altogether, these imaging techniques form the basis for the understanding and treatment of nowadays medical challenges in neurology, cardiology and oncology. Especially the diagnosis of cancer benefits from the sensitivity to biological processes that can be provided by PET scanner systems. Together with the anatomic information from CT or MRI systems, this allows a precise localization and quantification of cancerous tissue and thus supports treatment planning and monitoring.

The continuous improvement of scanner systems in terms of sensitivity and resolution is not only crucial for applications on human patients. An important fraction of the examinations with such systems is carried out in pre-clinical animal studies, which are still required to develop and evaluate new treatment methods. Especially the imaging of small animals, like mice or rats, puts high requirements on the imaging equipment due to their small size and limited blood volume. Thus, small animal imaging systems usually have to explore new detector concepts and image reconstruction methods to further extend the limits. Besides the resulting achievements in the treatment and drug development due to increased scanner performance, the new detector concepts at this technological forefront are also successively transferred into clinical scanner systems, which again allows a better and more precise treatment of patients.

1.1 Motivation

The desire to continuously push the limits of current detector technology in positron emission tomography is unfortunately facing the more and more tight requirements regarding development and manufacturing costs. Clinical systems can compensate this by using custom-made integrated circuits in sufficiently high quantities to reduce the overall price. However, early development and research setups for new detector concepts are only hardly compatible with the cost optimized data acquisition architecture of large volume clinical scanner systems. Usually, this necessitates the time-consuming development and testing of dedicated readout systems for each specific detector setup.

This work aims at the exploration of possible synergies between the field of positron emission tomography and high energy physics. Both areas are usually working at the forefront of detector technology, with however some difference in size: Whereas PET systems are including some 1000–10000 front-end channels, experiments in high energy physics today easily range up to millions of channels. Thus, a large effort is already invested by the high energy physics community in the development of high performance and cost efficient readout and data acquisition electronics. In fact, the principle of a typical scintillator based PET detector front-end is similar to a calorimeter detector, which is present in almost every high energy physics setup. The wide use of reconfigurable logic devices in nowadays electronics design, allows the modification and adaptation of already existing readout modules from high energy physics experiments to new tasks.

A first evaluation of possible applications for a free running sampling analog-to-digital converter system with the MADPET-II small animal PET scanner, based on LSO-APD detectors, was already started in [Man04]. This first exploration is now continued, with a thorough look at different feature extraction algorithms and data processing/interpolation approaches, also taking into consideration the hardware implementation effort. Simulations provide thereby a first performance comparison, which is then verified with real measurement data. Finally, all algorithm and hardware developments lead to a flexible and cost effective readout system, suitable for different PET scanner systems, including the growing field of multimodal PET/MRI imaging with *silicon photomultiplier* based detectors.

Free running sampling ADCs are already used in commercial PET systems [M⁺03, B⁺05], however, currently only for tasks like amplitude determination and offset compensation. The determination of the arrival time of the detector signal is usually realized with dedicated analog application specific integrated circuits (ASICs) and time-to-digital converters. Other small animal PET scanner systems, which are based on the *block detector* concept, also use sampling ADCs to acquire the full signal waveform, but move the pulse analysis and feature extraction tasks into a post processing computer [S⁺06, G⁺06b]. In addition, the pulse detection and trigger decision sometimes require additional detection logic and are not derived from the sampled ADC data [S⁺02]. The focus of this work is therefore on the integration of all data processing steps into the digital data acquisition hardware, including also a precise time extraction for each sampled signal pulse. Thus,

the required electronics effort can be reduced and the use of configurable logic circuits allows more flexibility compared to ASIC based solutions.

The LabPET™ small animal PET scanner [F⁺09] uses a comparable data acquisition approach with free running ADCs and subsequent processing of the sampled data in field programmable gate arrays (FPGAs) and digital signal processors (DSPs). First evaluations in [L⁺05, L⁺09, F⁺08] explored already some digital processing algorithms for a BGO-APD based detector assembly, however, without deeper analysis of pulse shape influences on the timing and amplitude extraction performance.

1.2 Overview

The interdisciplinary field of positron emission tomography combines the knowledge from many scientific areas, ranging from medicine over mathematics and physics to electric engineering and computer science. Thus, different scientific backgrounds have to be supported by shortly introducing the underlying principles of the applied methods.

A basic introduction into the physics and applications of positron emission tomography is given in *Chapter 2*. This covers the basic physics of positron emission and annihilation followed by the various detection methods for the emerging annihilation photons. Since different detector assemblies and acquisition concepts are used in current clinical and preclinical PET systems, a short overview is given and the basic nomenclature is introduced. Finally, the mathematical process of reconstructing an image from acquired data is outlined, with an outlook to the more and more important field of multimodality imaging.

Chapter 3 introduces the MADPET-II small animal PET system, which motivated this work. The unique detector concept of single channel readout provides superior count rate performance and reconstruction performance, but requires increased effort for the data acquisition. After the presentation of the detector front-end electronics, an overview of the existing analog ASIC based data acquisition system is given. This chapter includes also a presentation of the calibration routines, which are required to operate the system. The time calibration is therefore especially important to ensure a good signal to noise ratio in the reconstructed image. After the description of two existing time calibration procedures, an analytical approach, based on pseudo-inverse matrices, is presented. By utilizing the well determined matrix structure, this method allows a fast and memory efficient determination of the time calibration coefficients, without computing intense function minimization or matrix decompositions.

Chapter 4 targets a new sampling ADC based data acquisition concept for the MADPET-II system. By moving signal processing tasks from the analog to the digital domain, the data acquisition gains in flexibility and becomes more cost effective. Therefore, this chapter starts with an introduction to signal sampling and mathematical modeling of the specific detector pulse shapes from the existing preamplifier modules. This is then further extended to the particular tasks that have to be addressed by the digital processing logic,

1 INTRODUCTION

namely trigger decision, amplitude extraction and time extraction. To evaluate and compare the performance, the different algorithms are simulated with the previously derived signal model. As the data acquisition system is also aiming for a hardware based implementation, the description of the individual functional blocks focuses especially on an efficient implementation in programmable logic. Finally, an outlook for an optimized pulse shape in a future redesign of the preamplifier modules is given.

The realization of a large number of sampling ADC channels with subsequent data processing is described in *Chapter 5*. At first, the electronic modules for the COMPASS sampling ADC system are described, which were used for first measurements on MADPET-II. As this system is based on data processing in field programmable gate arrays, a short introduction into field programmable gate arrays and the design process for programmable logic is given. This is followed by a description of the specific firmware features for the MADPET-II readout system. At the end, a new sampling ADC design for the Advanced Telecom Computing Architecture standard is presented, which may provide the basis for the sampling ADC based MADPET-II data acquisition system.

Chapter 6 summarizes the measurement results that were obtained with the sampling ADC based electronics and describes the underlying data analysis methods. This covers the first evaluation measurements with a prototype system of the MADPET-II scanner and the setup of a small scale readout system for a lab course PET scanner. Finally, first measurements with the upcoming silicon photomultiplier based detectors are described, including the challenging combination of multimodal PET/MR measurements.

Finally, *Chapter 7* summarizes the achieved results and gives an outlook to further possible developments.

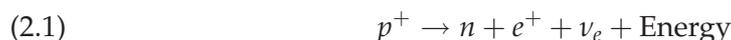
Chapter 2

Introduction to Positron Emission Tomography

Positron emission tomography (PET) is nowadays widely used as a diagnostic tool in nuclear medicine, especially for questions in oncology, neurology and cardiology. The basic principle relies on the detection of photons, arising from a positron-electron annihilation inside the patient's body. In contrast to other tomographic imaging techniques, like X-ray computed tomography (CT) and magnetic resonance imaging (MRI), PET requires the additional application of special positron emitting *tracers* to the patient, prior to the scan procedure. The actual PET scan is then a completely passive process which detects the emerging photons and finally generates a picture of the tracer activity distribution inside the patient's body. By a proper selection of the applied tracer and its interaction inside the body, the PET examination can be focused on different metabolism processes and also visualize the biological activity *in vivo*.

2.1 Positron Decay and Tracers

The underlying principle of positron emission tomography is the β^+ decay of radioactive atoms. During this process, a proton (p^+) in the atomic nucleus is transformed into a neutron (n), a positively charged antielectron or positron (e^+) and an electron neutrino (ν_e). By this conversion the atomic number of the decaying atom is reduced by one and since the neutron stays in the nucleus, the atomic mass number is unchanged.



The positron is emitted from the nucleus and is slowed down by atomic collisions within a few millimeters around the originating atom, depending on its energy and the surrounding material. Together with an electron the positron can form an atom-like com-

pound, called a positronium which has a lifetime of about 10^{-10} s. Since the positron is the antiparticle of the electron, both combine in an annihilation reaction which converts their masses into two annihilation photons with energies of 511 keV each. If the initial electron-positron pair is at rest, both photons are emitted in opposite directions (180° angle) because of momentum conservation. Due to small remaining momentum during the annihilation process the photons can deviate from the 180° angle by about $\pm 0.25^\circ$ [CSP03].

Figure 2.1 gives a schematic view of the decay and detection process in positron emission tomography. The error of non-collinearity is illustrated by the second photon on the right side, which deviates from the perfect 180° emission line and is registered by a nearby detector element. Together with the positron range, the non-collinearity influences therefore the minimum achievable spatial resolution of a PET system, since the following image reconstruction is always assuming a perfect collinear photon pair.

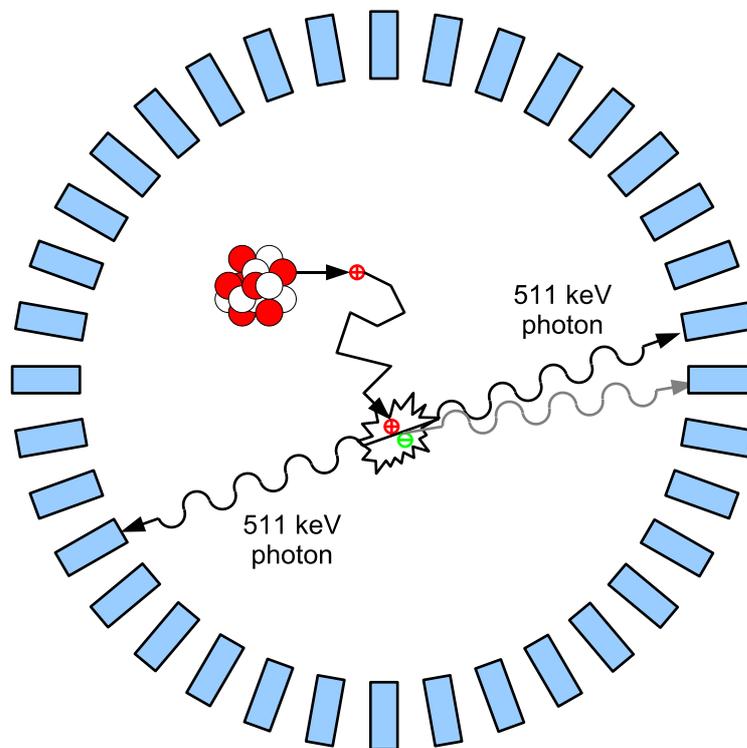


Figure 2.1: Schematic view of positron decay, positron range in matter, annihilation and tomographic photon detection. The second photon on the right side (*grey*) illustrates the effect of non-collinear annihilation. [CSP03]

Due to the energy of 511 keV of the two emitted photons, there are two possible interaction processes in matter which can influence the photon trajectories. The photoelectric effect can cause the absorption of one photon by the excitation and ejection of an orbital electron from an atom. The remaining vacancy in the shell—arising from the ejected electron—is then filled by an electron from a higher shell, which leads to the emission of characteristic X-rays or emission of Auger electrons from the outer shell. The second

possible interaction is Compton scattering. During this process the photon is scattered on an outer shell electron of an atom and transfers only a part of its energy to the electron. Compton scattering causes the photon to change its direction, according to the scattering angle, which also determines the energy loss of the photon.

Since the radioactive instability of the atomic nucleus does not interfere with the chemical behavior of the atom, instable atoms can be used for chemical synthesis just like non-radioactive isotopes. This allows to synthesize biologically active substances, like for example glucose, which are radioactively labeled. These tracers are then applied to the patient and participate in the metabolism processes inside the body. As cancer tissue for example shows a higher glucose metabolism it will also consume more of the radioactive glucose and therefore accumulate a higher concentration of the positron emitter than the surrounding tissue. As the PET imaging reconstructs the activity distribution, the malicious tissue can be identified due to the higher number of positron decays. Some widely used tracer radionuclides for PET are listed in Table 2.1.

Tracer isotope	Half-life	Application
^{11}C	20.3 min.	cardiology, oncology
^{13}N	10.0 min.	cardiology
^{15}O	2.07 min.	cardiology, neurology
^{18}F	110 min.	cardiology, neurology, oncology

Table 2.1: Tracers for PET and their applications.

Since the half-lives of the tracer isotopes are relatively short, most of them have to be produced just-in-time before the PET examination. Only ^{18}F can be stored for a reasonable time (some hours) and is thus easily available for measurements during the development of a PET system.

The production of the tracer nuclides is done by means of a cyclotron for particle acceleration. The cyclotron accelerates hydrogen ions ($^1_1\text{H}^{\pm}$), deuterons ($^2_1\text{H}^+$) or α particles ($^4_2\text{He}^{2+}$) to energies of about 8–15 MeV, which then are directed to a production target. Inside the target material the accelerated particles induce further nuclear reactions ((p, n) , (d, n) , (d, α) , (α, n)) which then produce the tracer isotope. After the irradiation of the target, the created tracers have to be separated from the target material and processed further to obtain the bioactive radiopharmaceutical. Due to the high operation and maintenance costs of the whole production process, only a limited number of sites can afford to produce and use the short living tracer nuclides. PET imaging without in-house tracer production is almost only possible with ^{18}F based tracers, due to the longer decay time.

Through radiochemical processing, the positron emitting isotopes have to be extracted from the target material and can then be used for the synthesis of radiopharmaceuticals. Depending on the PET application, different biological processes have to be addressed

by the tracers. If the main interest is in the glucose-metabolism, for example in oncology and neurology, ^{18}F -fluorodeoxyglucose (FDG) can be used as a radioactive glucose analog to monitor glucose uptake by tumors or certain areas of the brain. Due to the advantageous half-life, FDG is also the most widely used positron-emitting pharmaceutical nowadays. Similar uptake monitoring can also be done with fat and amino-acid based metabolisms using ^{11}C based tracers. The more short-lived tracer nuclides like ^{15}O and ^{13}N are mainly used for monitoring of blood flow properties and to distinguish between vital and necrotic tissue in cardiology and neurology.

2.2 PET Detectors

The basic principle of positron emission tomography is the coincident detection of photons, originating from positron-electron annihilations and the subsequent reconstruction of the distribution of the positron emitting tracer substance from this data. The fact that both photons, which are emitted to opposite sides, have to be detected usually results in a ring-like arrangement of detectors around the imaging volume, as already shown in Figure 2.1. Due to interactions of the photons within the imaging object and the close time relation of the coincident photons, specific requirements arise for the detection system.

2.2.1 Requirements

After the application and distribution of the tracer inside the patient's body, a scan procedure has to detect the emerging photons from the positron-electron annihilation and convert the energy information to electrical pulses for further processing. Since a large fraction of the generated photons are scattered until they reach the detectors, their energy can range theoretically from almost 0 eV up to the full 511 keV from the positron electron annihilation. Since scattered photons also change their flight direction, the collinearity assumption of the two opposing annihilations photons is no longer valid for this case. This would lead to errors in the reconstruction process and therefore as well reduce the final image quality. Thus a sufficient detector energy resolution is necessary to distinguish between scattered photons which have already left some fraction of their energy and unscattered photons with 511 keV.

Another requirement for the detector implementation is time resolution. To be able to coincide the detection time of two photons from the same annihilation, a time resolution in the ns range is required. This allows to identify the positron annihilation as a *line of response* (LOR) between the two opposing detectors. The LOR information of multiple interactions is then used for the image reconstruction. If the time resolution can be reduced into the ps range, the concept of *time of flight* (TOF) PET becomes feasible. This allows also to confine the position of the annihilation along the LOR because of different travel lengths of the photons. Due to the velocity of the photons ($c \approx 3 \times 10^{10}$ cm/s), for example a 1 cm TOF spatial resolution would require a precision of about 66 ps in the

time information. The gained information can then improve the image quality, because the tracer localization can be determined more precisely with less photon events.

The spatial resolution of the detection system defines how precise the LOR confines the real positron annihilation event. This is influenced by the size of the individual detection elements as well as by their arrangement around the imaging object. The *field of view* FOV gives a value for the area that is covered by the detector system and which can be imaged. Other contributions come from the inevitable blurring due to the positron range in matter and the non-collinear annihilation because of the residual momentum of the positron when it annihilates (see Section 2.1). The minimum system spatial resolution is therefore limited at about 1 mm, depending on the detector design and also the applied tracer.

Figure 2.2 summarizes the specific requirements to a PET system due to the different photon interactions. In the optimal collinear case, the emitted photons hit two opposing detector pairs. The time resolution determines how good the two coincident photons can be distinguished from other interactions. If one of the photons is scattered, it is deflected from the real line of response and hits a nearby detector. Due to the energy loss caused by the scattering, such events can be identified and can be treated separately in the image reconstruction. The last case is an annihilation near the edge of the tomograph's field of view. Since the detectors have a certain radial extend, the achievable LOR information is broadened compared to an interaction in the center which degrades the image quality toward the edges of the FOV.

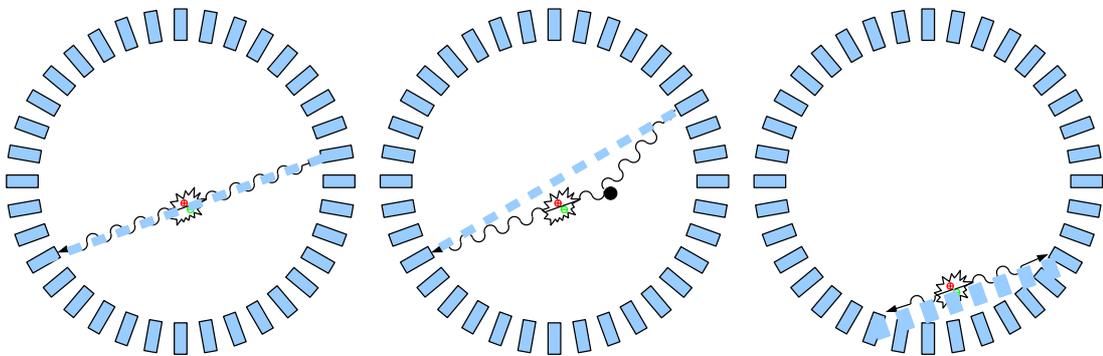


Figure 2.2: Detection of annihilation photons under different conditions. Perfect collinear case (*left*), scattered case (*middle*) and detection on the edge of the field of view (*right*). [CSP03]

2.2.2 Photon Detection

Compared to visible light in the energy range of about 2 eV to 3 eV, the positron annihilation photons have a much higher energy of 511 keV. Therefore, the conversion to an electrical pulse requires additional effort compared to a simple light detector. As already mentioned in Section 2.1, photons in the 511 keV energy range interact with matter either by Compton scattering or photoelectric absorption. Since Compton scattering requires

multiple interactions to convert the full photon energy, it is not guaranteed that the conversion takes place inside one detector. By scattering, the photon can as well be redirected to a neighboring detector and leave the remaining energy there. For good photon detection efficiency it is therefore beneficial to increase the interaction probability of photoelectric absorption, because this process converts all the energy within one step. Since this probability increases with higher atomic number Z of the detector (“heavier” detector materials) most detectors are based on “heavy” scintillator materials which convert the photon energy into visible light, combined with photodetectors to convert the light to electrical pulses. Recently, also pure semiconductor based detectors became available to allow more efficient and smaller detector modules.

2.2.2.1 Scintillator Based Detectors

The first usage of scintillator materials to convert high energy gamma radiation into lower energy (visible) light dates back into the first half of the twentieth century [Hof75]. Since then, various materials have been developed for this purpose and were used in applications ranging from research in high energy physics over industrial measurement devices to medical imaging. Characteristic properties for scintillators are the high density, respectively the high photon absorption efficiency, high light yield per detected photon and a well determined and fast light pulse output. Table 2.2 lists some commonly used scintillators for nuclear medicine applications.

	Material	Density	Z_{eff}	Photon yield (per keV)	Peak emission wavelength	Decay time
1	NaI(Tl)	3.67 g/cm ³	50	38	415 nm	230 ns
2	BGO	7.13 g/cm ³	74	8	480 nm	300 ns
3	LSO(Ce)	7.40 g/cm ³	66	20–30	420 nm	~40 ns
4	LYSO(Ce)	7.10 g/cm ³	60	32	420 nm	~41 ns
5	BaF ₂	4.89 g/cm ³	54	10	225 nm, 310 nm	0.6 ns, 630 ns
6	LaBr ₃ (Ce)	5.29 g/cm ³	46	63	380 nm	17–35 ns
7	LuYAP(Ce)	7.10 g/cm ³	60	13	375 nm	23 ns + slow

Table 2.2: Common scintillator materials. [K⁺07]

¹NaI(Tl): thallium-doped sodium iodine

²BGO: Bi₃Ge₄O₁₂, bismuth germanate

³LSO(Ce): Lu₂(SiO₄)O:Ce, cerium-doped lutetium oxyorthosilicate

⁴LYSO(Ce): (Lu-Y)₂(SiO₄)O:Ce, cerium-doped lutetium-yttrium oxyorthosilicate

⁵BaF₂: barium fluoride

⁶LaBr₃(Ce): cerium-doped lanthanum bromide

⁷LuYAP(Ce): Lu_{0.7}Y_{0.3}AlO₃:Ce, cerium-doped lutetium-yttrium orthoaluminate perovskite

The selection of a scintillator material contributes also significantly to the final system cost. Besides the high material and production costs for scintillator crystals, some materials are hygroscopic. The detector design has to account for this by proper encapsulation which makes handling and production even more difficult and expensive.

The emerging light from the scintillator is converted to electric pulses by a photodetector. Since the early scintillation detectors, the photomultiplier tube (PMT) is well established for this purpose. The advantages are its high gain (10^6 to 10^7), a fast signal rise time (about 1 ns) and good quantum efficiency at typical emission wavelengths of common scintillators. Additionally, photomultipliers are available in various shapes and sizes. However, as it is an electron tube based device there are also disadvantages: the required power supply is in the range of 1000 V and has a high influence on the amplification. Additionally, PMTs are very sensitive to magnetic fields, which requires magnetic shielding when they have to be used near magnetic fields. Since the various electrodes of the PMT are sealed within an evacuated glass tube, there is also a lower limit for the minimum size of a scintillator/PMT based detector.

As an alternative to photomultipliers, a semiconductor based photodetector can be used to detect the scintillation light. This allows to address the drawbacks of the PMT based detectors, namely size and the sensitivity to magnetic fields. The simplest detector would be a reverse biased silicon based photodiode. Incident light with a certain wavelength range can create electron hole pairs in the semiconductor material which are then separated by the biasing voltage. Since the typical scintillator light output for 511 keV photons is in the range of only a few thousand photons, the arising signals from simple electron-hole pair generation is quite small which gives a low signal to noise ratio (SNR). To overcome this limitation, avalanche photodiodes (APDs) use the generated charge from a conventional photodiode structure to create additional charge carriers in a high electric field avalanche region. By this, a gain in the order of 10^2 can be achieved. Figure 2.3 gives a schematic view of an avalanche photodiode structure.

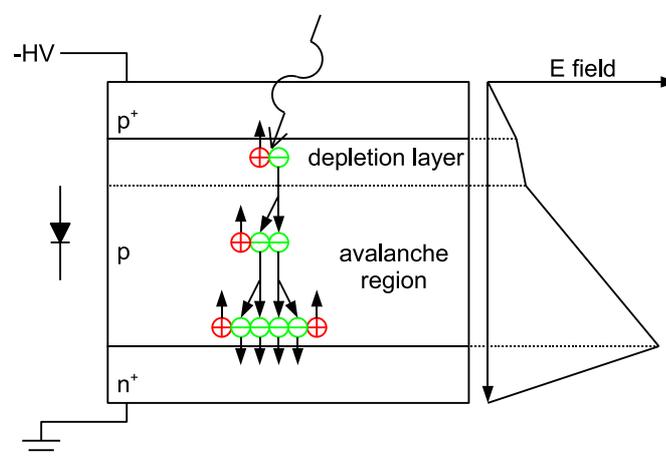


Figure 2.3: Schematic view of an avalanche photodiode. [HP04]

Avalanche photodiodes are already widely used as light detectors, but can not yet provide the high gain of PMTs. Therefore, additional low-noise amplification electronics is required before the electrical signal can be processed further. The supply voltage for APDs, which also influences the gain of the device, is in the range of 300–400 V. This corresponds to the linear operation region below the so called *breakdown voltage*, which generates some 100–1000 electron-hole pairs per incident photon.

Above the breakdown voltage the gain exponentially increases, resulting in high signal currents (in the mA range) already for single incident photons. Usually, this *Geiger-mode* operation is not used in standard APDs, since the proportionality between incoming photon flux and output signal is lost and the high current flow may lead to thermal damage of the diode structure. Especially designed single-photon or Geiger-mode avalanche photodiodes allow to detect very low photon fluxes and to derive a precise time information from the first incident scintillator photon, which results in typically very low time jitter in the ps region. However, to avoid thermal damage to the diode structure, special quenching circuits are required to stop the avalanche by e.g. lowering the bias below the breakdown voltage.

Recent developments have advanced the principle of Geiger-mode APDs into pixelated detector devices, called *silicon photomultipliers* (SiPM) [B⁺03, G⁺04, B⁺95] or *multi-pixel photon counters* (MPPC) [Y⁺06, G⁺06a, Y⁺07]. These detectors cover the active area with an array of small Geiger-mode APD cells, each with an independent quenching resistor. Once it is hit by an incoming photon, each cell gives a “digital” signal and quenches by itself due to the current flow in the quenching resistor. The sum of all cell signals then gives a quantized analog information about the incoming photon flux. This concept combines the good timing properties of the Geiger-mode APDs with the analog relation between photon flux and signal current in standard APDs. Figure 2.4 illustrates the SiPM detector principle.

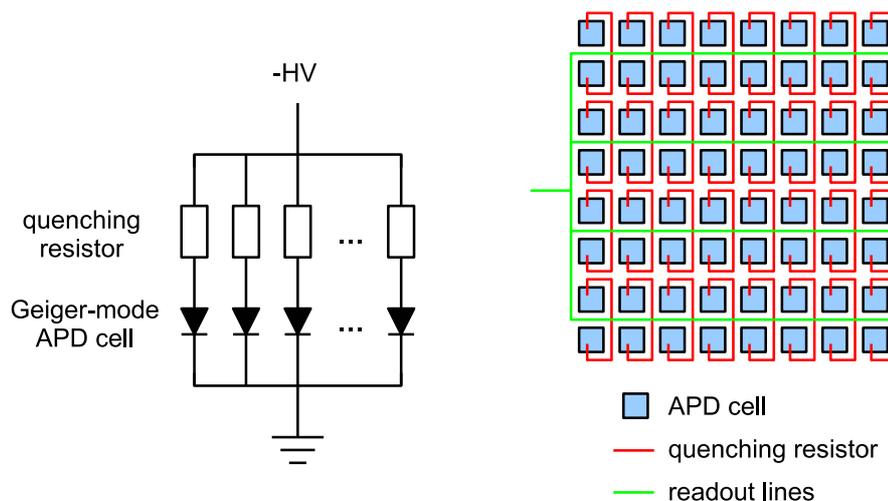


Figure 2.4: Schematic view of a silicon photomultiplier detector (*left*) and illustrated realization on silicon (*right*).

The number of cells can range from 100 to more than 10000 with typical sensitive areas starting at 1 mm² up to 9 mm². Due to the Geiger-mode operation of every single cell, the achievable gain is similar to conventional photomultipliers (about 10⁶). A drawback for the silicon photomultiplier design is the reduced quantum efficiency, since the quenching resistors and readout lines reduce the overall active area of the device. As every cell has specific dead-time properties and gives only a “digital” signal, irrespective of the number of simultaneous incident photons on that cell, the achievable energy resolution is closely related to the cell number of the device.

Similar to conventional APD devices, silicon photomultipliers are changing their properties with varying temperature. This affects in particular the breakdown voltage and thus the gain as well. The advantages over PMTs are the small size of the semiconductor devices which allows an increased readout channel density and the insensitivity to magnetic fields.

2.2.2.2 Semiconductor Based Detectors

In pure semiconductor based detectors, the gamma ray interaction directly creates electron hole pairs which can then be read as an electrical pulse. As standard semiconductor materials like silicon (¹⁴Si) or germanium (³²Ge) have a relatively low atomic number, the interaction probability with high energy photons is quite low compared to scintillators, which requires thicker detectors for good detection efficiency. However, the achievable energy resolution is superior, since more charge carriers (about 1 electron per 3 eV radiation energy) are generated compared to a scintillator/photodetector combination (about 1 electron per 100 eV). Unfortunately, silicon and germanium based semiconductor materials show a high thermal background noise at room temperature, compared to the expected gamma ray signals. To reduce the thermal noise, the detectors are usually cooled with liquid nitrogen ($T = 77 \text{ K} / -196 \text{ }^\circ\text{C}$), which causes complicated and expensive detector assemblies.

Compound semiconductors based on the IIb-IV elements (e.g. ⁴⁸Cd and ⁵²Te) overcome some of the drawbacks with silicon and germanium based semiconductor detectors. Due to the higher atomic numbers of the source materials, the compounds CdTe⁸ and CdZnTe⁹ (CZT) show a higher gamma ray stopping efficiency compared to pure silicon or germanium. Additionally, these detector materials can be operated at room temperature or only moderate cooling, since the thermal noise is significantly lower. Therefore, pixelated detector modules are possible with compact integration of the readout electronics. However, there are still technological challenges in the production process, regarding the maximum crystal size of the semiconductor and its purity which influence the price and performance of these detectors.

⁸CdTe: cadmium telluride

⁹CdZnTe: cadmium zinc telluride

2.2.3 Detector Systems

As described in Section 2.1, the positron annihilation emits two collinear photons with the same energy of 511 keV. To be able to detect both photons and to define a LOR for the reconstruction process, photon detectors have to be installed at opposing sides of the imaging object. Simple systems may achieve this by two parallel detector planes, which are rotated around the patient in order to acquire data from all emission angles. Though this concept reduces the detector costs, it has only a limited detection efficiency since most emitted photons do not pass any of the two detectors. To compensate for the reduced photon statistics, either the measurement time or the applied tracer activity has to be increased, which either is not desirable. To increase the number of detected photons, common PET systems use a full ring of detectors which surrounds the patient. Almost all clinical systems are based on scintillator/photomultiplier based detectors, whereas the scintillator/photodiode based systems are mainly used in small animal imaging. Pure semiconductor detectors (CdTe, CdZnTe) are currently only used in prototype research systems and are not yet commercially available.

To achieve sufficient spatial resolution, the required detector crystal size has to be in the region of about 2 mm to 6 mm and provide coverage with ideally no dead areas. The *block detector* concept allows to read many relatively small scintillator crystals with less photomultipliers of a much bigger size. Therefore, a solid scintillator block (BGO or LSO) is segmented into an array of smaller independent crystals which couple their scintillation light into a common light guide. This shared light guide is then read by photomultipliers which allow to reconstruct the initial crystal position by calculation and discrimination of the x and y position through weighting of the PMT signals. The block detector reduces the number of readout channels but puts constraints on the maximum count rate, since the crystal identification is perturbed by multiple hits in the same block. Due to the fast signal rise time of the PMTs, block detectors show good timing properties. Figure 2.5 shows a typical block detector with 8×8 scintillator crystals, read by four photomultipliers which allows to scale the number of required readout channels by a factor of 16.

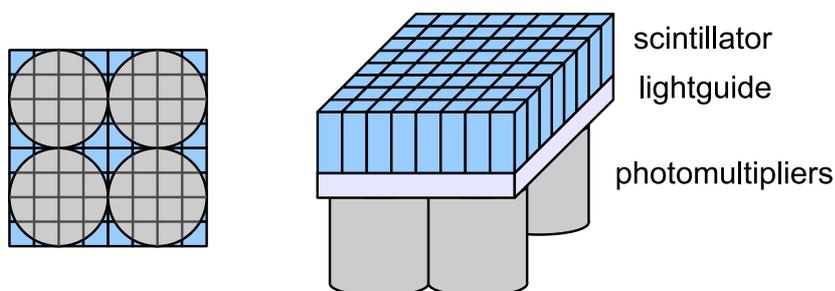


Figure 2.5: Schematic view of the block detector concept. [Phe06]

The substitution of the PMTs by photodiodes allows to reduce the size of the detector system, however at the cost of additional amplification electronics for the diode signals. With smaller photodetectors or the monolithic integration of photodiode arrays, even the independent readout of each scintillator crystal becomes feasible and avoids the crystal identification problems of the block detector. However, the number of readout channels increases dramatically, thus this concept is only used for small PET systems.

In order to fully absorb the 511 keV photons in the scintillator material, the crystal length has to be in the order of 20 mm to 30 mm. If no information on the interaction position along the scintillator is available, the spatial resolution degrades outside the center of the field of view (FOV), due to parallax errors. If *depth of interaction* (DOI) information is available, the photon interaction can be confined along the scintillator crystal and allows to enhance the spatial resolution on the edges of the FOV. This effect is illustrated in Figure 2.6. The DOI information can be achieved by means of different methods: Depending on the interaction depth of the photon, the scintillation light has to travel different lengths in the crystal to reach the photodetector. In block detectors, the varying spread of the light can be used to extract the interaction depth in the crystal. Another method separates the single crystal into multiple segments (normally two) which are either made from different scintillator materials with different decay constants or which are read with independent photodetectors. The last implementation doubles the number of readout channels and requires a thin photodetector for the first layer readout, but has the advantage that both detector layers are completely independent. The approach with different decay constants (*phoswich detector*) keeps the readout channel count low but requires pulse analysis of the detector signal to distinguish between both scintillators. Additionally, the different scintillators will also have divergent efficiencies which degrades the overall detector performance. By connecting photodetectors on both ends of a scintillator crystal the actual interaction position inside the crystal can also be determined.

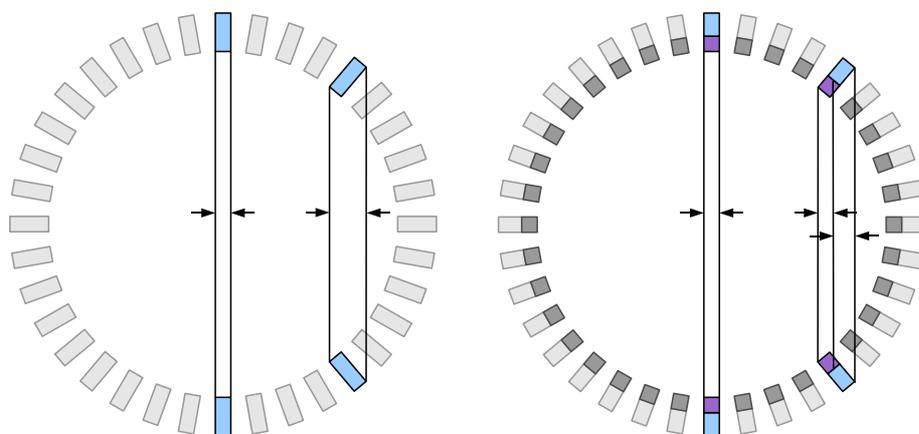


Figure 2.6: Improvement of spatial resolution by using depth of interaction (DOI) information (*right*).

Since the detector modules which surround the imaging object are build from a matrix structure of scintillators, this forms several detector planes along the axial direction of the PET scanner. Depending on the allowed photon coincidences between the different planes, the system sensitivity can be increased at the cost of higher data rates and more complex image reconstruction procedures. In a simple 2D data acquisition, collimators are placed between adjacent detector planes to ensure that only coincidences from the same plane are accepted. This produces a series of independent 2D transaxial images which are all acquired with the same sensitivity. In 3D acquisition mode the collimators are removed and coincidences are allowed between all planes along the axis. This increases the sensitivity in the center of the system, since this point is covered by most of the detectors. Since the angle of incidence of each detector is not limited in 3D mode, also more scattered events are now detected (up to 70% of the total events) and have to be processed by the readout electronics. By allowing correlations only between neighbouring detector planes, the sensitivity can still be increased without the large increment in scattered events as in full 3D acquisition mode. Figure 2.7 illustrates the concepts of 2D/3D data acquisition.

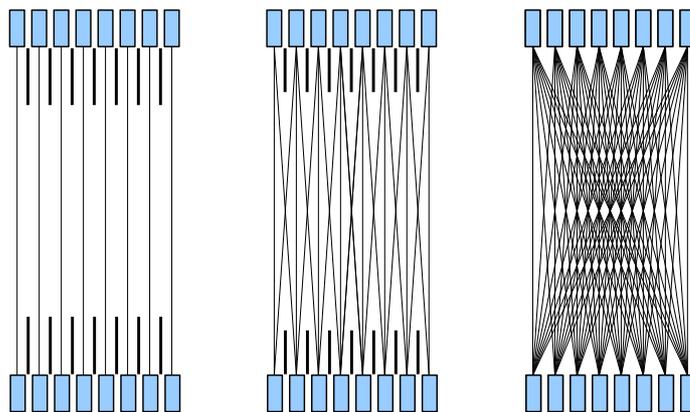


Figure 2.7: 2D/3D data acquisition concepts. Simple 2D acquisition (*left*), enhanced sensitivity 2D acquisition (*mid*) and full 3D acquisition (*right*). [Phe06]

2.3 Image Reconstruction

Through the image reconstruction procedure, data from the individual photon detectors are combined to visualize the tracer distribution inside the PET scanner. Ideally, this allows a precise determination of the acquired amount of radioactivity in the imaged object and therefore to quantize the rate of biological activity. As a first step, the events from opposing detector pairs are combined to LORs which intersect the imaging object. Since the coincidentally emitted photons hit the detectors at almost the same time (within some 100 ps due to time of flight effects), this is achieved by searching for events which occurred within a certain *coincidence time window*. Because of the limited time resolution of the scintillator/photodetector combination and the readout electronics, typical window

sizes are in the order of about 6 ns to 12 ns. If two photon events fall into this range, they are classified as a valid coincidence and the corresponding LOR is accepted.

Given that the radioactive decay of the tracer happens at any time, also *random coincidences* are detected by this, if two detected photon signals arise from two different positron decays at almost the same time. To reduce the background which is introduced by the random coincidences their respective rate can be estimated statistically and subtracted from the total number of coincident events for each detector pair. This is especially important if a quantitative information about the tracer distribution is required. Ideally, the random coincidence rate is already reduced by selecting the coincidence window as small as possible, typically twice the time resolution of the detector system. Additional background arises from the detection of scattered photons which give a wrong LOR information for the decay position, compared to unscattered events. The correction for scattered events requires more elaborative data processing, since it is influenced by the scatter probabilities of the imaged object and the detectors. As well, the photon attenuation inside the imaged object has to be corrected. Figure 2.8 shows the three possible coincident event types and their corresponding LORs. Depending on the actual tracer activity, also multiple coincident or random events can occur in the detector ring. If the ambiguity of multiple hits can not be resolved by sufficient information about interaction time or photon energy, usually these events have to be ignored in the reconstruction process.

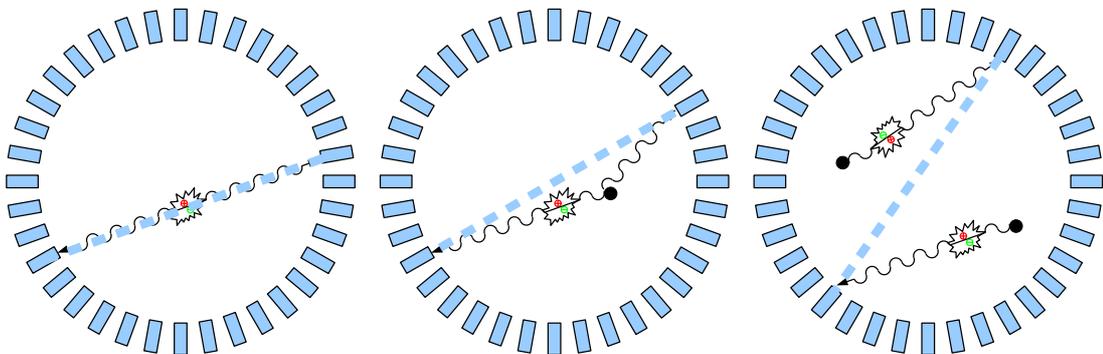


Figure 2.8: Coincident event types and corresponding line of response (LOR). True coincidence (*left*), scattered coincidence (*mid*) and random coincidence (*right*). [Phe06]

After the LOR corrections, the transition to image data can be implemented by either analytical or iterative reconstruction methods. The less computing intensive analytical approach transforms the LOR data directly into a cross-sectional image of the tracer distribution. Therefore, several projection profiles $p(r, \phi)$ of the imaging object under different projection angles ϕ are derived from the LOR data. Via *backprojection* of all N profiles a reconstructed image $f'(x, y)$ of the true tracer distribution $f(x, y)$ can be computed:

$$(2.2) \quad f'(x, y) = \sum_{i=1}^N p(r_i, \phi_i),$$

$$(2.3) \quad \text{with } r_i = x \cos \phi_i + y \sin \phi_i.$$

Unfortunately, the image quality of simple backprojection is distorted since the inner part of the image (small r values) is covered by more projections than the outer part. This introduces a $1/r$ blurring compared to the true image:

$$(2.4) \quad f'(x, y) = f(x, y) * (1/r).$$

The reconstruction artifact can be avoided by more advanced reconstruction algorithms, upon the most widely used is *filtered backprojection* (FBP). FBP applies in addition a high-pass filter on the projection data to remove the $1/r$ blurring. Since this enhances high frequency noise as well, the reconstructed images also appear more noisy compared to simple backprojection.

To enhance the image quality further, more computing-intensive iterative reconstruction methods can be used. These algorithms start with an estimation $f^*(x, y)$ of the reconstructed image and derive the projection data $p^*(r, \phi)$ of the estimation via forward projection. The projection data are then compared to the real measured projections to determine how the initial estimated image f^* must be modified to better match the measured data. The error values e in the projection space are then back-projected to the image space to get the actual image errors which are used to update the estimation f^* for the next iteration. This process is repeated until a certain convergence criteria is fulfilled and the the image estimation f^* corresponds to the measured data precisely enough.

The advantage of iterative methods is that more information about the detector system can be used to achieve a better quality in the reconstructed image. The forward projection process can include detection probabilities for each individual detector element if exposed to a positron source at a certain position inside the active volume of the scanner. The *system matrix* for the forward projection can be obtained analytically or by simulating the full detector system, including all physical interactions of the radiation with the components of the scanner. However, since each iteration step includes a full backprojection operation, the required computing resources are substantially higher than for analytical methods. Additionally, the system matrix can become quite large since it includes relations between all reconstruction image pixels and the detector projections. For 3D image reconstruction the size of the system matrix is even increased further and can reach sizes up to several gigabytes, which also complicates the reconstruction process.

Since time is a critical aspect for clinical examinations, the analytical reconstruction algorithms were for a long time the only feasible reconstruction technique for PET, although

limited in image quality. Nevertheless, since the available computing power is continuously increasing, also iterative image reconstruction techniques are now available for clinical use. However, since the different reconstruction methods also show different noise properties, more care has to be taken by the examining physician in the interpretation of the acquired PET images as more reconstruction options are available.

2.4 Multimodality Imaging

Another challenging task during the medical review of PET images is the precise location of lesions inside the patient. Since the plain PET images are only originating from the tracer positron emission, other anatomical details besides e.g. the tumor are only hardly visible. To get a better picture of the patient's anatomy, other diagnostic imaging procedures like X-ray computed tomography (CT) or magnetic resonance imaging (MRI) are needed [Che06]. These imaging modalities allow nowadays an accurate view of soft tissue and bones inside the body and to identify the different organs. During an *image fusion* procedure the CT or MRI images have then to be combined with the information from the PET scan. Additionally, the CT provides also density information which is needed for proper attenuation correction of the body tissue in the PET image reconstruction.

To simplify the procedure of multimodality imaging and to avoid image mismatches due to intermediate patient movement, combined scanner systems for PET/CT are available today [SMS07] and used in everyday clinical operation. Unfortunately, the X-ray based CT imaging adds additional dose to the patient, besides the radiation from the PET tracer. Therefore, the transition from CT to MRI would be favorable to reduce the overall dose. Besides the higher soft-tissue contrast, MRI provides in addition functional imaging possibilities, which are not available in CT [vSS09]. Compared to CT scanners, MRI systems are though more complex and put higher constraints on the added PET electronics. Currently, combined PET/MR systems are in development and first prototype systems appeared [Sch08]. Due to the sub-millimeter resolution of current MR systems, this combination has its application especially in the field of neurological science and brain imaging.

Especially challenging for the combination of PET and MRI systems is the integration of scintillator based detectors into existing MRI systems. Since MRI relies on a well determined and uniform magnetic field in the imaging volume, additional detector components are restricted e.g. to non-magnetic materials. Further on, the available space for additional PET detector components is limited by the confined inner diameter of the MRI system. In addition, the MRI scanner generates powerful radio-frequency bursts during its operation, which may interfere with the electronic circuits of added detector assemblies.

Chapter 3

The MADPET-II Small Animal PET System

The development and evaluation of new drugs, as well as the understanding of diseases and their treatment makes it necessary to do examinations and studies with living biological systems. Although the basic metabolism processes may be studied with simple cell cultures on the laboratory level, there remain still uncertainties once these results are scaled to the complex biological system inside the body. By careful pre-clinical studies of new treatment methods with small animals like mice or rats, the risk for a following human application has therefore to be minimized. Additionally, the evaluation of the effectiveness of different therapy options is hardly possible or ethically acceptable with human beings. Hence, some quantification with animal studies is necessary during the medical development process. The technological improvement in nuclear imaging helps in this field, by providing non-invasive measurement options which allow to acquire multiple measurements to reduce the number of required animals for such studies.

3.1 Clinic versus Small Animal PET Imaging

Compared to “standard” clinical PET scanners, the imaging of small animals obviously has to deal with smaller imaging objects than the human sized systems. This difference in size imposes also slightly different requirements to the corresponding detector systems. Since typical clinic systems provide an inner diameter for patient access in the order of 70 cm, the total number of scintillator crystals is in the order of 24336 to 32448 (Siemens Biograph™ TruePoint PET/CT [SMS07]). For a cost effective readout, the clinical systems use the block detector concept which reads several scintillator crystals with fewer photomultiplier tubes. For the Biograph™ system, 169 LSO crystals are combined into a block which is read by four photomultipliers. This scales the number of required readout channels from 32448 down to 768. Due to the combined optical readout of several crystals in the block, the actual crystal position has to be reconstructed from the photomultiplier signals. Since this process is distorted when multiple crystals within a detector block are hit,

this limits the maximum count rate of the total system. The achievable spatial resolution for such a system is in the order of 4–5 mm. With optimized reconstruction algorithms this can then be further reduced down to about 2 mm.

For small animal PET systems, the field of view and the inner diameter of the scanner can be reduced since they are only focused on animals with the size of a few centimeters. This reduces already the number of required detector channels. The small size of the imaged animals requires however an increased spatial resolution of the PET system, since the lesions that have to be imaged are now also scaled to only a few millimeters. This can be achieved by reducing the pixel size of the scintillator crystals, which however increases again the detector channel number. Additionally, the average count rate for each scintillator crystal is increased, since the imaging object is now closer to the detector ring as compared to a clinical system. This may limit the system performance due to increased distortions in the crystal identification process of a block detector concept.

Due to the small inner diameter of the detector ring and a typical scintillator length of about 20 mm, parallax errors on the edge of the FOV become critical. To resolve these errors, additional information about the interaction position inside the scintillator crystal (depth of interaction) is required.

3.2 The MADPET-II Detector Concept

The *Munich Avalanche Photodiode Positron Emission Tomograph* (MADPET) was the first system that implemented a 1:1 coupling between LSO as scintillation crystals and avalanche photodiodes (APDs) as light detectors [Zie01]. This concept avoids the crystal identification step of the block detector and therefore allows to measure higher count rates without distortions. Another advantage is the small size of the APD photodetectors and the reduced operation voltage compared to photomultiplier tubes. The system is build from six detector modules, each including a 2×8 array of LSO/APD detectors which gives 96 readout channels. Since the detector modules do not cover a complete ring, most emerging photon pairs never reach a detector module which limits the sensitivity of the system. Additionally, the number of projections, covered by the six detector modules, is not sufficient for a successful image reconstruction. Therefore, to cover more projections of the imaging object, the detector modules have to be rotated, to achieve a uniform coverage of the imaging object.

The MADPET-II system continues the concept of a 1:1 coupling between LSO and APDs and tries to resolve the limitations of the prototype system. A total number of 18 detector modules as shown in Figure 3.1 provide an almost full coverage of the gantry with scintillator material. The next generation of APD matrices combines 4×8 pixels, each with 2×2 mm² active area on one ceramic carrier. To address the limited resolution on the outer FOV due to parallax errors, each detector module is divided into two layers which are independently read by APD arrays. The first layer, which forms the innermost ring, is build from 6 mm long scintillator crystals, whereas the second layer includes 8 mm

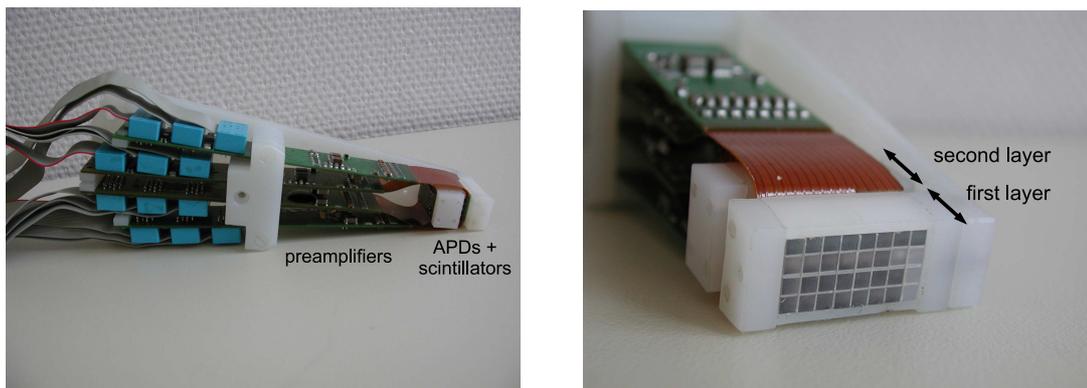


Figure 3.1: MADPET-II dual layer detector module with integrated preamplifiers (*left*) and close-up of the scintillator crystals of the first layer (*right*).

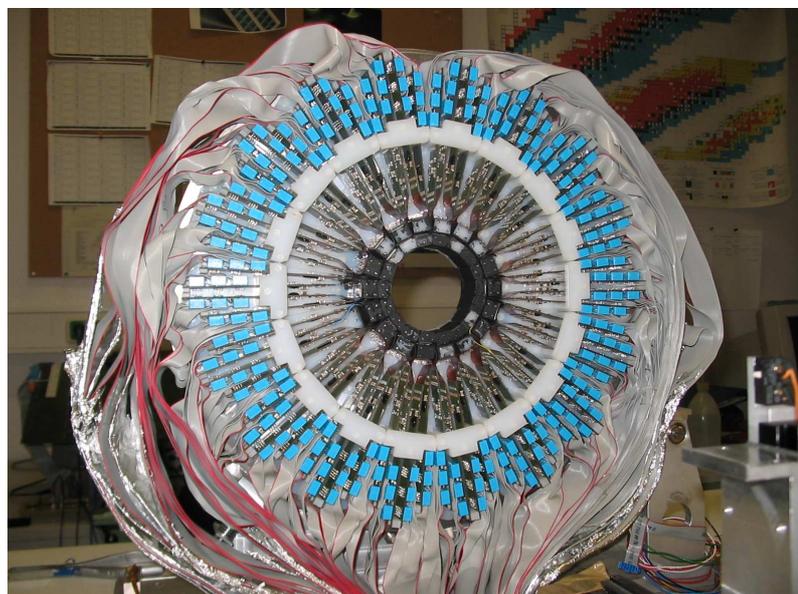


Figure 3.2: MADPET-II detector front-end.

long crystals. This allows to reach a better confinement of the position information in the outer FOV compared to monolithic 14 mm long scintillators which was evaluated with Monte Carlo simulations [Raf01]. Figure 3.2 shows the fully equipped MADPET-II detector front-end.

The simulations showed that about 40% of coincident photon pairs are both detected in the first crystal layer and about 14% only in the second layer. Though, about 46% of the coincidences are detected in both layers. Since the 511 keV photons can undergo multiple scatter interactions in the scintillator material until they are fully absorbed, it is also possible that a single photon is scattered in multiple crystals. Due to the 1:1 readout of the scintillator crystals, each signal from a scatter interaction can be acquired separately and is available for the following reconstruction process. Therefore, it is necessary that the data are acquired in *singles-list-mode* format which records an independent event with timestamp, position and energy information for every single detector interaction. This



Figure 3.3: Hamamatsu Photonics S8550 8×4 APD array (left) and APD array with mounted scintillator crystals for the first detector layer (right).

allows different reconstruction methods to operate on all available data. Further Monte Carlo simulations of the detector system were also used to derive a system matrix of the detector system which is used for the iterative image reconstruction process [Raf04].

3.2.1 APD Readout Electronics

The light from the LSO scintillator crystals is detected by avalanche photodiode (APD) arrays (S8550) which are manufactured by Hamamatsu Photonics [Pic01a, HP06]. Figure 3.3 shows the APD module and the mounted scintillator crystals for the first layer of the tomograph. The array is build from two monolithic 8×2 APD matrices with common cathode which are both independently biased by high voltage power supply channels. The individual APD channels are independently connected to charge sensitive preamplifiers via the scheme shown in Figure 3.4. 16 preamplifier channels are implemented in an ASIC [Pic01b] and directly mounted and bonded onto the front-end printed circuit boards (PCB). The ASIC also includes a differential line driver to transmit the amplified signal to the data acquisition electronics.

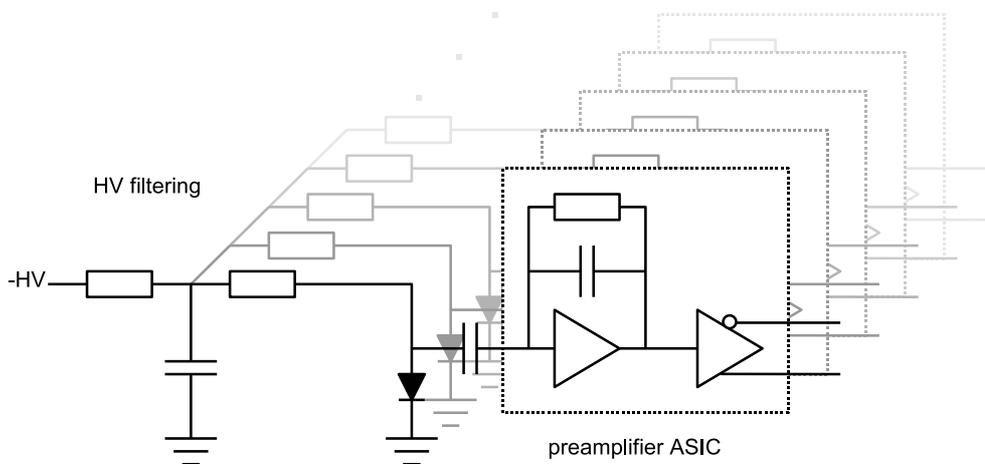


Figure 3.4: APD high voltage supply circuit and preamplifier connection.

To be able to have as less material near or between the scintillator crystals, the readout electronics is mounted on a flexrigid PCB which allows the mounting of all electronics in the pie-slice shaped module shown in Figure 3.1. Every 16-channel preamplifier module is then connected to the data acquisition system via a flat cable, which also provides the supply voltage for the ASIC and the APD bias voltage.

3.2.2 Data Acquisition

The 32 differential signal lines from one APD array are connected to a 9U¹ VME² based data acquisition (DAQ) board. The DAQ board is responsible for pulse detection and extraction of time and amplitude information. The timestamp of each detector event allows later on the matching with the event of the coincident photon which may be detected in an opposing detector element. The amplitude information corresponds to the energy of the interacting photon and enables a discrimination of compton scattered events from the 511 keV signals.

In the MADPET-II system, the time and amplitude detection task is implemented in analog signal processing ASICs [Spa06], which include four independent processing channels for the preamplified APD signals. Each channel implements as input stage a differential line receiver with selectable gain for the preamplifier signals. Following the receiver, the signal is split into a shaping stage and a passive CR high pass filter to derive signals for a constant fraction discriminator (CFD) stage to extract the time information of the signal. The shaped signal is also evaluated by a peak detector circuit to store the signal amplitude. To be able to set amplitude trigger thresholds, the shaped signal is also compared to an external reference voltage to generate an enable signal for the CFD output. The ASIC is mounted together with tuning resistors on a small mezzanine card which is then connected onto the VME carrier card.

For each of the four channels, the analog ASIC gives a digital signal from the constant fraction discriminator and an analog signal from the peak detector. The CFD signal which serves also as trigger signal is then evaluated by a 15 bit time-to-digital converter (TDC) on the VME board, which gives a time value with a 333 ps quantization. As well, the analog peak detector output is digitized by an external 10 bit successive approximation analog-to-digital converter (ADC). To generate the trigger threshold voltage an additional digital-to-analog converter (DAC) is required for each channel. The readout of the TDCs and ADCs and a first buffering of the detected events is implemented in field programmable gate arrays (FPGAs). A detailed description of the existing data acquisition system is given in [Spa08]. Table 3.1 summarizes the number of required components for a single VME board and the full system with 1152 readout channels.

¹**1U**: 1 rack Unit = 1.75 inch \approx 44.45 mm. As the rack unit corresponds to the size of the enclosing VME crate, the module itself is slightly smaller than 9U.

²**VME**: Versa Module Eurocard

	VME board	Full system
Channels	32	1152
VME boards	1	36
ASIC	8	288
TDC	32	1152
ADC	32	1152
DAC (4 channels)	8	288
FPGA	8 + 1	324
Total number	89	3204

Table 3.1: Component count for the existing MADPET-II data acquisition system.

Since the 36 VME boards do not fit into a single VME crate, the total system is distributed between two crates. Each crate is controlled by a clock and control interface board (CCI) which is responsible for supplying a central clock and configuration data to all boards within a crate. The event data from the data acquisition boards are read via the VME backplane and transferred via a Gigabit Ethernet interface on the CCI board to the control computer and disk storage.

3.3 Detector Calibration Procedures

Given that the gain of the APD photodetectors and also the LSO scintillator light output is temperature dependent, the relation between the detected photon energy and the amplitude of the measured electrical signal is not fixed [Spa05b]. Additionally, the detector channels have slightly differing signal properties like rise time and propagation delay values due to component variations or cable length. As a result of the temperature dependence, it is required that the whole detector system is operated within well defined environmental conditions, especially at a constant ambient temperature. Additionally, once a constant operating temperature is reached, the detector system has to be calibrated before real PET measurements can be done.

For the calibration of PET systems various protocols are established, which allow a reliable and time efficient calibration and quality assurance during everyday clinical use of the system. For the MADPET-II system, a radioactive “flood-source” is placed inside the PET scanner to illuminate all photon detectors uniformly. Typically ^{18}F -FDG is used for this purpose as it is easily available. The calibration procedure then involves the following steps:

- Adjustment of the trigger thresholds,
- Event data acquisition for offline data processing,
- Energy calibration from offline data,
- Time calibration from offline data.

The trigger threshold adjustment ensures equal trigger rates on all channels to get an even event statistics over all detectors. This first step is necessary since the signal amplitude may vary significantly due to APD gain and preamplifier variations. The energy and time calibration is then derived from recorded data after the acquisition.

3.3.1 Energy Calibration

The energy calibration derives a coefficient for each channel which relates the digital value of the extracted signal amplitude with a physical meaningful energy value of the detected photon. Therefore, the ADC event amplitudes of all channels are independently histogrammed to extract the mean value μ_p of the 511 keV ^{18}F -photopeak from the spectrum for example by a Gaussian fit. The energy calibration coefficient can then be calculated by $c_{\text{ADC} \rightarrow \text{Energy}} = 511 \text{ keV} / \mu_p$. Figure 3.5 shows the energy histogram of one MADPET-II APD channel, obtained from sampling ADC data and a simple peak hold algorithm (see Section 4.3.3.1). Due to the trigger conditions of the ADC which are described in Section 4.2, there is no sharp amplitude cut for low energies ($E < 70$ ADC channels) but a smeared edge instead.

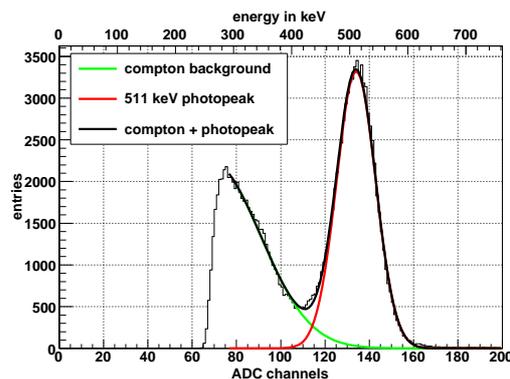


Figure 3.5: Energy histogram for a single MADPET-II detector channel. A sum of two Gaussian functions (*black*) is fitted to the raw data, modeling the photopeak (*red*, $\mu_p = 134$, $\sigma_p = 8.81$) and the compton scattered background (*green*, $\mu_c = 68.8$, $\sigma_c = 22.1$). The calibrated energy axis is shown on the top, based on the position of the 511 keV photopeak.

The energy calibration routine now fits a sum of two Gaussian functions to the amplitude histogram and uses the mean value of the photopeak to determine the energy calibration coefficient.

The fit function is given by

$$(3.1) \quad f(x) = A_c \cdot e^{-\frac{(x-\mu_c)^2}{2\sigma_c^2}} + A_p \cdot e^{-\frac{(x-\mu_p)^2}{2\sigma_p^2}}.$$

The calibration for one channel results in the energy scaling shown on the top of the histogram in Figure 3.5. With the calibrated energy values it is then possible to define physically more meaningful trigger thresholds in keV for all front-end channels simultaneously. The offline data processing can therefore reject for example compton scattered photons up to a certain energy level and use only events from the photopeak.

In addition, the fit results also provide information about the achievable energy resolution of the detector system. The energy resolution can be calculated from the mean μ_p and standard deviation σ_p of the photopeak by

$$(3.2) \quad (\Delta E/E) = \frac{\sigma_p}{\mu_p},$$

or

$$(3.3) \quad (\Delta E/E)_{\text{FWHM}} = \frac{2\sqrt{2 \ln 2} \cdot \sigma_p}{\mu_p} \approx \frac{2.35 \cdot \sigma_p}{\mu_p}.$$

In literature, the energy resolution is usually quoted with the *full width at half maximum* (FWHM) value given in (3.3), which corresponds to the extent of the photopeak at half of its amplitude. For the exemplary detector channel in Figure 3.5, the energy resolution calculates to 6.6% or 15.5% FWHM, respectively. The obtainable energy resolution for the sampling ADC based data acquisition system for MADPET-II is discussed in more detail in Section 6.2.1.

3.3.2 Existing Time Calibration

The time calibration has to correct for time delay offsets between the APD channels, arising from unmatched cable lengths and other component variations which may change the signal timing properties. Since each detector element is compared with many opposing detectors during the image reconstruction to search for coincident events, large channel delay mismatches would as well require to choose a larger coincidence window. However, a larger coincidence window increases the probability to detect random coincidences which results in a reduced signal to noise ratio in the reconstructed image.

To search for the coincidence events between two detector channels—which define a so called line of response (LOR)—the time-stamps of all events from one detector channel are subtracted from all time-stamps of the other channel to obtain a time difference histogram. Ideally, for only coincident events in the two detectors, this results in a peak at a time difference value of zero. Due to the limited time resolution of the detectors and readout electronics, the peak is not ideally sharp and may be approximated by a Gaussian shape. Figure 3.6 shows an example configuration with four detector channels, resulting in four possible LOR combinations.

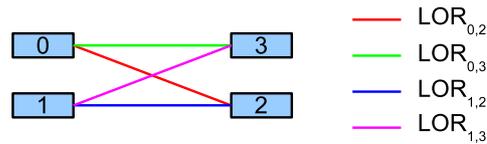


Figure 3.6: Example of 2 + 2 detectors in coincidence, resulting in 4 possible line-of-response (LOR) combinations.

The size of the coincidence window then defines what range around the zero value is accepted for coincident events and used for the image reconstruction. All remaining events outside the coincidence window are treated as random events and therefore discarded. The problem of a time-misalignment between the individual detector channels is illustrated in Figure 3.7. For uncalibrated time delay offsets, the overall time difference histogram is broadened due to the shifted LOR time differences, which would require a wide coincidence window setting. In the calibrated case, the positions of the LOR time differences almost match, thus resulting in a narrow system-wide time difference histogram and thus a small coincident window setting.

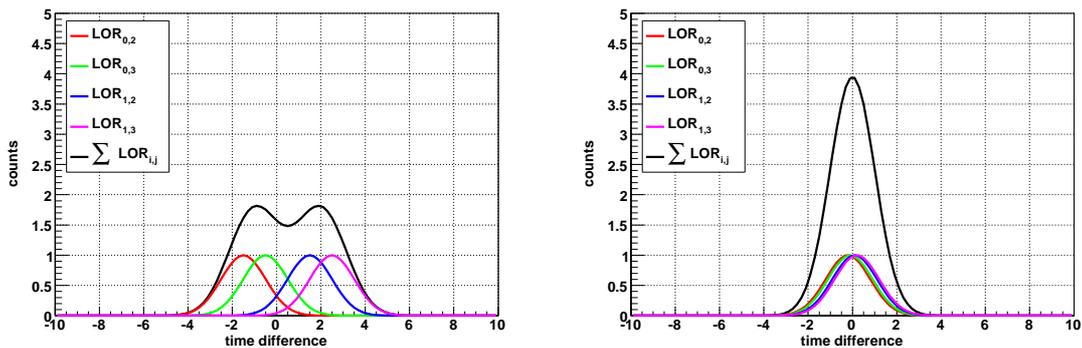


Figure 3.7: Illustration of the time-misalignment influence on the time resolution. For misaligned LOR time differences, the system-wide time difference histogram broadens (*left*). For calibrated channel delays, the mean values of the LOR time differences are almost equal, which results in a narrow system-wide time difference histogram (*right*).

To optimize the time resolution of the full detector system, the individual channel delays have to be determined and included in the data analysis. Therefore, the additive delay corrections for all channels can be derived by two possible ways. Either by measuring relatively to an external timing probe reference [McE07] or by extraction from an acquired

flood-source dataset. Both methods determine a correction delay δ_k for each channel k , which is then added to all raw timestamps $\mathbf{t}_{\text{raw}}^{(k)}$ which are acquired for the corresponding channel k

$$(3.4) \quad \mathbf{t}_{\text{cal}}^{(k)} = \mathbf{t}_{\text{raw}}^{(k)} + \delta_k.$$

Ideally, the correction delay then shifts the LOR time differences in such a way, that the individual mean positions overlap as shown on the right side of Figure 3.7.

3.3.2.1 Timing Probe Calibration

The timing probe consists of a positron emitting point source with a rather long half-life time (like ^{68}Ge with $T_{1/2} = 271$ days or ^{22}Na with $T_{1/2} = 2.6$ years) which is directly embedded into a plastic scintillator. Therefore, emitted positrons can give an additional signal in the scintillator prior to annihilation into the two 511 keV photons. By coupling the scintillator to a fast photomultiplier, an electrical reference signal for the positron decay events can be obtained. The calibration delays for each channel can then be extracted from the time difference between the reference signal and the event timestamps from the registered 511 keV photons in the MADPET-II detector channels. Figure 3.8 illustrates the measurement setup and the evaluated coincidences for the timing probe based calibration.

A drawback of the timing probe procedure is that the maximum activity of the embedded positron source is limited due to size and saturation/pileup effects in the readout system. In addition, the activity constantly degrades further with the half-life time of the used isotope. Since a certain number of events is required to achieve reliable results, the measurement time for the calibration increases proportionally as the source decays. Furthermore, the timing probe requires one additional acquisition channel with very good time resolution as well as an additional high voltage supply channel for the photomultiplier.

3.3.2.2 Iterative Time Calibration

Alternatively, the delay values may be derived from an acquired flood-source dataset by an iterative software algorithm [Spa05a]. This allows to use for example the easily available ^{18}F -FDG positron emitter with a sufficiently high activity. The size of the calibration source should therefore match the field of view of the detector system, since only LORs which are traversing the source are evaluated as shown in Figure 3.9.

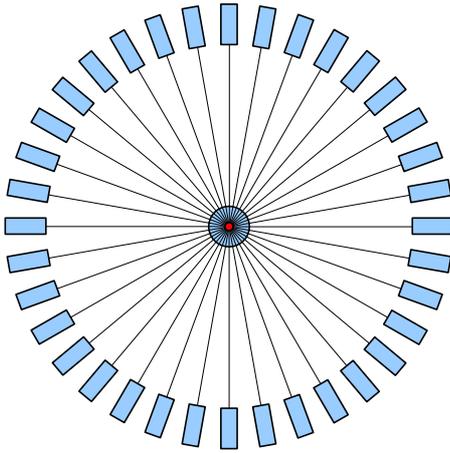


Figure 3.8: Measurement setup for the time calibration with a centered timing probe. The radioactive source (*red*) is embedded in the timing probe scintillator and the coincidences are evaluated between the timing probe and each detector channel.

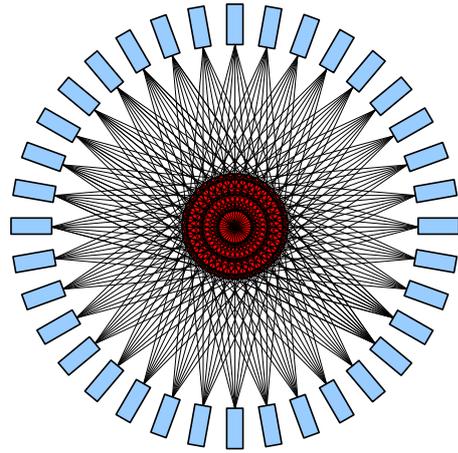


Figure 3.9: Ideal measurement setup for the iterative time calibration. The radioactive flood source (*red*) is centered in the scanner and the coincidences which are crossing the source are evaluated.

To simplify the determination of the LOR geometry, the scanner system may be logically divided in fixed segments (e.g. two half circles). The coincidences can then be evaluated between channels in different segments. For the division into two segments, the total number of LORs is $L = (K/2)^2$ for a total number of K channels in the system. Since the iterative method analyzes all valid LORs of the detector system, it requires a higher number of recorded events compared to the timing probe measurement to achieve sufficient statistics in the individual time coincidence histograms of each LOR.

The coincidence histograms are obtained by evaluating the time difference between all events of two opposing detectors which define the corresponding LOR l . From the resulting distribution, a mean time difference can then be extracted for each LOR, e.g. by fitting a Gaussian function to the histogram data. Due to the individual channel delays, the resulting mean values μ_l will differ for all L LORs. Once all LORs are combined to a system-wide coincidence histogram by summing all contributions, this effect leads to the mentioned broadening of the time difference peak, thus degrading the overall time resolution.

The iterative time calibration now tries to determine the correction delays δ_k in a way to minimize the resulting standard deviation of the calibrated mean values. Therefore, the calibrated mean values are calculated by

$$(3.5) \quad \mu_{\text{cal}}^{(l)} = \mu_l + \delta_{k_a(l)} - \delta_{k_b(l)},$$

where $k_a(l)$ and $k_b(l)$ are functions of the LOR number l , which give the two channel

indexes that define the LOR. The standard deviation of the resulting system-wide time histogram can then be expressed as a function of the correction delays [Spa05a]

$$(3.6) \quad f(\delta) = \frac{1}{\sqrt{(K/2)^2 - 1}} \cdot \sqrt{\sum_{l=0}^L \left(\mu_l + \delta_{k_a(l)} - \delta_{k_b(l)} - \mu_{\text{offset}} - \frac{\sum_{i=0}^{(K/2)-1} \delta_i - \sum_{j=K/2}^{K-1} \delta_j}{K/2} \right)^2},$$

with

$$(3.7) \quad \begin{aligned} k_a(l) &= \frac{l - \text{mod}(l, K/2)}{K/2}, \\ k_b(l) &= (K/2) + \text{mod}(l, K/2). \end{aligned}$$

Equation (3.6) is then minimized over all k calibration delays, e.g. by using the MINUIT [JR75, MIN] software package of the ROOT framework [BR97, ROO]. Unfortunately, the computing effort is significantly increased compared to the timing probe method, as the minimization has to repeatedly evaluate Equation (3.6). Additionally, the total number of LORs L is scaled by $L = (K/2)^2$ with the total channel number K , which increases the computational effort even further for large systems.

3.3.3 Analytic Time Calibration

If the process of calculating the various LOR time difference histograms is formulated in a more ordered way, the time calibration problem can be reduced to a matrix based equation. This leads to a least squares solution which allows to determine the channel delays δ_k directly from the available LOR time histogram data [MPT⁺09]. The various LOR time differences $t_{i,j}$ are determined similar to the previous methods, by calculating the time differences between all channel pairs (i, j) , resulting in the LOR time difference $t_{i,j}$. In contrast to the iterative algorithm in the previous Section 3.3.2.2, the LORs are now numbered by the two channel indexes $i \in \{0, 1, \dots, K-2\}$ and $j \in \{0, 1, \dots, K-1\}$ instead of the single LOR number l . The total number of LORs L for all possible combinations of the K channels in an unsegmented system is given by $L = K(K-1)/2$. The LOR time differences are calculated from the channel time-stamps by

$$\begin{aligned}
 & t_0 - t_1 & & = & t_{0,1} \\
 & t_0 & -t_2 & & = & t_{0,2} \\
 & & & \vdots & & \\
 (3.8) \quad & t_0 & & -t_{K-1} & = & t_{0,K-1} \\
 & t_1 & -t_2 & & = & t_{1,2} \\
 & t_1 & & -t_3 & = & t_{1,3} \\
 & & & \vdots & & \\
 & & & & & t_{K-2} - t_{K-1} = t_{K-2,K-1}
 \end{aligned}$$

The Equations (3.8) can then be rearranged in matrix notation with the LOR generator matrix \mathbf{A} :

$$(3.9) \quad \mathbf{A} \cdot \mathbf{t} = \mathbf{t}_{\text{LOR}}$$

$$\begin{bmatrix}
 1 & -1 & 0 & 0 & \cdots & 0 & 0 \\
 1 & 0 & -1 & 0 & \cdots & 0 & 0 \\
 \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\
 1 & 0 & 0 & 0 & \cdots & 0 & -1 \\
 0 & 1 & -1 & 0 & \cdots & 0 & 0 \\
 0 & 1 & 0 & -1 & \cdots & 0 & 0 \\
 \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\
 0 & 0 & 0 & 0 & \cdots & 1 & -1
 \end{bmatrix} \cdot \begin{bmatrix} t_0 \\ t_1 \\ \vdots \\ t_{K-1} \end{bmatrix} = \begin{bmatrix} t_{0,1} \\ t_{0,2} \\ \vdots \\ t_{0,K-1} \\ t_{1,2} \\ t_{1,3} \\ \vdots \\ t_{K-2,K-1} \end{bmatrix}$$

The LOR time differences $t_{i,j}$ for all acquired events are then histogrammed and the mean value $\mu_{i,j}$ of each distribution can be extracted, e.g. by fitting a Gaussian function with constant offset to account for the random coincidence background. Once the mean values are determined for the uncalibrated input times, the channels usually will differ, which would again cause the broad time distribution if all LOR data are combined into a global coincidence time histogram.

To correct for this effect, all LOR mean values should be equal, which can be achieved by adding a calibration offset vector Δ to the LOR time difference mean vector μ_{LOR}

$$(3.10) \quad \mu_{\text{LOR cal}} = \mu_{\text{LOR}} + \Delta.$$

In the case of $\Delta = -\mu_{\text{LOR}}$, the calibrated LOR means are simply aligned at zero time difference. The calibration offsets Δ then have to be converted into the channel calibration delays δ , which can be applied to the raw input data, according to Equation (3.4). By using the LOR generator matrix \mathbf{A} from Equation (3.9) both calibration datasets can be linked by

$$(3.11) \quad \mathbf{A}\delta = \Delta.$$

Thereby, the matrix \mathbf{A} is of size $(K(K-1)/2) \times K$, the channel delay vector δ comprises K elements and the LOR offset vector Δ holds data for all $K(K-1)/2$ LORs. Therefore, Equation (3.11) forms an overdetermined linear equation system. As $\text{rank}(\mathbf{A}) = K-1$, no unique solution exists and a least squares solution has to be calculated which minimizes $\|\mathbf{A}\delta - \Delta\|^2$. This can be achieved either with iterative algorithms, like the conjugate gradient method, or directly via the *Moore-Penrose*³ *pseudo-inverse* \mathbf{A}^+ through

$$(3.12) \quad \delta_{\text{min}} = \mathbf{A}^+\Delta.$$

The pseudo-inverse \mathbf{A}^+ can be determined by the *singular value decomposition* (SVD) [GK65, GR70] of the matrix \mathbf{A} into

$$(3.13) \quad \mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T,$$

where both \mathbf{U} and \mathbf{V} are square matrices of size $(K(K-1)/2) \times (K(K-1)/2)$ and $K \times K$, respectively. The singular values of \mathbf{A} are arranged in the diagonal matrix \mathbf{S} of size $(K(K-1)/2) \times K$, with only $K-1$ non-zero diagonal elements due to the rank defect of \mathbf{A} . From these, the matrix \mathbf{S}^+ is obtained by taking the reciprocal of all non-zero singular values. The pseudo-inverse matrix \mathbf{A}^+ is then calculated by

$$(3.14) \quad \mathbf{A}^+ = \mathbf{V}\mathbf{S}^+\mathbf{U}^T.$$

³Named after Eliakim Hastings Moore (1862–1932) [Moo20] and Sir Roger Penrose (*1931) [Pen55]

Unfortunately, determining \mathbf{A}^+ via the SVD method is computing and memory intensive. While the initial matrix \mathbf{A} is only sparsely populated with ± 1 values, the decomposed matrices \mathbf{U} and \mathbf{V} are usually fully populated with floating point numbers [P⁺08]. However for changing channel numbers, the calculation of the pseudo-inverse via the complete SVD method is not necessary if the advantageous structure of \mathbf{A} is considered. Therefore, the first three non-trivial cases are listed in the following for the LOR matrix \mathbf{A}_K and the corresponding transposed pseudo-inverse \mathbf{A}_K^{+T} for increasing channel numbers K :

$$\mathbf{A}_3 = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}, \quad \mathbf{A}_3^{+T} = \begin{bmatrix} 1/3 & -1/3 & 0 \\ 1/3 & 0 & -1/3 \\ 0 & 1/3 & -1/3 \end{bmatrix},$$

$$\mathbf{A}_4 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \quad \mathbf{A}_4^{+T} = \begin{bmatrix} 1/4 & -1/4 & 0 & 0 \\ 1/4 & 0 & -1/4 & 0 \\ 1/4 & 0 & 0 & -1/4 \\ 0 & 1/4 & -1/4 & 0 \\ 0 & 1/4 & 0 & -1/4 \\ 0 & 0 & 1/4 & -1/4 \end{bmatrix},$$

$$\mathbf{A}_5 = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}, \quad \mathbf{A}_5^{+T} = \begin{bmatrix} 1/5 & -1/5 & 0 & 0 & 0 \\ 1/5 & 0 & -1/5 & 0 & 0 \\ 1/5 & 0 & 0 & -1/5 & 0 \\ 1/5 & 0 & 0 & 0 & -1/5 \\ 0 & 1/5 & -1/5 & 0 & 0 \\ 0 & 1/5 & 0 & -1/5 & 0 \\ 0 & 1/5 & 0 & 0 & -1/5 \\ 0 & 0 & 1/5 & -1/5 & 0 \\ 0 & 0 & 1/5 & 0 & -1/5 \\ 0 & 0 & 0 & 1/5 & -1/5 \end{bmatrix}.$$

The pseudo-inverse was determined numerically, using the SciLab [Gom98] scientific software package. For convenience, the floating point solution of the matrix elements is converted to a fractional integer notation. If \mathbf{A}_K and the pseudo-inverse matrix \mathbf{A}_K^+ are now successively evaluated for increasing channel numbers K , the following relation is found

$$(3.15) \quad \mathbf{A}_K^+ = \frac{1}{K} \cdot \mathbf{A}_K^T.$$

Due to the regular structure of \mathbf{A}_K , it can be easily generated dynamically during the calibration process, without the need of storing any matrix in full or sparse format.

Unfortunately, a drawback of the matrix \mathbf{A} , which uses all possible LOR combinations in the scanner system, is that usually the calibration source is not illuminating all realized LORs. Since the two positron annihilation photons are emitted in opposite directions, originating from the radioactive source, LORs which are not intersecting the source are usually not populated with useful data during the calibration run. In fact, the only contributions may arise from random coincidences, scattering interactions in multiple detectors or optical crosstalk between neighboring channels which neither are useful events for the calibration. This problem can be reduced by dividing the scanner into opposing segments, similar to the iterative calibration in Section 3.3.2.2 and evaluating only LORs between the two segments.

3.3.3.1 Segmented LOR Geometry

By dividing the scanner system logically into two opposing segments and accepting only coincidences between the two segments, a major part of the calibration background events due to scattering and crosstalk can be reduced. For this configuration a similar LOR generator matrix \mathbf{B}_{K_a, K_b} of size $K_a K_b \times (K_a + K_b)$ can be formulated to describe the individual time differences defined by the LORs. The segmentation of the scanner system results in a block structure of \mathbf{B}_{K_a, K_b} , which depends on the channel numbers K_a and K_b in both segments. In the following, two examples of \mathbf{B}_{K_a, K_b} and the corresponding transposed pseudo-inverse matrices $\mathbf{B}_{K_a, K_b}^{+T}$ are listed:

$$\mathbf{B}_{2,2} = \begin{bmatrix} 1 & 0 & \text{''} & -1 & 0 \\ 1 & 0 & \text{''} & 0 & -1 \\ \text{---} & \text{''} & \text{---} & \text{---} & \text{---} \\ 0 & 1 & \text{''} & -1 & 0 \\ 0 & 1 & \text{''} & 0 & -1 \end{bmatrix}, \quad \mathbf{B}_{2,2}^{+T} = \begin{bmatrix} 3/8 & -1/8 & \text{''} & -3/8 & 1/8 \\ 3/8 & -1/8 & \text{''} & 1/8 & -3/8 \\ \text{---} & \text{''} & \text{---} & \text{---} & \text{---} \\ -1/8 & 3/8 & \text{''} & -3/8 & 1/8 \\ -1/8 & 3/8 & \text{''} & 1/8 & -3/8 \end{bmatrix},$$

$$\mathbf{B}_{3,2} = \begin{bmatrix} 1 & 0 & 0 & \vdots & -1 & 0 \\ 1 & 0 & 0 & \vdots & 0 & -1 \\ \hline 0 & 1 & 0 & \vdots & -1 & 0 \\ 0 & 1 & 0 & \vdots & 0 & -1 \\ \hline 0 & 0 & 1 & \vdots & -1 & 0 \\ 0 & 0 & 1 & \vdots & 0 & -1 \end{bmatrix}, \quad \mathbf{B}_{3,2}^{+T} = \begin{bmatrix} 2/5 & -1/10 & -1/10 & \vdots & -4/15 & 1/15 \\ 2/5 & -1/10 & -1/10 & \vdots & 1/15 & -4/15 \\ \hline -1/10 & 2/5 & -1/10 & \vdots & -4/15 & 1/15 \\ -1/10 & 2/5 & -1/10 & \vdots & 1/15 & -4/15 \\ \hline -1/10 & -1/10 & 2/5 & \vdots & -4/15 & 1/15 \\ -1/10 & -1/10 & 2/5 & \vdots & 1/15 & -4/15 \end{bmatrix}.$$

The block structure is also preserved for higher channel counts. Therefore, a general description of the pseudo-inverse can be formulated:

(3.16)

$$\mathbf{B}_{2,2}^{+T} = \begin{bmatrix} a & b & \vdots & c & d \\ a & b & \vdots & d & c \\ \hline b & a & \vdots & c & d \\ b & a & \vdots & d & c \end{bmatrix}, \quad \mathbf{B}_{3,2}^{+T} = \begin{bmatrix} a & b & b & \vdots & c & d \\ a & b & b & \vdots & d & c \\ \hline b & a & b & \vdots & c & d \\ b & a & b & \vdots & d & c \\ \hline b & b & a & \vdots & c & d \\ b & b & a & \vdots & d & c \end{bmatrix}, \quad \mathbf{B}_{3,3}^{+T} = \begin{bmatrix} a & b & b & \vdots & c & d & d \\ a & b & b & \vdots & d & c & d \\ a & b & b & \vdots & d & d & c \\ \hline b & a & b & \vdots & c & d & d \\ b & a & b & \vdots & d & c & d \\ b & a & b & \vdots & d & d & c \\ \hline b & b & a & \vdots & c & d & d \\ b & b & a & \vdots & d & c & d \\ b & b & a & \vdots & d & d & c \end{bmatrix}.$$

The individual matrix elements can be determined directly from the matrix dimensions K_a and K_b by

$$(3.17) \quad \begin{aligned} a &= b + \frac{1}{K_b}, \\ b &= -\frac{1}{f} \cdot (K_a^2 + K_a K_b), \\ c &= d - \frac{1}{K_a}, \\ d &= \frac{1}{f} \cdot (K_b^2 + K_a K_b), \\ f &= K_a K_b (K_a + K_b)^2. \end{aligned}$$

Since the regular structure of \mathbf{B}^+ is well determined by the LOR arrangement, the matrix can be generated on the fly by using Equations (3.17). The individual channel delays are then determined, similar to the previous method, according to Equation (3.12) which gives the least-squares solution.

3.3.3.2 Fan-Sum LOR Geometry

Since all previous methods are evaluating individual LORs, a certain number of events is required for each LOR time difference histogram, to be able to determine the mean position of the LOR time difference reliably. For LORs on the periphery of the scanner, which are not directly crossing the calibration source, again, only scattered events are contributing. Thus, it is difficult to extract a proper value for the mean LOR time. Additionally, depending on the overall system size, the measurement time increases with higher channel numbers, since events are distributed among a growing number of LORs in the system. Therefore, a method to increase the statistics for the individual time difference histograms is necessary.

This can be achieved by evaluating the *fan-sum* of multiple LORs, which originate at one detector in the first scanner segment and are targeting all opposite detectors in the second segment, similar to the work in [LGMY01]. Thus, the statistics in the fan-sum time difference histogram is increased and allows a more reliable extraction of the mean value. As the histogram is build from the sum of multiple individual LORs, the time difference peak is broadened due to the superposition of the misaligned LOR data. However, since only the mean value of the time difference peak is extracted, this effect is of minor importance. In contrast, since the histograms includes now more statistics, the peak extraction via fit routines is more precise compared to the calculation of a simple arithmetic mean value.

To treat all detector elements equally, a fan-sum has to be determined for each individual detector, within both scanner segments. The matrix \mathbf{C}_{K_a, K_b} , which defines the fan-sum time differences, is always segmented into four blocks, independent of the channel numbers K_a and K_b in both segments. The size of \mathbf{C}_{K_a, K_b} is given by $(K_a + K_b) \times (K_a + K_b)$:

$$(3.18) \quad \mathbf{C}_{2,3} = \begin{bmatrix} 1 & 0 & \vdots & -1/3 & -1/3 & -1/3 \\ 0 & 1 & \vdots & -1/3 & -1/3 & -1/3 \\ \hline 1/2 & 1/2 & \vdots & -1 & 0 & 0 \\ 1/2 & 1/2 & \vdots & 0 & -1 & 0 \\ 1/2 & 1/2 & \vdots & 0 & 0 & -1 \end{bmatrix},$$

$$(3.19) \quad \mathbf{C}_{K_a, K_b} = \begin{bmatrix} \mathbf{I}_{K_a} & K_b^{-1} \cdot \mathbf{1}_{K_a, K_b} \\ K_a^{-1} \cdot \mathbf{1}_{K_b, K_a} & -\mathbf{I}_{K_b} \end{bmatrix}.$$

When the pseudo-inverse is determined for various channel configurations, a similar structure as in the previous Section 3.3.3.1 can be found:

$$(3.20) \quad \mathbf{C}_{2,3}^{+\Gamma} = \begin{bmatrix} a & b & e & e & e \\ b & a & e & e & e \\ g & g & c & d & d \\ g & g & d & c & d \\ g & g & d & d & c \end{bmatrix},$$

$$(3.21) \quad \mathbf{C}_{K_a, K_b}^{+\Gamma} = \begin{bmatrix} b \cdot \mathbf{1}_{K_a, K_a} + (a - b) \cdot \mathbf{I}_{K_a} & e \cdot \mathbf{1}_{K_a, K_b} \\ g \cdot \mathbf{1}_{K_b, K_a} & d \cdot \mathbf{1}_{K_b, K_b} + (c - d) \cdot \mathbf{I}_{K_b} \end{bmatrix}.$$

The pseudo-inverse matrix elements are thereby given by

$$(3.22) \quad \begin{aligned} a &= 1 + b, \\ b &= -\frac{1}{f} \cdot (K_b^3 + K_a K_b^2 + K_a^2 K_b), \\ c &= d - 1, \\ d &= \frac{1}{f} \cdot (K_a^3 + K_a^2 K_b + K_a K_b^2), \\ e &= -\frac{1}{f} \cdot K_a K_b^2, \\ g &= \frac{1}{f} \cdot K_a^2 K_b, \\ f &= K_a K_b (K_a + K_b)^2. \end{aligned}$$

Similar to the previous cases, the Equations in (3.22) can be used to build the pseudo-inverse matrix on the fly during the evaluation of the equation system. Thus, no additional storage space for the full matrix is required.

3.3.3.3 Simulations and Real Data Results

The performance of the analytical time calibration algorithms was evaluated with Monte-Carlo simulations and real detector data, acquired with the MADPET-II data acquisition described in Section 3.2. For the Monte-Carlo simulations, uniformly distributed ($\sim \mathcal{U}[0, 100]$) channel delays were randomly generated at first. From the channel delays, the mean values of the LOR time differences can be determined by using the LOR generator matrices according to Equation (3.11). The obtained values now correspond to the LOR mean values, which could as well be determined from real acquired data by evaluating the individual LOR time difference histograms. With the corresponding pseudo-inverse of the LOR generator matrix, the calibration values for the channel delays are determined as given in Equation (3.12) and can be subtracted from the initial random channel delays. If the calibrated channel delays are again transformed into LOR time differences, the resulting histogram should ideally have all entries at the zero position, since all channel delays are now supposed to be equal.

Table 3.2 summarizes the simulation results for the three different LOR geometries. Table 3.3 gives the computational complexity and memory requirements for the matrix vector multiplication with the pseudo-inverse and the achieved computing time on an Intel[®] Pentium[®] 4 PC with 3.06 GHz. The total number of simulated channels was 1000 and 500 + 500 for the segmented and fan-sum cases, respectively.

	LOR geometry	Uncalibrated		Calibrated	
		Mean	RMS	Mean	RMS
LOR time difference	A (all)	-1.92	41.53	0.00	0.00
	B (segmented)	-2.62	41.52	0.00	0.00
	C (fan-sum)	-2.62	29.36	0.00	0.00
Channel delays	A (all)	49.78	29.39	49.78	0.00
	B (segmented)	49.78	29.39	49.78	0.00
	C (fan-sum)	49.78	29.39	49.78	0.00

Table 3.2: Simulation of the time calibration methods for the three different LOR geometries.

Table 3.2 shows that all three LOR geometries are equally effective for the time calibration procedure as all reach the zero mean value with zero RMS for the calibrated LOR time differences. Regarding the number of floating point operations, method A is most effective, since most elements of the pseudo-inverse matrix are zero. However, this pseudo-inverse matrix has the most entries of all three methods. Since the online generation of the matrix entries loops over all possible matrix elements, the required computing time increases.

The two remaining methods require more multiply-add operations, since the pseudo-inverse is fully populated, but the matrix size is reduced which reduces the run time as

LOR geometry	Matrix size	Memory requirements ($\delta + \Delta$)	Floating point operations (multiply-add)	Computing time
A (all)	$K^2(K-1)/2$ 499500000	$K + K(K-1)/2$ 500500	$K(K-1)$ 999000	7.7 s
B (segmented)	$K_a K_b (K_a + K_b)$ 250000000	$K_a K_b + K_a + K_b$ 251000	$K_a K_b (K_a + K_b)$ 250000000	3.4 s
C (fan-sum)	$(K_a + K_b)^2$ 1000000	$2(K_a + K_b)$ 2000	$(K_a + K_b)^2$ 1000000	< 0.1 s

Table 3.3: Properties of the time calibration: Computational complexity, memory requirements and computing time for the three different LOR geometries. (500 + 500 channels, Intel® Pentium® 4 PC with 3.06 GHz)

well. The smallest floating point requirements can be achieved with the fan-sum method C, resulting in almost negligible computing time. The memory requirements are determined by the number of channels and LORs/fan-sums in the system. Therefore, also method C is most efficient since the LOR number is minimized.

For the evaluation with real detector data, a dataset from an ^{18}F -FDG flood source measurement, acquired with the existing MADPET-II data acquisition from Section 3.2.2, was used. The measurement setup included all 1152 detector channels, however with about 250 noisy or nonworking channels, which are suppressed by the applied energy cut of $400 \text{ keV} < E < 700 \text{ keV}$. The final number of valid events in the accepted energy range was 431×10^6 . For simplicity, the time calibration procedures are always considering the full channel number of $K = 1152$, which leads to the number of individual LOR histograms listed in Table 3.4. For nonworking channels, the time histogram mean value is assumed to be already perfectly aligned at zero, thus not requiring any additional calibration delay. The time differences are evaluated and histogrammed in the range of -100 ns to $+100 \text{ ns}$.

LOR geometry	Number of LORs	Average LOR entries
A (all)	662976	—
B (segmented)	331776	~ 17
C (fan-sum)	1152	~ 9668

Table 3.4: Number of individual LORs for the three time calibration methods. The total channel number is $K = 1152$ and 431×10^6 events were analyzed in the energy range of $400 \text{ keV} < E < 700 \text{ keV}$.

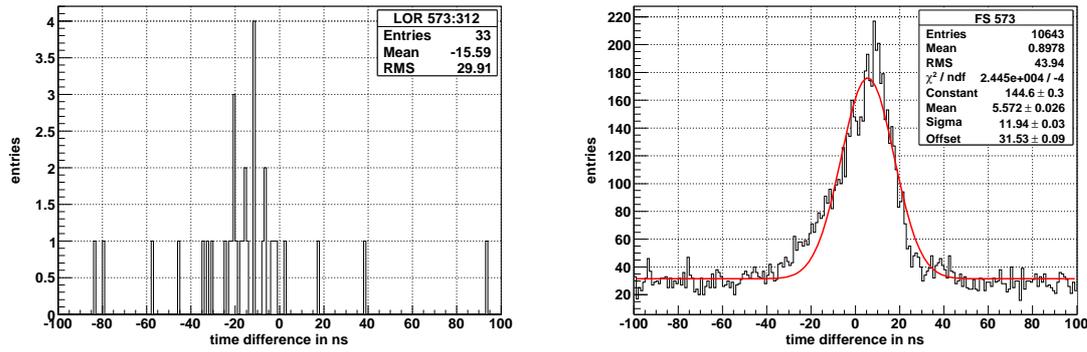


Figure 3.10: Comparison of the LOR time difference histograms for the segmented method **B** (*left*) and the fan-sum method **C** (*right*). The number of total events is 431×10^6 . The fan-sum histogram shows also the Gaussian fit for the first calibration iteration.

As already explained in the previous description of the calibration procedures, method **A** is not a useful LOR geometry choice since many invalid LORs are included. This leads to the high LOR number given in Table 3.4 and therefore this method is not evaluated further.

The logical segmentation of the scanner into two opposing segments in method **B** reduces the LOR number, but still requires a large number of histograms for the time differences. This leads as well to relatively low statistics in some time difference histograms. Thus, the mean value can not be extracted reliably by a Gaussian function fit and the arithmetic mean of all histogram entries has to be used instead.

The fan-sum method **C** requires only one histogram for each detector channel, which leads also to significantly more entries in each histogram. Therefore, the mean value of the time difference peak can be extracted by a fit with a Gaussian function and constant background which achieves better results than the arithmetic mean of all entries. Figure 3.10 compares two time difference histogram examples for method **B** (*left*) and **C** (*right*).

Figure 3.11 shows the improvement in time resolution for the full MADPET-II detector system, obtained with 10 iterations of the fan-sum method. The full width at half maximum (FWHM) of the time resolution improves from $t_{\text{uncal}} = 29.11$ ns FWHM to $t_{\text{cal}} = 16.99$ ns FWHM. The histogram shape is determined by two Gaussian functions for the peak and a constant value for the background arising from random coincidences, resulting in the fit-function

$$(3.23) \quad f(x) = A_1 \cdot e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} + A_2 \cdot e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} + A_3.$$

The FWHM time resolution is then obtained by numerically searching for the two solutions of $f(x_i) = \max(f(x))/2$ and determining the absolute difference $|x_1 - x_2|$.

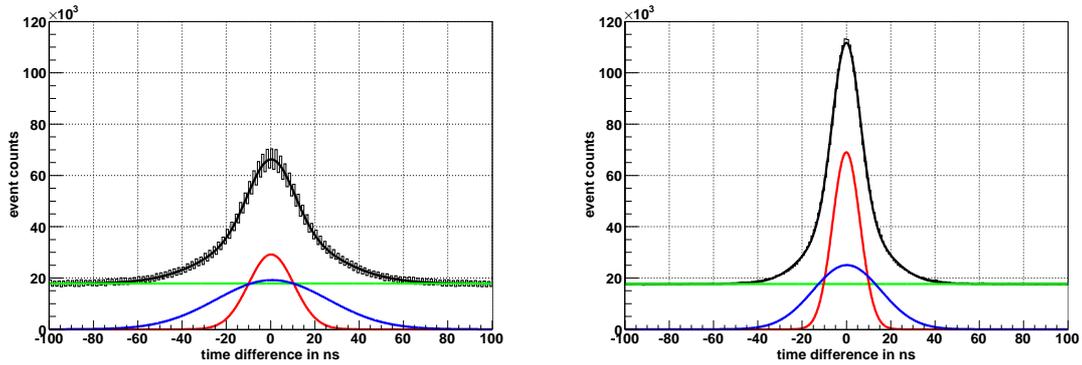


Figure 3.11: Time difference histograms of the full MADPET-II system before (*left*) and after (*right*) time calibration with 10 iterations of the fan-sum method. The time resolution improves from $t_{\text{uncal}} = 29.11$ ns FWHM to $t_{\text{cal}} = 16.99$ ns FWHM. The histogram shape is characterized by two Gaussian functions (*red*, *blue*) and a constant contribution from the random coincidences (*green*).

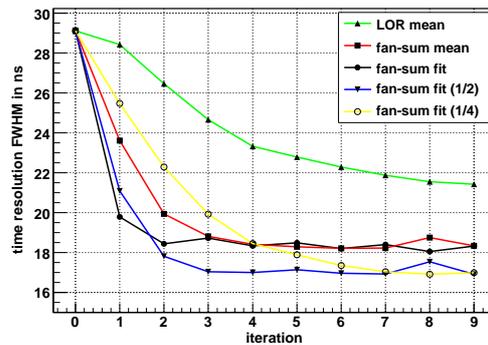


Figure 3.12: Improvement of different time-calibration methods after multiple iterations of the calibration algorithm.

The improvement of the time resolution after multiple calibration iterations is shown in Figure 3.12 for different algorithm variations. Therefore, for each iteration the LOR or fan-sum time difference histograms are evaluated and the calibration values from the previous iteration are used as corrections. The results from each new iteration are then added to the already determined calibration delay values.

Since the determination of the LOR histogram mean values for the segmented geometry suffers from the low statistics, it also shows the worst performance with only slow convergence. The fan-sum approaches with either the arithmetic histogram mean or peak-fit mean values show better convergence and reach a time resolution around 18 ns FWHM. However, if the determined delay values for each individual iteration are considered, the delays do not stabilize but are oscillating between two solutions, which limits the achievable time resolution.

To further improve the result, the determined delay values can be scaled by a damping factor before they are added to the “final” calibration set. By a scaling of 1/2, the overall

time resolution improves to about 17 ns after four iterations. By further increasing the scaling to $1/4$, the algorithm requires more iterations to converge, but the time resolution is not improving any further.

When compared with the timing probe and iterative time calibration from Section 3.3.2.1 and Section 3.3.2.2 which are evaluated in [Tap08], the analytical approach gives comparable results with the iterative time calibration. The direct delay measurement with the timing probe provides however an even further improvement of the time resolution, but requires therefore a full day measurement due to the limited activity in the probe. Ideally, the calibration procedure should combine both methods to achieve the best result. The timing probe measurement therefore determines directly the initial correction values without any ambiguities caused by the overdetermined equation system. The analytical time calibration is then useful to recalibrate the system before the everyday use or during long term measurements and to correct small variations due to environmental changes, like e.g. temperature variations.

Chapter 4

Data Acquisition Based on Sampling ADCs

The analog ASIC based data acquisition system described in the previous chapter implements more or less the standard NIM¹/CAMAC² module based signal flow which is commonly used in laboratory setups. Accordingly, the detector signals are processed first on the analog signal level with discriminators and peak detectors followed by digitization with time- or amplitude-to-digital converters. Since the time critical part of signal detection resides in the analog domain, the speed requirements for the data converters and the digital readout are only defined by the expected event rate, which is in the kHz range for MADPET-II. Once the selected processing logic is implemented into an ASIC, the circuit is fixed and can only be parameterized by predefined inputs of the ASIC.

A different approach could refer to a digital oscilloscope which acquires the complete detector signal waveform upon a trigger condition. The required time and amplitude information must then be extracted from the sampled and quantized signal by applicable postprocessing algorithms. Contrary to the analog processing approach, all signal processing operations are now in the digital domain. This is more demanding regarding the data converter selection, since the analog waveform must be sampled at a sufficiently high sampling rate to retain the time and amplitude information. Once the signal processing is implemented in software or programmable hardware, this concept though provides more flexibility compared to the analog approach. This is similar to developments already established in mobile communications and software defined radio which successively move more and more signal operations, like modulation and filtering, into the digital domain.

¹NIM: Nuclear Instrumentation Module

²CAMAC: Computer Automated Measurement And Control

The task of determining the time and amplitude/energy information from the analog APD detector pulses can be divided into four steps, which have to be addressed by either proper component selection or the implemented processing algorithms:

- Signal sampling,
- Trigger decision,
- Amplitude extraction,
- Time extraction.

The initial step of sampling the analog detector signal determines first requirements for the selection of suitable analog-to-digital converters which then provide a continuous stream of sampled digital data words. The trigger decision process finds valid detector pulses in the data stream and discriminates them from the present background noise. Once individual detector pulses are extracted from the ADC data, they can be independently evaluated for the amplitude and start time of the pulse.

4.1 Signal Sampling

In contrast to radio frequency applications, the signals arising from the LSO scintillator and APD photodetector include no oscillating components like carrier or modulation frequencies. Instead, they can be modeled by a series connection of filter elements which transform the step input of the initial 511 keV photon signal. To simplify the model of chained filter elements, the corresponding transfer functions can be expressed in the Laplace domain, which is linked to the time domain via the Laplace transform (see Appendix C.2). Thus, the time domain convolution of multiple chained transfer functions simplifies to a multiplication in the Laplace domain, which allows to split the overall detector pulse shape into the individual contributions from the elements of the signal path. A schematic overview of the model for the analog signal flow is given in Figure 4.1.

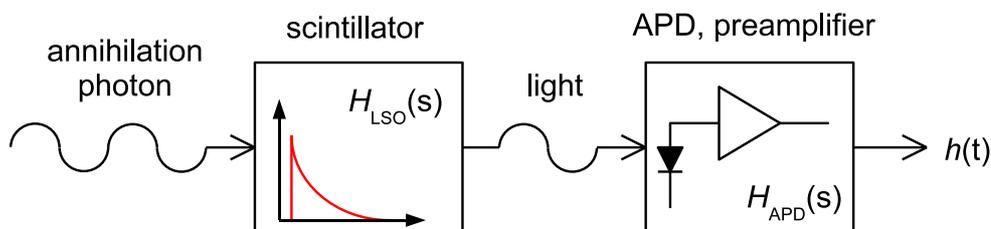


Figure 4.1: Schematic view of the time domain detector pulse shape model $h(t)$, which is defined by the transfer functions in the Laplace domain of the scintillator light decay $H_{\text{LSO}}(s)$ and the APD/preamplifier combination $H_{\text{APD}}(s)$.

4.1.1 Detector Signal Model

At first, the 511 keV photon is converted by the LSO scintillator into visible light. The output light pulse shape of LSO follows an exponential decay of the form $e^{-t/\tau}$, with values of τ in the range of 40 ns to 50 ns [MS92, Lud95]. Therefore, the scintillator element can be described as a high-pass filter with a unit step excitation in the signal chain. This results in the following system transfer function $H_{\text{LSO}}(s)$ in the Laplace domain:

$$(4.1) \quad H_{\text{LSO}}(s) = \mathcal{L}(e^{-t/\tau_{\text{LSO}}} \cdot u(t)) = \frac{s \cdot \tau_{\text{LSO}}}{1 + s \cdot \tau_{\text{LSO}}} \cdot 1/s.$$

The excitation from the detected photon is given by the unit step function $u(t)$ at the filter input, which transits from 0 to 1 at $t = 0$ when the light emission from the scintillator starts. The unit step corresponds to the inverse Laplace transformation \mathcal{L}^{-1} of the expression $1/s$

$$(4.2) \quad u(t) = \mathcal{L}^{-1}(1/s) = \begin{cases} 0 & \text{for } t < 0, \\ 1 & \text{for } t \geq 0. \end{cases}$$

The next element in the signal chain is the APD, followed by the ASIC with the charge sensitive preamplifier and the differential line driver as already shown in Figure 3.4. As the light detection process of the APD shows no significant delay, the impulse response is mainly influenced by the following amplifier and driver circuits. From [Pic01b], the amplifier/driver combination is specified with a pulse rise time of 26 ns for an input capacitance of 15 pF, which corresponds to the typical capacitance of the connected APD. The pulse fall time is set to 1 μ s by the internal preamplifier feedback. To accommodate to the given rise and fall values, the APD readout circuit can be seen as a low-pass/high-pass filter combination, which combines both exponential rise and decay functions in the time domain. Since the signal fall time of the preamplifier is much bigger than the decay time constant of the LSO scintillator, the preamplifier integrates the input signal from the scintillator/APD. This is again modeled by a unit step excitation, with a $1/s$ term in the transfer function.

$$(4.3) \quad H_{\text{APD}}(s) = H_{\text{LP}}(s) \cdot H_{\text{HP}}(s) \cdot 1/s = \underbrace{\frac{1}{1 + s \cdot \tau_{\text{LP}}}}_{\text{low-pass}} \cdot \underbrace{\frac{s \cdot \tau_{\text{HP}}}{1 + s \cdot \tau_{\text{HP}}}}_{\text{high-pass}} \cdot 1/s.$$

The specified pulse rise and fall times in [Pic01b] allow two different interpretations: The given values can correspond directly to the τ values of the low- and high-pass components, which parametrize the rising and falling exponential functions in the time domain.

On the other hand, when seen from the measurement perspective, the signal rise time is usually defined as the transition time from 10% to 90% amplitude. Similarly, the signal fall time or pulse decay time specifies the transition time from 90% to 10% of the maximum amplitude³. This approach requires the τ values to be calculated from the given rise and fall times. This can be achieved for example by evaluating the time domain expression of the signal pulse $h_{\text{APD}}(t) = \mathcal{L}^{-1}(H_{\text{APD}}(s))$ numerically.

With the assumed values for $t_r = 26$ ns and $t_f = 1000$ ns the two time constants can be determined to $\tau_{\text{LP}} \approx 15$ ns and $\tau_{\text{HP}} \approx 454$ ns. When the final pulse shape in Figure 4.4 is compared with measured pulse shapes from the detector elements in Figure 4.5, it becomes obvious that the given values are not referring to the 10% and 90% time definition, but giving directly the values for $\tau_{\text{LP}} = 26$ ns and $\tau_{\text{HP}} = 1000$ ns. Therefore, the 10% and 90% rise and fall time values are $t_r \approx 45$ ns and $t_f \approx 2199$ ns, respectively.

The final pulse shape at the preamplifier/driver output is described by the time domain convolution of the scintillator and the APD circuit response function, which is simplified to a multiplication in the Laplace domain (4.4). Equation (4.5) gives the expression for the time domain representation of the convoluted signal, which is simply a weighted sum of three exponentials as given in (4.6).

$$(4.4) \quad \begin{aligned} H(s) &= H_{\text{LSO}}(s) \cdot H_{\text{APD}}(s) \\ &= \frac{\tau_{\text{LSO}} \cdot \tau_{\text{HP}}}{(1 + s \cdot \tau_{\text{LSO}}) \cdot (1 + s \cdot \tau_{\text{LP}}) \cdot (1 + s \cdot \tau_{\text{HP}})}. \end{aligned}$$

$$(4.5) \quad \begin{aligned} h(t) = \mathcal{L}^{-1}(H(s)) &= \frac{\tau_{\text{LSO}} \cdot \tau_{\text{HP}}^2 \cdot e^{-t/\tau_{\text{HP}}}}{\tau_{\text{HP}}^2 + (-\tau_{\text{LP}} - \tau_{\text{LSO}}) \cdot \tau_{\text{HP}} + \tau_{\text{LSO}} \cdot \tau_{\text{LP}}} \\ &- \frac{\tau_{\text{LSO}} \cdot \tau_{\text{LP}} \cdot \tau_{\text{HP}} \cdot e^{-t/\tau_{\text{LP}}}}{(\tau_{\text{LP}} - \tau_{\text{LSO}}) \cdot \tau_{\text{HP}} - \tau_{\text{LP}}^2 + \tau_{\text{LSO}} \cdot \tau_{\text{LP}}} \\ &+ \frac{\tau_{\text{LSO}}^2 \cdot \tau_{\text{HP}} \cdot e^{-t/\tau_{\text{LSO}}}}{(\tau_{\text{LP}} - \tau_{\text{LSO}}) \cdot \tau_{\text{HP}} - \tau_{\text{LSO}} \cdot \tau_{\text{LP}} + \tau_{\text{LSO}}^2}. \end{aligned}$$

$$(4.6) \quad h(t) = \alpha \cdot e^{-t/\tau_{\text{HP}}} + \beta \cdot e^{-t/\tau_{\text{LP}}} + \gamma \cdot e^{-t/\tau_{\text{LSO}}}.$$

Figure 4.2 shows the pulse shapes of $h_{\text{LSO}}(t)$ and $h_{\text{APD}}(t)$ for values of $\tau_{\text{LSO}} = 41$ ns, $\tau_{\text{LP}} = 26$ ns and $\tau_{\text{HP}} = 1000$ ns. The convoluted pulse shape $h(t)$ is shown in Figure 4.3. A comparison of the derived pulse model with “real” acquired data from the detector is given in Figures 4.4 and 4.5. Figure 4.4 shows the two pulse shapes for the different rise/fall time parameterizations. Compared to the measured pulse shapes, it can be seen

³Rise and fall time are defined in a standard from the American National Standards Institute, Inc. [ATI01]

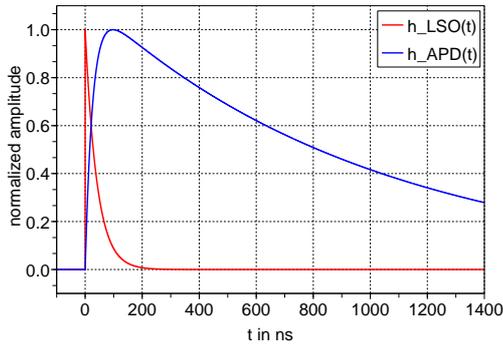


Figure 4.2: Normalized pulse shapes of $h_{\text{LSO}}(t)$ (red) and $h_{\text{APD}}(t)$ (blue).

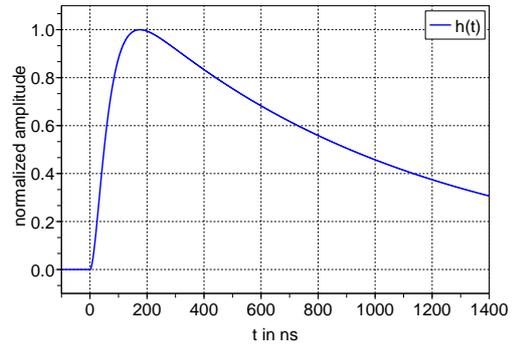


Figure 4.3: Convolved pulse shape $h(t) = h_{\text{LSO}}(t) * h_{\text{APD}}(t)$.

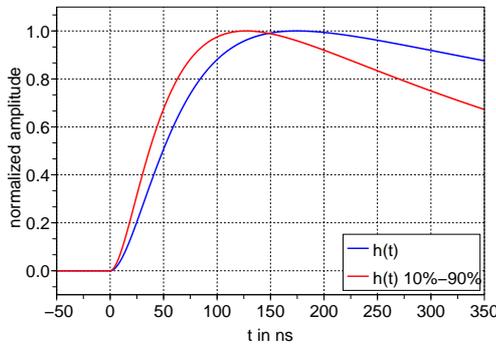


Figure 4.4: Magnified rising edge of $h(t)$ for 10%–90% rise/fall parameterization (red) and τ parameterization (blue).

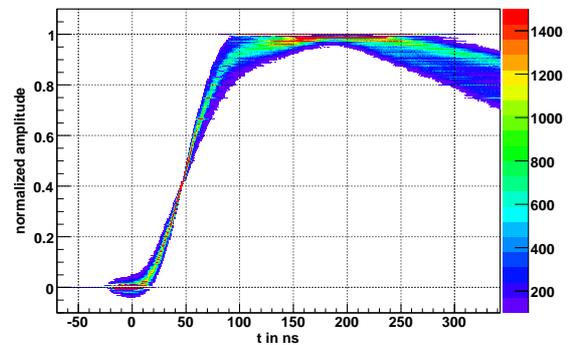


Figure 4.5: Normalized and time aligned detector pulses, acquired with a sampling ADC.

that the given values from the preamplifier specification correspond directly to the τ values of the high-pass and low-pass components.

4.1.2 ADC Requirements

The main component of the data acquisition system is the analog-to-digital converter (ADC) which samples the incoming analog signal and converts it to digital data values. This process inevitably introduces errors to the signal, since it involves a quantization of the initial analog pulse in time as well as in amplitude. The time quantization is related to the ADC sampling frequency f_s , whereas the amplitude quantization depends on the ADC resolution. Generally, the quantization error is reduced with higher sampling rates and a higher ADC resolution. However, the “ideal” ADC requirements have to be realized with “real” hardware components, which requires a trade-off between both the ADC properties and also the component price.

A lower constraint on the sampling frequency is usually applied by the Nyquist-Shannon sampling theorem, which states that an analog signal has to be sampled at least at twice the frequency of its highest frequency component (4.7). This allows a successful reconstruction of the initial analog waveform from the sampled data, without the introduction of aliasing errors (4.8).

$$(4.7) \quad h[n] = h(n \cdot t_s) \quad \text{for} \quad 1/t_s = f_s \stackrel{!}{>} 2 \cdot f_{\max}$$

$$(4.8) \quad h(t) = \sum_{n=-\infty}^{\infty} h[n] \cdot \frac{\sin\left(\frac{t-n \cdot t_s}{t_s}\right)}{\frac{t-n \cdot t_s}{t_s}} = \sum_{n=-\infty}^{\infty} h[n] \cdot \text{sinc}\left(\frac{t-n \cdot t_s}{t_s}\right).$$

Continuous time signals are thereby denoted with parentheses () whereas sampled, discrete time signals are written with brackets []. The frequency spectrum of continuous time signals can be expressed by the Fourier-transform

$$(4.9) \quad H(\omega) = \mathcal{F}(h(t)) = \int_{-\infty}^{\infty} h(t) \cdot e^{-j\omega t} dt.$$

To determine the frequency components of $h(t)$, also the amplitude of the Laplace transformation $|H(s)|$ can be evaluated along the imaginary axis. Since $s = \sigma + j\omega$, this corresponds to the Fourier transformation of $h(t)$ and will give the frequency response. Figure 4.6 shows the frequency response of $h(t)$. The scaling on the frequency axis can be derived by evaluating a sine function with known frequency in the Laplace domain. For $f(t) = \sin(2\pi t)$, which describes a 1 GHz sine wave if t is in nanoseconds, $F(s) = \mathcal{F}(f(t))$ has two poles at $\pm j2\pi$. Therefore, 2π on the ω axis corresponds to a frequency of 1 GHz.

4.1.3 Effects due to Sampling

4.1.3.1 Aliasing

Sampling the signal $h(t)$ at a sampling rate of f_s causes a repetition of the original frequency spectrum $H(\omega)$ at multiples of the sampling rate. This is similar to mirroring $H(\omega)$ at multiples of $f_s/2$. Additionally, the repeated spectra of the sampled signal are scaled by a factor of $1/f_s$. Since the repeated spectra are overlapping, this introduces aliasing errors.

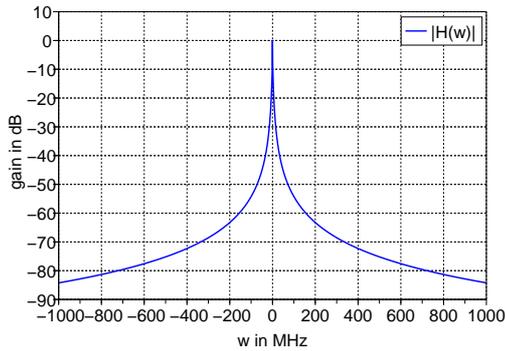


Figure 4.6: Normalized frequency response $H(\omega) = \mathcal{F}(h(t)) = |H(s)|_{s=j\omega}$.

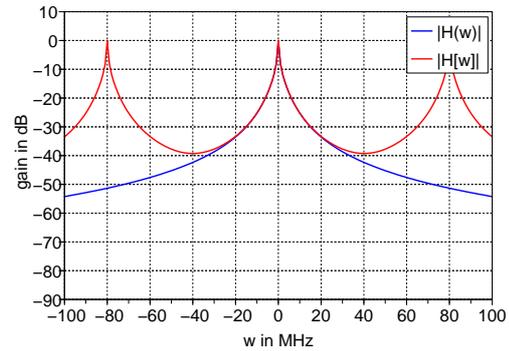


Figure 4.7: Normalized frequency response $H[\omega]$ of $h(t)$ when sampled at 80 MHz (red) compared to the spectrum $H(\omega)$ of the continuous signal (blue).

Figure 4.7 shows this effect for the spectrum $H[\omega] = \mathcal{F}(h[t])$ of a sampled signal in comparison to the spectrum $H(\omega) = \mathcal{F}(h(t))$ of the continuous signal. Given that the continuous waveform is sampled at $f_s = 80$ MHz, frequency components above 40 MHz are mirrored. This causes the deviation between the sampled and continuous spectrum in the range of 20–40 MHz.

Since all poles of $H(s)$ are real valued, the major frequency component is the DC offset at 0 MHz. As the optimal signal reconstruction without aliasing, according to (4.8), is not necessarily required to extract the interesting time and amplitude information, the optimal ADC sampling rate will be determined mainly by the extraction algorithms. However, from Figure 4.6 it can be assumed, that ADCs with a cost effective sampling frequency in the 80 MHz range may be suitable for the specified signal properties, since the aliased frequency contributions are already suppressed by -40 dB.

4.1.3.2 Quantization

Due to the limited amplitude resolution of the ADC, the amplitude of the continuous signal is quantized into discrete steps. For an ADC resolution of N bits and a maximum input amplitude A_{\max} , the smallest resolvable signal step is therefore $A_{\max}/2^N$. This also determines how precise the pulse amplitude can be resolved, which directly contributes to the achievable energy resolution of the detector. Further on, the analog signal amplification and the ADC circuit define what dynamic range of the provided ADC resolution is usable for the expected input signal. Again, the use of adequate processing algorithms may allow to reduce the ADC resolution, which will lower the final price for the data acquisition system.

The limited sampling rate introduces also a quantization in time, which causes the already mentioned aliasing effects in the frequency domain. In addition, since the occurrence of detector pulses is not correlated with the ADC sampling frequency, the pulses

are sampled with varying time delays/offsets respectively to the fixed ADC clock. This is due to the fact that the initial radioactive decay—which eventually results in a digitized detector pulse—is a random process. Therefore, each analog detector pulse is sampled with a certain time offset ϕ relative to the ADC clock, which can be described by expanding the sampling equation (4.7) to

$$(4.10) \quad h[n] = h(n \cdot t_s + \phi) \quad \text{for } \phi \in [0, t_s[.$$

To be able to achieve a precise time information for the detector pulse, the time reconstruction has to determine the offset value ϕ , based on the sampled $h[n]$ data points. Otherwise, the time resolution would be limited by the ADC sampling interval t_s , which can not be adjusted to arbitrarily small values. However, compared to the time properties of the radioactive decay, usual sampling rates in the MHz range are sufficiently small to assume constant decay rates between consecutive sampling points. Therefore, since the ADC clock is independent to the occurrence of the detector pulses, the ϕ values are uniformly distributed over the full sampling interval range:

$$(4.11) \quad \phi \sim \mathcal{U}[0, t_s[.$$

4.2 Trigger Decision

Once the analog signal waveform is digitized, all following processing steps operate on the digital data stream $x[n]$ ⁴. As the ADC provides a new sampled data word each $1/f_s$ time interval, the data also have to be processed with this sampling rate. Since the actual event rate per channel is only around 1–10 kHz, the trigger decision allows to reduce the mean data rate for the following processing elements if only a fraction of the incoming data samples are processed for every detected event. This allows also to share the time and amplitude extraction circuit between several input channels to optimize resource usage for the implementation in programmable logic or an ASIC.

To discriminate a valid detector pulse from the background noise, a decision based on the pulse amplitude is usually implemented with CAMAC/NIM based leading edge discriminators. Since the amplitude is correlated with the energy of the detected particle, an amplitude threshold allows also to suppress events from compton scattered photons with low signal amplitudes which can usually not be used for the PET image reconstruction. Once the signal exceeds the defined threshold value A_{thr} , the transition of the trigger output $T[n]$ from $0 \rightarrow 1$ indicates that the current data should be processed further, otherwise the incoming ADC data samples are ignored after the trigger system.

⁴As the ADC is also providing sampled data when no APD/preamplifier pulse is present, a different variable $x[n]$ denotes the sampled data words in contrast to the APD/preamplifier pulse shape $h[n]$.

$$(4.12) \quad T[n+1] = \begin{cases} 0 & \text{for } x[n] \leq A_{\text{thr}}, \\ 1 & \text{for } x[n] > A_{\text{thr}}. \end{cases}$$

However, due to the small APD signal in the range of 10–50 mV which is amplified by the charge sensitive preamplifier circuit, the whole detector front-end is also sensitive to noise from external electromagnetic interference (EMI). Though this influence can be reduced by a proper shielding of the detector setup, there may still remain EMI components which are amplified into the trigger threshold range and can cause false event triggers. Therefore, a more robust trigger decision is desirable.

4.2.1 Signal Averaging

A simple approach to reduce the background noise for the trigger decision is a preceding averaging of the input signal, followed by the threshold comparison of the averaged value $\bar{x}[n]$. Depending on the number of signal samples N_{avg} used in the average calculation, the standard deviation of the averaged signal $\bar{x}[n]$ can be reduced by a factor of $1/\sqrt{N_{\text{avg}}}$ compared to the original signal [Lyo04]. The averaged value is given by

$$(4.13) \quad \bar{x}[n] = \frac{1}{N_{\text{avg}}} \cdot \sum_{i=1}^{N_{\text{avg}}} x[n-i].$$

The averaging has to operate continuously at the sampling rate of the ADCs to generate an averaged data stream for the threshold decision and can be implemented either as a finite impulse response (FIR) or a recursive infinite impulse response (IIR) filter. The FIR implementation directly maps (4.13) into a multiply-accumulate structure as shown in the top-left of Figure 4.8 with the transfer function in the z -domain⁵

$$(4.14) \quad H_{\text{FIR}}(z) = \frac{1}{N_{\text{avg}}} \cdot \sum_{i=0}^{N_{\text{avg}}-1} z^{-i}.$$

To avoid the delayless multi-term summation, the filter structure can be transposed, which reverses the data flow without affecting the transfer function and thus placing the delay elements between the two-term adders. This structure allows a higher clock speed when implemented in hardware because of the pipelined adders.

⁵The z -transform of discrete time signals with $z = r \cdot e^{-j\omega}$ corresponds to the Laplace transform of continuous time signals whereas $s = \sigma + j\omega$. In z -domain transfer functions the z^{-1} operator can be interpreted as a unit time delay. This allows a direct mapping of the transfer function into a schematic of the filter structure. A more detailed overview of the z -transform is given in Appendix C.3.

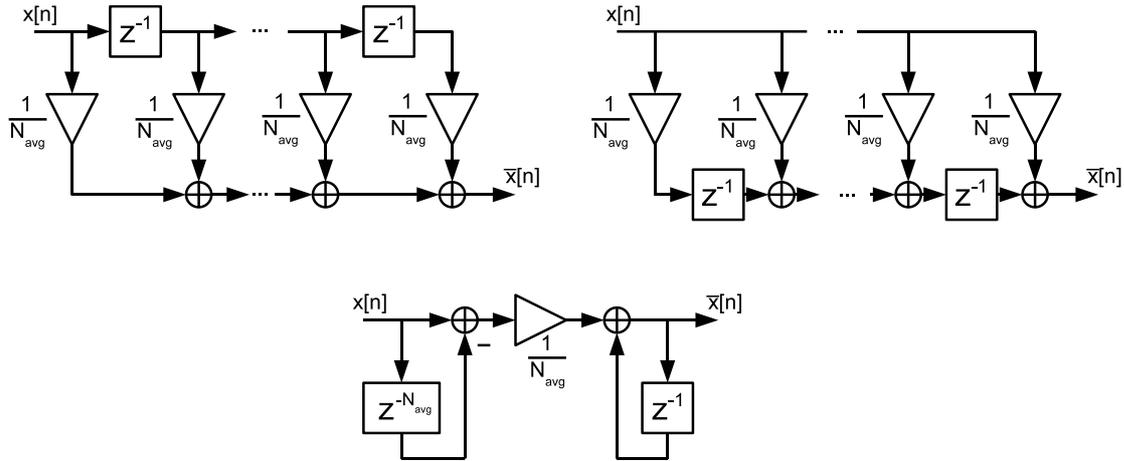


Figure 4.8: Implementation alternatives for the average filter. FIR direct form (*top-left*), FIR direct form II (transposed) (*top-right*), recursive implementation (RRS) (*bottom*).

A recursive average implementation allows to further reduce the number of computational elements to just two adders and delay elements as shown on the bottom of Figure 4.8. This realization is also called a *recursive-running-sum* (RRS) filter. The FIR transfer function in (4.14) can therefore be rewritten in the form of an IIR filter with the zeros in the numerator and the pole from the feedback in the denominator:

$$(4.15) \quad H_{RRS}(z) = \frac{1}{N_{avg}} \cdot \frac{1 - z^{-N_{avg}}}{1 - z^{-1}}.$$

Due to the feedback path, IIR filters can potentially become unstable which occurs for z-domain transfer functions that include poles outside the unit circle ($|z_{\infty}| > 1$). For the recursive average filter, stability is guaranteed due to the pole at $z = 1$ which is canceled by the corresponding zeros in the numerator.

The frequency response of the average filter can be determined by evaluating the magnitude of the transfer function $|H(z)|_{z=e^{j\omega}}$ along the unit circle for $z = e^{j\omega}$. Figure 4.9 shows the frequency response for different average window sizes N_{avg} . By increasing N_{avg} , the cut-off frequency of the low-pass type response can be reduced.

If the window size N_{avg} is chosen to be a power of 2, the $1/N_{avg}$ division can be easily realized by a binary shift operation for integer data. Since this corresponds to a simple remapping of bit lines when implemented in programmable logic, no additional computing time is required for this operation.

A remaining limitation of the recursive average implementation is that it requires at least N_{avg} delay elements, respectively memory locations. Thus, large window sizes become very resource demanding when implemented in hardware. To remove the $z^{-N_{avg}}$ delay chain, it can be approximated for example by an additional feedback loop which subtracts the current average output value. To avoid the creation of a delayless loop, the

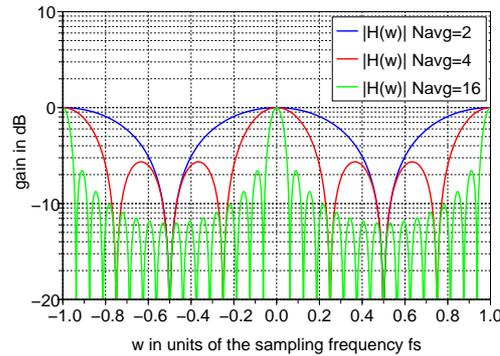


Figure 4.9: Frequency response of the average filter for different window sizes N_{avg} . The x-axis is scaled in units of the sampling frequency f_s .

feedback is taken from the delayed output value in the integration register. The resulting implementation is shown on the left side of Figure 4.10 and the transfer function is given by

$$(4.16) \quad H_{\text{IIR}1}(z) = \frac{1}{N_{\text{avg}}} \cdot \frac{1}{1 - \left(z^{-1} - \frac{1}{N_{\text{avg}}} \cdot z^{-1} \right)}.$$

If a registered output of the filter is required, it can be realized by the integrator feedback register which causes $\bar{x}_2[n]$ to become the filter output. Since the additional output delay only adds a zero in the center of the unit circle, the shape of the frequency response is not changed. The resulting transfer function is given by

$$(4.17) \quad H_{\text{IIR}1}^{(2)}(z) = \frac{1}{N_{\text{avg}}} \cdot \frac{z^{-1}}{1 - \left(z^{-1} - \frac{1}{N_{\text{avg}}} \cdot z^{-1} \right)}.$$

The last remaining bottleneck in the circuit are the cascaded adders which sum up the input for the accumulation register. Since both additions are required within every clock cycle, the highest possible clock frequency is limited by the carry path through both adders. Although this can be addressed by more sophisticated adder implementations, the most resource efficient solution for an FPGA based system is an additional register/delay element between both adders. This gives the structure shown on the right side of Figure 4.10 with the following transfer functions

$$(4.18) \quad H_{\text{IIR}2}(z) = \frac{1}{N_{\text{avg}}} \cdot \frac{z^{-1}}{1 - \left(z^{-1} - \frac{1}{N_{\text{avg}}} \cdot z^{-2} \right)},$$

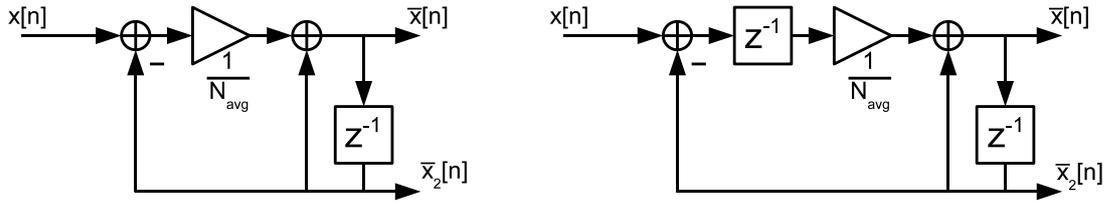


Figure 4.10: Optimized trigger average filters with eliminated $z^{-N_{\text{avg}}}$ delay chain (left) and fully pipelined implementation (right).

$$(4.19) \quad H_{\text{IIR}2}^{(2)}(z) = \frac{1}{N_{\text{avg}}} \cdot \frac{z^{-2}}{1 - \left(z^{-1} - \frac{1}{N_{\text{avg}}} \cdot z^{-2} \right)}.$$

Figure 4.11 shows the frequency response of both implementations, compared with the accurate average implementation for $N_{\text{avg}} = 4$ from Figure 4.9. Both IIR implementations show a smaller passband than the RRS average filter and a smooth stop band with no ripple. In addition to the reduced resource usage, the uniform attenuation of higher frequency components is also beneficial.

The purpose of the data averaging is the suppression of background noise for the trigger threshold decision. Once the averaged data stream is exceeding the trigger threshold, the unfiltered ADC data can be further processed in the following processing stages. The pipelined design with register stages after every adder component enables the operation at high ADC sampling rates for real time data processing. The low resource requirements are also advantageous for multi-channel implementations.

It should be remarked, that the actual trigger threshold value does not correspond directly to the pulse amplitude, due to the preceding average filter of the input data. Therefore, the threshold cannot achieve a sharp amplitude cut, but results in a “smeared” transition region at small amplitude values, as already indicated in Section 3.3.1. Since the real amplitude value of the signal is precisely evaluated in a subsequent processing step, this effect has no negative influence.

4.3 Amplitude Detection

Once a valid signal from the scintillation detector is identified by the trigger logic, the parameters of the sampled pulse must be extracted. At first, the signal amplitude is derived, since it is also required for the following time extraction. However, due to noise and the time and amplitude quantization in the ADC, the “real” signal amplitude may not be directly accessible from the sampled data stream. Additionally, the signal amplitude usually includes a baseline offset, which has to be subtracted from the sampled

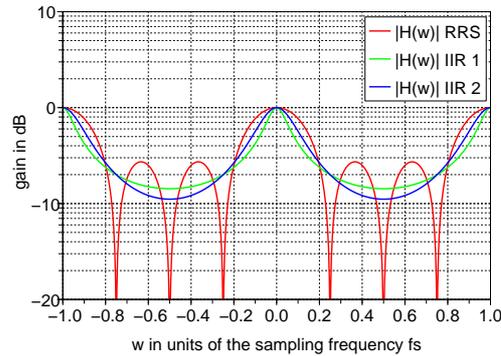


Figure 4.11: Frequency response of the optimized trigger average filters compared to the accurate RRS average filter for $N_{\text{avg}} = 4$. The x-axis is scaled in units of the sampling frequency f_s .

data values. Since the detected signals arise from the radioactive tracer decay, they appear randomly in time which may also lead to pile-up of multiple signals. This distorts the sampled analog data values and has to be considered during amplitude detection.

4.3.1 Pile-Up

As the trigger decision does not yet provide a sample accurate time information for any possible pulse amplitude, the pulse has to be evaluated over a certain time interval to reliably cover the peak of the signal. However, due to the random occurrence of the detector pulses, also *pile-up* events are possible: Due to the long decay tail of the detector signals (with $t_f \approx 2197$ ns for the preamplifier output) there is some probability for new pulses to occur while the previous pulse has not yet decayed. Since the new pulse is added to the decaying signal, the detected peak value of the second signal is overestimated.

The fraction of events which potentially are distorted by pile-up can be estimated, based on the trigger rate per channel. The detection of the radioactive positron decay is therefore described by a Poisson process [Kno99], which gives an expression for the probability of the occurrence of x pulses within a certain time interval t if the pulses are occurring at an average event rate r

$$(4.20) \quad P(x) = \frac{(r \cdot t)^x \cdot e^{-r \cdot t}}{x!}.$$

The probability for a certain time interval without any pulse (which is confined by a single pulse at the beginning and a single pulse at the end) is then given by

$$(4.21) \quad P_2(t) = r \cdot P(0) = r \cdot e^{-r \cdot t}.$$

Pile-up of two pulses then occurs if the interval Δt between the two pulses is smaller than the actual width of the pulse. The probability for this case is given by

$$(4.22) \quad P_3(\Delta t) = \int_0^{\Delta t} P_2(t) dt = \int_0^{\Delta t} r \cdot e^{-r \cdot t} dt = 1 - e^{-r \cdot \Delta t}.$$

With the given values for the expected average trigger rate $r = 10000 \text{ s}^{-1}$ and a pulse width of $\Delta t \approx 3000 \text{ ns}$, the probability for the pile-up of two events can be calculated to $P_3(\Delta t) \approx 0.03$. This simplified model considers only the case when a single second pulse occurs within the decay time of the first one. For higher counting rates, there will also be non negligible contributions of multi-pulse pile-up cases which would also have to be included in the calculation and further increase the pile-up probability.

However, depending on the time interval which is required to detect the signal peak, two different pile-up cases can happen. If the second pulse occurs during the first peak detection time, the maximum value of the first pulse is falsified and the second pulse is not registered at all as an independent event. On the other hand, if the second pulse occurs after the peak value of the first pulse is processed successfully, the decay tail only biases the second pulse and the first pulse is detected successfully. As the pulse maximum is reached after about 200 ns, the probability that both pulses are merged into one peak value is $P_3(200 \text{ ns}) \approx 0.002$. Accordingly, at least one of the two piled-up signal pulses is detected correctly in the remaining events with a probability of $P_3(\Delta t) - P_3(200 \text{ ns}) = 0.028$.

4.3.2 Signal Conditioning

4.3.2.1 Baseline Correction

To avoid offset distortions of the pulse amplitude value, some information about the baseline of the current pulse is required. In the simplest case, the baseline is a more or less constant offset of the analog signal path which is determined once and can then be subtracted from the incoming data stream. Since this value is assumed to be stable during data taking it is called *pedestal value* or only *pedestal*. The pedestal values are usually determined during a dedicated calibration run and are applied during the following data

acquisition. Since the offset values are not completely static but may still vary with temperature and other environmental changes, the calibration procedure has to be repeated from time to time to derive a new set of pedestals.

Although the constant pedestal values can be easily implemented with configuration registers, they are not suitable to compensate for example effects arising from pulse pile-up. To compensate these effects, an online calculation which continuously evaluates the preceding signal history is required. A simple implementation may reuse the average calculation from the trigger decision to determine an average value for the baseline before the rise of the signal pulse. This can correct for slow baseline variations but is however not optimal in the pile-up case, when the baseline is determined by the faster decay of a preceding pulse tail.

To avoid the distortion of the calculated baseline value by pulse data with high amplitudes, the average calculation may only include data values which do not exceed a certain distance from the current baseline value. These correction schemes are for example implemented in the ALTRO⁶ readout ASIC [ALT03] for the ALICE⁷ experiment at the Large Hadron Collider (LHC) at CERN⁸.

4.3.2.2 Tail Cancellation

To reduce the risk of signal pile-up, the long signal decay tail may be compensated with an appropriate filter operation based on the signal model $H(s)$ or $h(t)$ which were already derived in Equation (4.4) and (4.6) in Section 4.1.1

$$H(s) = \mathcal{L}(h(t)) = \frac{\tau_{\text{LSO}} \cdot \tau_{\text{HP}}}{(1 + s \cdot \tau_{\text{LSO}}) \cdot (1 + s \cdot \tau_{\text{LP}}) \cdot (1 + s \cdot \tau_{\text{HP}})},$$

$$h(t) = \alpha \cdot e^{-t/\tau_{\text{HP}}} + \beta \cdot e^{-t/\tau_{\text{LP}}} + \gamma \cdot e^{-t/\tau_{\text{LSO}}}.$$

The corresponding transfer function $H(z)$ in the discrete z -domain can be derived from the continuous Laplace-domain representation $H(s)$ in Equation (4.4) by the bilinear transform method. This is a standard approach for the analytical design of IIR filters, which allows to approximate a continuous prototype transfer function with a discrete filter representation. Therefore, the Laplace parameter s in $H(s)$ is replaced by the expression

$$(4.23) \quad s = \frac{2}{t_s} \cdot \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right),$$

⁶ALTRO: ALice Tpc Read Out

⁷ALICE: A Large Ion Collider Experiment

⁸CERN: Conseil Européen pour la Recherche Nucléaire, now *European Organization for Nuclear Research*

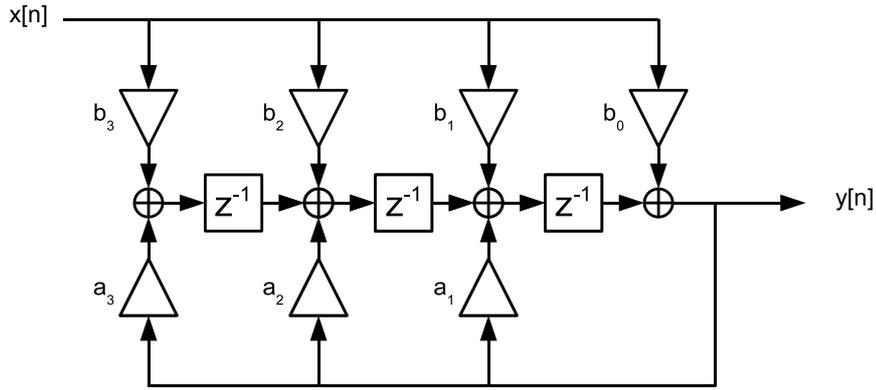


Figure 4.12: IIR filter structure to generate the sampled pulse shape of $H(z)$.

which requires information about the sampling data rate $f_s = 1/t_s$. The resulting z -domain transfer function is then of the form

$$(4.24) \quad H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3}}{1 - (a_1z^{-1} + a_2z^{-2} + a_3z^{-3})}.$$

Again, the numerator of $H(z)$ parametrizes the forward path of an IIR filter, whereas the denominator determines the poles of the recursive feedback path. Once $H(z)$ is excited by a single delta pulse, it reproduces the desired pulse shape of $H(s)$ at the sampling points, spaced by t_s . Figure 4.12 shows the structure of the resulting IIR filter.

To remove the long pulse decay tail, the sampled data which are described by $H(z)$ are fed through a tail cancellation filter $H_{\text{tcf}}(z)$ which then produces a shorter pulse at the output. The transfer function for the tail cancellation filter can be derived by the following formula

$$(4.25) \quad H_{\text{tcf}}(z) = \frac{H_2(z)}{H(z)},$$

where $H_2(z)$ determines the resulting pulse shape after the tail cancellation filter. If the output $h_2[n] = \mathcal{Z}^{-1}(H_2(z))$ is idealized to a simple delta-pulse $\delta[n]$ (with $\mathcal{Z}(\delta[n]) = 1$), $H_{\text{tcf}}(z)$ becomes just the inverse of $H(z)$. The resulting weight of the delta pulse output, after the tail cancellation filter is then proportional to the amplitude of the initial input pulse $H(z)$.

Since $H_{\text{tcf}}(z)$ is an IIR filter, the stability of the final filter system has to be considered in the calculation process, including number rounding and quantization effects, to guarantee the required behavior. Ideally, the tail cancellation filter is operating on the continuously sampled data stream, which however requires a dedicated processing stage for

each input channel. This approach is also used in the ALTRO ASIC [ALT03], but may not be easily applicable to multichannel applications using programmable logic devices, due to the high resource requirements.

4.3.3 Amplitude Measurement

4.3.3.1 Peak-Hold

The simplest approach to extract the maximum value x_{\max} from the sampled pulse data is the digital implementation of a peak-hold circuit, similar to the analog peak detection circuit which is also realized in the MADPET-II analog processing ASIC [Spa06]. By comparing each incoming sample word with the currently determined maximum value, the overall peak value is determined successively by

$$(4.26) \quad x_{\max}[n+1] = \begin{cases} x[n] & \text{for } x[n] > x_{\max}[n], \\ x_{\max}[n] & \text{for } x[n] \leq x_{\max}[n]. \end{cases}$$

After all N samples are processed, $x_{\max}[N]$ holds the biggest value which occurred in the analyzed data. Since the processing requires only a comparison operation and a storage register for the x_{\max} value, the hardware implementation can be very efficient as shown in Figure 4.13.

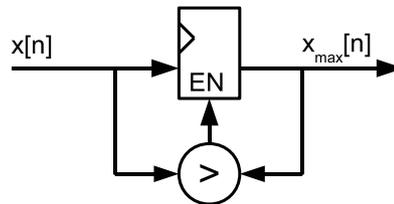


Figure 4.13: Peak-hold circuit implementation.

As described in Section 4.1.3.2, the analog signal from the detector originates from a radioactive decay and the occurrence in time is therefore randomly distributed. Since the ADC sampling frequency is fixed, the analog pulse is then usually sampled with a varying time shift depending on the relative time position of the decay event compared to the ADC sampling points. As the time shift also affects the position of the pulse maximum, the simple peak-hold approach underestimates the real maximum value in most of the cases which is illustrated in Figure 4.14.

The actual error of the extracted peak value is influenced by the pulse shape, as well as the ADC sampling frequency and resolution. By evaluating the analog pulse shape $h(t)$ around the peak position the introduced distortion can be quantified. The position of

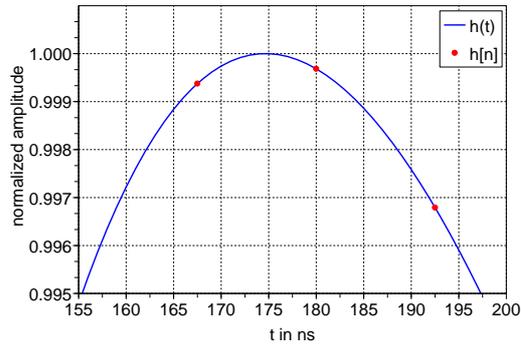


Figure 4.14: Underestimation of the maximum value when derived from sampled data $h[n]$, compared to the continuous pulse $h(t)$.

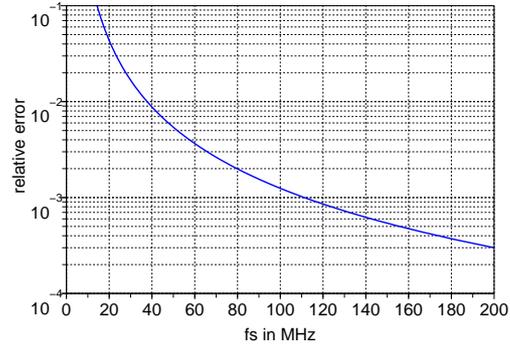


Figure 4.15: Relative maximum error versus sampling frequency for the peak-hold method.

the real pulse peak can be calculated from the first derivative of $h(t)$ by solving numerically $h'(t_{\text{peak}}) = 0$. For the given pulse parameters from Section 4.1.1 ($\tau_{\text{LSO}} = 41$ ns, $\tau_{\text{LP}} = 26$ ns and $\tau_{\text{HP}} = 1000$ ns) the resulting peak position is at $t_{\text{peak}} \approx 174.66$ ns.

The relative peak underestimation error has been calculated numerically for different ADC sampling frequencies. Since the analog pulse can start at any arbitrary offset ϕ relative to the fixed ADC clock (cf. Section 4.1.3.2), the simulation has to evaluate all possible positions of the two sampling points around the peak position and determine the deviation from the real peak value as illustrated in Figure 4.14. The maximum deviation of the two sampling values is then shown in Figure 4.15 for different ADC sampling frequencies.

4.3.3.2 Parabola Interpolation

To compensate the error which is introduced by the time spacing of the ADC sampling values, several sampling points can be combined to interpolate a more precise value for the pulse amplitude. Since a linear interpolation can not improve the amplitude measurement, a quadratic interpolation is the next simplest approach [Mod91]. Therefore, the sample value closest to the peak is first determined by the peak-hold method. In addition, with the two adjacent sampling points a parabola function can be fitted to the sampled data as shown in Figure 4.16. By evaluating the maximum of the interpolated parabola, a more precise estimate for the pulse amplitude is obtained. The interpolation parabola is given by

$$(4.27) \quad p(\tau) = \alpha\tau^2 + \beta\tau + \gamma.$$

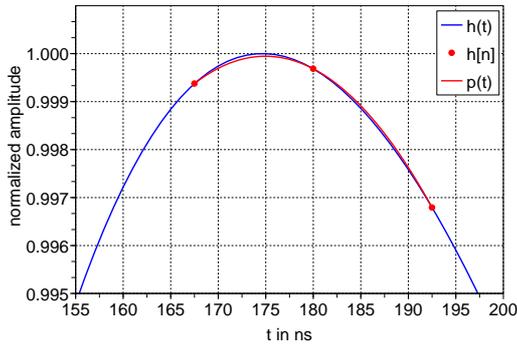


Figure 4.16: Parabola interpolation $p(t)$ of the continuous pulse $h(t)$, based on sampled data $h[n]$.

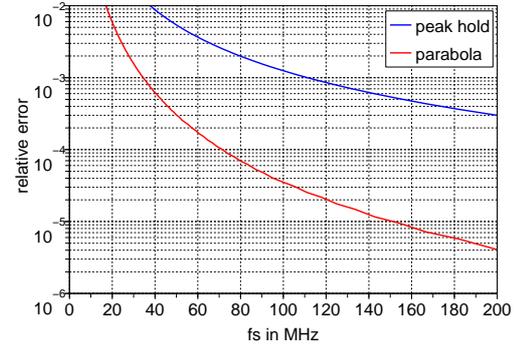


Figure 4.17: Relative maximum error versus sampling frequency for the parabola method (blue) and peak-hold (red).

At first, the peak-hold circuit has to determine the sample data maximum value $x_{\max} = x[n_{\max}]$ and the corresponding sample point index n_{\max} . The three parabola parameters can then be calculated by

$$(4.28) \quad \alpha = \frac{x[n_{\max} - 1] - 2 \cdot x[n_{\max}] + x[n_{\max} + 1]}{2 \cdot \Delta t^2},$$

$$(4.29) \quad \beta = \frac{x[n_{\max} + 1] - x[n_{\max} - 1]}{2 \cdot \Delta t},$$

$$(4.30) \quad \gamma = x[n_{\max}],$$

where $x[n_{\max} - 1]$ and $x[n_{\max} + 1]$ are the two adjacent sampling points to the peak-hold maximum. The distance of the sampling points is given by Δt . To find the interpolated maximum, Equation (4.27) has to be derived and solved for $p'(\hat{t}_{\max}) = 0$. This leads in a first step to the interpolated time-position of the maximum between the sampling points, relative to the time stamp of the n_{\max} sample

$$(4.31) \quad \begin{aligned} \hat{t}_{\max} &= \frac{-\beta}{2 \cdot \alpha} \\ &= \frac{x[n_{\max} - 1] - x[n_{\max} + 1]}{x[n_{\max} - 1] - 2 \cdot x[n_{\max}] + x[n_{\max} + 1]} \cdot \frac{\Delta t}{2}. \end{aligned}$$

By evaluating the parabola equation (4.27) at this position, finally the interpolated maximum value \hat{x}_{\max} is obtained

$$(4.32) \quad \begin{aligned} \hat{x}_{\max} &= \alpha \cdot \hat{t}_{\max}^2 + \beta \cdot \hat{t}_{\max} + \gamma \\ &= \frac{-\beta^2}{4 \cdot \alpha} + \gamma. \end{aligned}$$

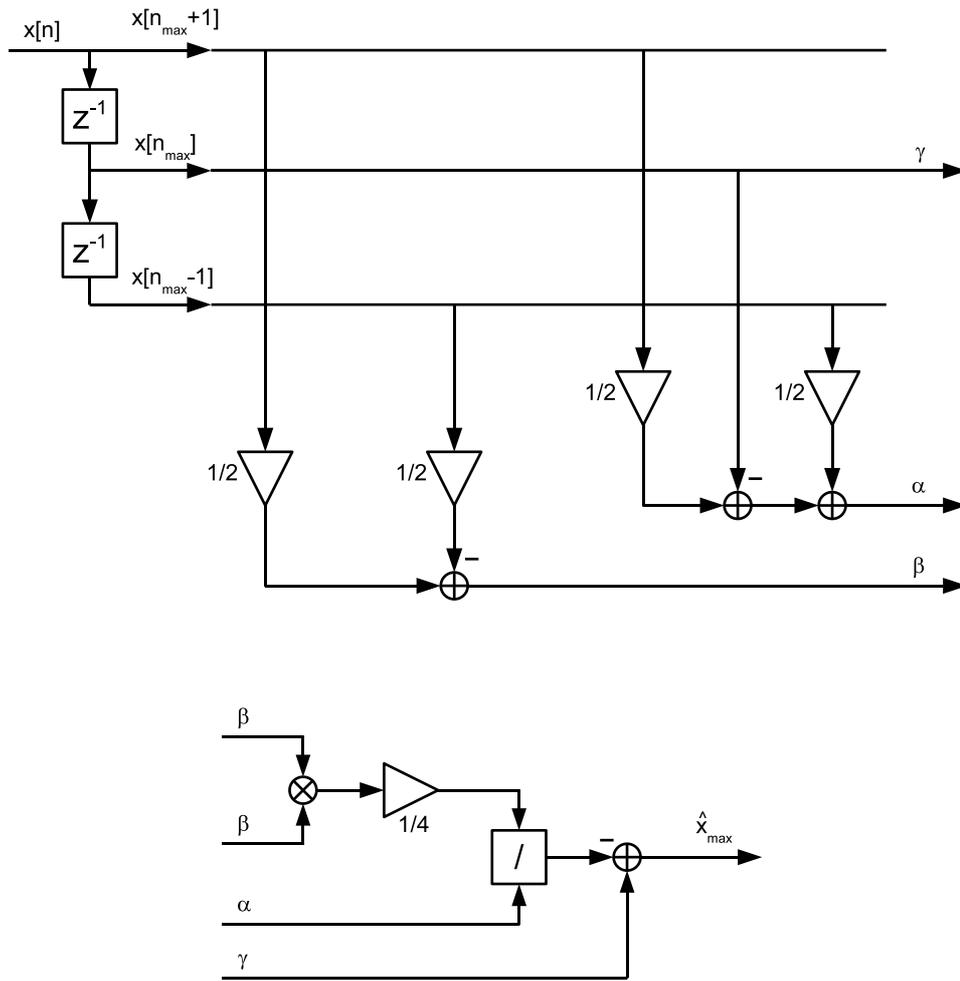


Figure 4.18: Data flow for the parabola based maximum interpolation.

Figure 4.17 shows the relative peak underestimation error, which is again calculated numerically for different ADC sampling frequencies. Compared to the peak-hold method, the error is significantly improved especially for lower sampling rates. However, the required hardware effort for the parabola interpolation is now more complex.

Figure 4.18 gives an overview of the data flow for the parabola maximum interpolation. To ease the calculation, the sampling interval for the interpolation can be set to $\Delta t = 1$, without changing the final result for \hat{x}_{\max} . The division operations $1/2$ and $1/4$ can be realized by a simple right-shift operation in the binary number system, without any additional delay or hardware effort. Only the division by α in (4.32) has to be implemented explicitly, which can be realized on one hand by a reciprocal lookup-table for the denominator, followed by a multiplication with the numerator. Since the memory requirements for the lookup-table rapidly grow once the bit-width of the numbers increases, a divider-algorithm-based implementation is usually more space efficient, at the cost of longer calculation times. Since the amplitude extraction is not as time critical as the previous trigger decision, a longer processing time can be accepted.

The total hardware requirement includes four additions, one multiplication and one division operation for the calculation of one parabola interpolated maximum value. In addition, the peak-hold functionality from Section 4.3.3.1 is required for the initial determination of $x[n_{\max}]$, which adds one compare operation. The peak-hold storage register can be combined with the two z^{-1} delay registers shown in Figure 4.18. Therefore, only two registers are required in total. However, due to the long critical path, which combines multiple operations in a chain without registers, the achievable clock speed is limited. If necessary, the critical path for the calculation of α , β and γ in the upper part of Figure 4.18 can be reduced by transposing the FIR structure and implementing a separate peak-hold circuit. Similarly, the final calculation of \hat{x}_{\max} can be pipelined by additional registers and a suitable implementation of the division operation.

4.3.3.3 Upsampling

Another approach for a more precise determination of the peak amplitude is based on first upsampling the initial data samples, followed by one of the previously described peak determination algorithms [F⁺08]. This concept implements only operations based on additions and multiplications which makes it suitable for signal processing hardware implementations. Further on, the upsampled data stream may also be useful for the subsequent pulse time calculations.

The upsampling operation describes how additional data points between initial sampling values can be calculated. Ideally, this process does not add or remove any information to/from the sampled analog signal itself. Instead it creates a dataset which could also be obtained by sampling the analog signal directly with the higher sampling rate. However, due to implementation constraints the interpolated data will usually not be identical to data which is sampled directly with a higher rate.



Figure 4.19: Upsampling data flow.

Figure 4.19 shows the principle data flow of the upsampling operation. Starting with the (low sampling rate) ADC values $x[n]$, the interpolated data sequence $y_{\text{int}}[n']$ is obtained in two steps [OS99, Lyo04]. First, an intermediate sequence $x_2[n']$ is created by adding zero valued samples between the data until the required number of interpolated data points is reached. A desired increment factor of M in the sampling frequency for the upsampled signal therefore corresponds to $M - 1$ additional zero values between two sampling points.

The intermediate data sequence then takes on the form

$$(4.33) \quad x_2[n'] = \begin{cases} x[n] & \text{for } n' = M \cdot n, \\ 0 & \text{otherwise.} \end{cases}$$

In the frequency domain, the additional zero values cause a scaling of the frequency axis by a factor of M , since the new sampling rate now is M times the old one. In addition, the frequency spectrum includes now also images of the original signal spectrum at multiples of the original sampling frequency. Figure 4.20 depicts the frequency normalized spectra of the original signal $X[f] = \mathcal{F}(x[n])$ compared to the expanded, intermediate signal $X_2[f] = \mathcal{F}(x_2[n'])$ for an upsampling factor of $M = 16$. Therefore, the expanded signal spectrum shows $M - 1$ additional image components of the initial signal.

In the second interpolation step, the zero padded signal $x_2[n]$ has to be low-pass filtered to remove the unwanted imaging components in the frequency domain. After the filter operation, the padded zero values of $x_2[n]$ then take on the interpolated signal values between the initial sampling points. Therefore, the required ideal filter response has to be of the form

$$(4.34) \quad H_{\text{int}}(\omega) = \begin{cases} M & \text{for } \omega \leq \pi/M, \\ 0 & \text{otherwise.} \end{cases}$$

Given that the original signal $x[n]$ has $1/M$ times less signal samples which carry the signal energy, the interpolation filter has to amplify in the pass-band by a factor of M . This compensates for the added zero values, which take over some part of the initial signal energy after the filter operation. Figure 4.21 shows the frequency response of two exemplary FIR upsampling filter implementations with filter length $L = 128$ and $L = 512$. The pass-band was chosen to extend up to $1/32$ of the normalized frequency units, since the first image component peaks already at $1/16$. This causes some overlap of the transition band with some part of the first image component. Additionally, the stop-band attenuation also varies with the implemented filter length.

Unfortunately, the ideal brick-wall filter function in (4.34) is not realizable with finite length digital filters and has to be approximated by a proper selection of the length and the pass- and stop-band specifications of the implemented interpolation filter. Due to the resulting non-optimal filtering, the remaining imaging frequency components are still present as disturbances in the interpolated signal and cause additional errors to the amplitude evaluation.

Figure 4.22 shows pulse shapes for different upsampling operations on the signal $h[n]$ with a fixed interpolation factor of $M = 16$. The length L of the interpolation FIR filter was varied, with fixed specifications for pass- and stop-band. For $L = 128$ the filter is

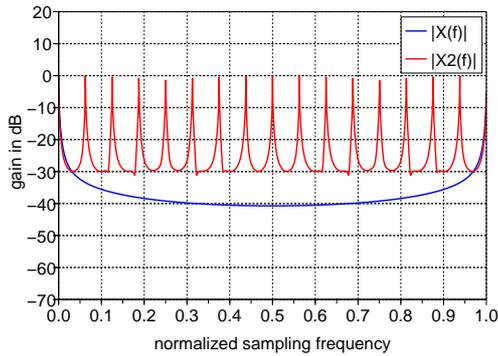


Figure 4.20: Spectrum of the original signal $X[f]$ (blue) and the expanded intermediate signal $X_2[f]$ (red).

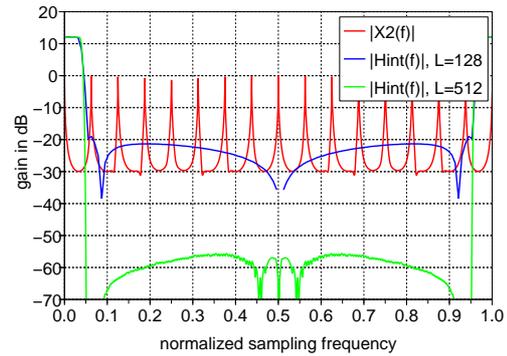


Figure 4.21: Frequency characteristics of different upsampling filters with $L = 128$ (blue) and $L = 512$ (green), compared to the spectrum of $x_2[n']$ (red).

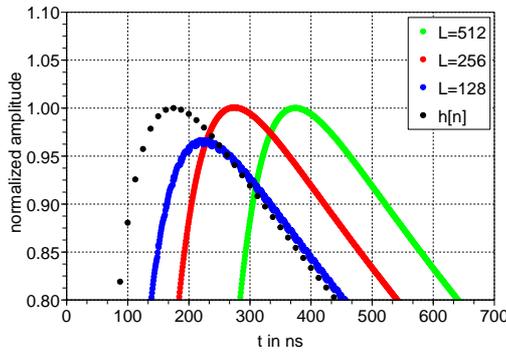


Figure 4.22: Upsampling of the signal $h[n]$ by $M = 16$, with varying interpolation filter length L .

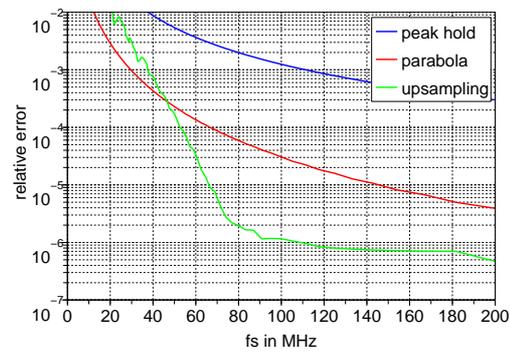


Figure 4.23: Relative maximum error versus sampling frequency for the upsampling method with $L = 512$ (green), the parabola method (red) and peak-hold (blue).

not able to suppress the imaging components sufficiently, which distorts the interpolated signal. With increased filter length, the attenuation of the imaging frequency components is increased and the amplitude of the interpolated signal is preserved. An increased filter length causes however also an additional delay of the upsampled signal.

Figure 4.23 compares the error of a peak-hold maximum determination from the upsampled dataset with the previously mentioned methods. Since the upsampling process includes information from multiple data points, the performance of the three-point parabola interpolation can be outreached with optimal filter settings. For low sampling rates, again, the real signal components are affected by the interpolation filter, which leads to a degradation of the pulse shape and therefore a fast increase of the error value.

The hardware requirements are mainly dominated by the upsampling filter implementation. For the described FIR implementation, the required number of registers, additions and multiplications is proportional to the filter length L . More efficient filter implemen-

tations may be possible, but require additional design and implementation effort to ensure a stable operation. For the expansion operation, which adds the $M - 1$ zero valued samples, two requirement options are possible. If an online upsampling of the input data stream is required, the clock frequency for all following operations has to be M -fold higher, compared to the input clock. Alternatively, a buffering scheme can be implemented for the input data, which requires more registers or a separate buffer memory.

Since the upsampling procedure adds new data values, the total amount of data for the following processing steps is increased. This increases as well the processing time proportional to M , if all operations are running with the same clock frequency. Therefore, also the constraints on the available processing time budget have to be considered for the upsampling approach.

4.3.3.4 Spline-Based Interpolation

To avoid the design problem of the interpolation filter, the signal interpolation can also be based on spline functions [Uns99]. Whereas the upsampling approach is based on the frequency representation of the input signal and is therefore closely related to sampling theory, the spline based interpolation at first focuses only on a "smooth" interpolation of the input signal. The input samples (which are called *knots*) are connected by piecewise polynomials of degree d , based on certain smoothness constraints. The resulting piecewise-continuous signal description can then be sampled at a higher rate to obtain the interpolated sampling points.

The basic building block for spline functions are the so called B-splines (with B for *basis* or *basic*). The B-spline functions start with a rectangular pulse for order $d = 0$. Higher orders are then obtained through recursive convolution of this pulse with itself:

$$(4.35) \quad \beta^0(t) = \begin{cases} 1 & \text{for } -\frac{1}{2} < t < \frac{1}{2}, \\ \frac{1}{2} & \text{for } |t| = \frac{1}{2}, \\ 0 & \text{otherwise.} \end{cases}$$

$$(4.36) \quad \beta^d(t) = \beta^{d-1}(t) * \beta^0(t) = \underbrace{\beta^0(t) * \beta^0(t) * \dots * \beta^0(t)}_{d+1}.$$

Additionally, $\beta^d(t)$ can also be specified directly, using the unit step function $u(t)$ as already defined in (4.2):

$$(4.37) \quad \beta^d(t) = \sum_{j=0}^{d+1} \frac{(-1)^j}{d!} \binom{d+1}{j} \left(t + \frac{d+1}{2} - j\right)^d \cdot u\left(t + \frac{d+1}{2} - j\right).$$

Although each B-spline is a polynomial of degree d , the spline function and its derivatives up to order $(d - 1)$ are defined to be continuous at the knots. Therefore, only one degree of freedom remains for each spline segment. For the case of uniformly and unity spaced support knots—which is valid for the case of ADC data, sampled at a fixed frequency—any spline function $s(t)$ can be obtained by the weighted sum of several shifted versions of B-splines $\beta^d(t)$:

$$(4.38) \quad s(t) = \sum_{k=-\infty}^{\infty} c[k] \cdot \beta^d(t - k).$$

For the interpolation of the sampled ADC data $x[n]$, again an upsampling factor M can be defined similar to the upsampling algorithm (Section 4.3.3.3). This corresponds to the uniform interpolation of $M - 1$ additional data points between the initial ADC samples $x[n]$. The interpolation procedure now first has to determine the B-spline coefficients $c[k]$, based on $x[n]$ as support knots for the spline function. In a second step, the continuous spline function $s(t)$ can be evaluated at an M times higher sampling rate to obtain the interpolated data.

In addition to the usual matrix based spline interpolation [dB78], this problem can also be addressed by a digital filter approach [UAE91a, UAE93a, UAE93b], which allows to reduce the computational effort significantly. Therefore, the continuous B-spline basis functions $\beta^d(t)$ are sampled to obtain a discrete B-spline function $b_M^d[k]$. The process of sampling at a higher sampling rate compared to the spline support knots, is described by the upsampling factor M :

$$(4.39) \quad b_M^d[k] = \beta^d(k/M).$$

This allows to derive z-domain filter coefficients from the sampled B-spline functions:

$$(4.40) \quad B_1^d(z) = \sum_{k=0}^{d+1} b_1^d[k] \cdot z^{-k}.$$

These coefficients can as well be derived by a set of recursion formulas as given in [UAE93a], which can be easier evaluated than (4.38), especially for higher spline degrees d . For cubic B-splines of degree $d = 3$ the resulting *indirect B-spline filter* is of the form

$$(4.41) \quad B_1^3(z) = \frac{z + 4 + z^{-1}}{6},$$

which corresponds to the sample values of the continuous and symmetric B-spline $\beta^3(-1) = 1/6$, $\beta^3(0) = 4/6$ and $\beta^3(1) = 1/6$. The problem of reconstructing signal points $s[k]$ from the spline coefficients $c[k]$ can then be formulated as a convolution with the sampled B-spline, based on (4.38):

$$(4.42) \quad s[k] = b_M^d[k] * c[k],$$

$$(4.43) \quad S(z) = B_M^d(z) \cdot C(z).$$

In the same way, the spline coefficients $C(z)$ can be derived from the signal $S(z)$ by filtering with the inverse of the sampled B-spline:

$$(4.44) \quad C(z) = B_M^d(z)^{-1} \cdot S(z),$$

with $S(z) = X(z) = \mathcal{Z}(x[n])$ representing the input data knots. The expression $B_M^d(z)^{-1}$ is also called the *direct B-spline filter*.

When the indirect cubic spline filter (4.41) is inverted, it becomes an IIR filter with one pole at $z_\infty = \sqrt{3} - 2$ and a second pole at $1/z_\infty$, which can be decomposed into a sum term:

$$(4.45) \quad \begin{aligned} B_1^3(z)^{-1} &= \frac{6}{z + 4 + z^{-1}} \\ &= \frac{-6 \cdot z_\infty}{(1 - z_\infty \cdot z^{-1})(1 - z_\infty \cdot z)} \\ &= \frac{-6 \cdot z_\infty}{(1 - z_\infty^2)} \left(\underbrace{\frac{1}{(1 - z_\infty \cdot z^{-1})}}_{\text{causal}} + \underbrace{\frac{1}{(1 - z_\infty \cdot z)}}_{\text{anticausal}} - 1 \right). \end{aligned}$$

The sum term then allows to extract two recursive equations: One with increasing time index for the causal term ($c_a[k]$) and one with reversed time index for the anticausal term ($c_b[k]$). Finally, the combination of both terms results in the final spline coefficients $c[k]$.

$$(4.46) \quad \begin{aligned} c_a[k] &= x[k] + z_\infty \cdot c_a[k - 1] && \text{for } k = 2, 3, \dots, K, \\ c_b[k] &= x[k] + z_\infty \cdot c_b[k + 1] && \text{for } k = K - 1, K - 2, \dots, 1, \\ c[k] &= \frac{-6 \cdot z_\infty}{(1 - z_\infty^2)} \cdot (c_a[k] + c_b[k] - x[k]) && \text{for } k = 1, 2, \dots, K. \end{aligned}$$

When the input signal is extended by its mirror image at the endpoints, the following initialization values for the recursion are obtained:

$$(4.47) \quad \begin{aligned} c_a[1] &= \sum_{k=1}^K z_{\infty}^{|k-1|} \cdot x[k], \\ c_b[K] &= c_a[K]. \end{aligned}$$

Based on (4.46) and (4.47) it is now possible to calculate cubic spline coefficients $c[n]$ based on input data samples $x[n]$. To be able to interpolate additional data points, the obtained spline coefficients are processed further. The problem to derive upsampled data can be described as a convolution, based on (4.38) and (4.42):

$$(4.48) \quad s_M[k'] = s[k'/M] = \sum_{i=-\infty}^{\infty} c[i] \cdot b_M^d[k' - i \cdot M],$$

$$(4.49) \quad s_M[k'] = b_M^d[k'] * c_M[k'].$$

As the spline coefficients are calculated from the non-upsampled input data, coefficients are only available for the condition $k' = M \cdot k$. The remaining values are therefore initialized with zeros:

$$(4.50) \quad c_M[k'] = \mathcal{Z}^{-1}(C(z^M)) = \begin{cases} c[k] & \text{for } k' = M \cdot k, \\ 0 & \text{otherwise.} \end{cases}$$

The convolution in (4.49) can then be evaluated in the z-domain [UAE91a]:

$$(4.51) \quad \begin{aligned} S_M(z) &= B_M^d(z) \cdot C(z^M) \\ &= B_1^d(z) \cdot \frac{1}{M^d} (B_M^0(z))^{d+1} \cdot C(z^M) \\ &= \underbrace{B_1^d(z)}_{\text{B-spline}} \cdot \underbrace{\frac{1}{M^{d+1}} (B_M^0(z))^{d+1}}_{\text{moving average}} \cdot M \cdot \underbrace{C(z^M)}_{\text{coefficients}}. \end{aligned}$$

Equation (4.51) allows now to extract the operations for the final interpolation procedure: First, the calculated spline coefficients $c[n]$ are expanded with zeros according to (4.50) and multiplied by M . Since $B_M^0(z)$ corresponds to a rectangular pulse of length M and unit amplitude, the expression $(1/M) \cdot B_M^0(z)$ realizes a moving average filter of size M .

Due to the exponent, $d + 1$ moving average filters are cascaded to smooth the upsampled coefficients. When combined with the upsampled operation, the first moving average can as well be interpreted as a 0 order interpolation of the coefficient values which is then followed by the d remaining average filters. Finally, the indirect B-spline filter is applied to the smoothed coefficient set, which provides the series of interpolated data samples.

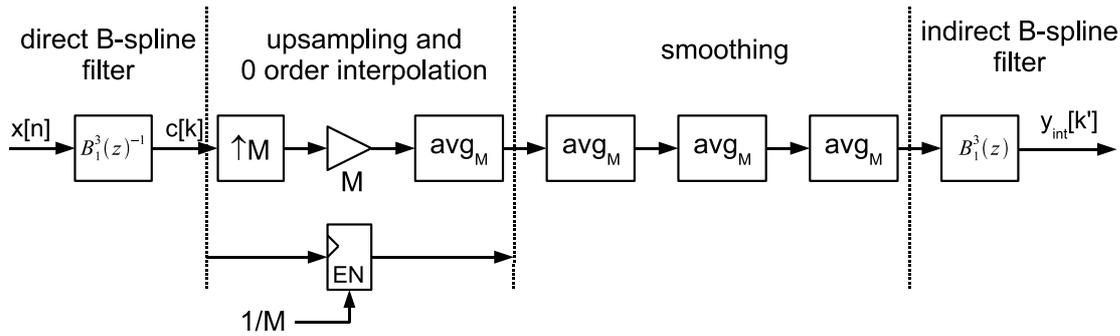


Figure 4.24: Data flow for the cubic spline based interpolation.

Figure 4.24 gives a full overview of the processing steps for the cubic spline based upsampling procedure: First, the incoming data are filtered by the direct B-spline filter (4.45), which is realized by the recursive equations (4.46). The calculated spline coefficients are then upsampled by M and extended by a 0 order interpolation, which repeats the coefficient values to replace the zeros in (4.50). Alternatively, the process of upsampling combined with the 0 order interpolation can be replaced by a simple storage register which holds every coefficient value for M clock cycles. After the upsampling operation, the coefficient values are smoothed by the three consecutive moving average filters of size M . The implementation of moving average filters was already discussed in Section 4.2.1. Finally, the smoothed coefficient set is filtered by the indirect B-spline filter (4.41) to obtain the interpolated data $y_{\text{int}}[k']$. Since this is a simple FIR filter, it can be implemented in a straightforward way as given in the two FIR cases in Figure 4.8.

Figure 4.25 shows the interpolated pulse shape of $h[n]$ with an upsampling factor of $M = 16$. Compared to the classic upsampling approach described in Section 4.3.3.3 the design of a steep frequency selective interpolation filter is no longer necessary. Figure 4.26 compares again the error for the determination of the pulse maximum value from an interpolated dataset. Therefore, the input signal was interpolated with $M = 16$ and $M = 32$, respectively, and the maximum was determined by the peak-hold method. For comparison, also the parabola interpolation method, based on the original input data, is shown. Again, the performance of the maximum estimation is improved, and can be increased with higher interpolation factors. Compared to the performance of the classic upsampling method in Figure 4.23, the spline based method gives better results especially at low sampling rates. In contrast to the classic upsampling approach, the spline interpolation still maintains a “smooth” signal shape in the case of low sampling rates and does not depend on optimized upsampling filters.

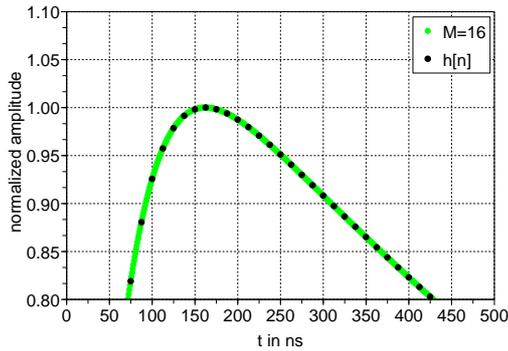


Figure 4.25: Cubic spline interpolation of $h[n]$ with upsampling factor $M = 16$.

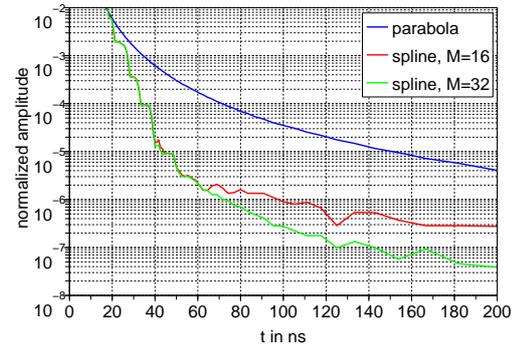


Figure 4.26: Relative maximum error versus sampling frequency for the cubic spline interpolation method with $M = 16$ (red) and $M = 32$ (green). The parabola method is shown in (blue) for comparison.

Regarding the hardware complexity, the structure in Figure 4.24 is more complicated than the classic upsampling algorithm. However, each of the sub-blocks is quite simple and requires only few operations. The most complex part is the implementation of the first direct B-spline filter block. Due to the causal/anticausal structure and the derived recursive equations (4.46) the input data have to be buffered for multiple read access. Additionally, intermediate storage is required for the $c_a[k]$ coefficient vector which has to be time reversed for the second iteration. Therefore, the spline interpolation can only operate on finite datasets and not on a continuous stream of input data. Since all expressions which involve z_∞ can be precalculated, the calculation of one spline coefficient involves four additions and three multiplications as given by (4.46). Additionally, the initialization of the $c_a[1]$ element requires two multiplications and one addition for each input data sample. However, since the $z_\infty^{|k-1|}$ term decreases with each step, the calculation can be aborted after a certain precision is reached.

Once the spline coefficients are calculated, the upsampling and first interpolation step require only one storage register. The following three stage smoothing operations require two additions within each stage and for each upsampled coefficient value. Each average filter block requires also $M + 1$ storage elements if the implementation is based on the recursive-running-sum structure in Figure 4.8. The final indirect B-spline filter can then be realized by a simple FIR structure with three multiplications and two additions. In total, for one input value $5 + M \cdot 8$ additions and $5 + M \cdot 3$ multiplications are executed.

4.3.3.5 Smoothing-Spline Interpolation

Especially for noisy input data, it is not necessarily desired to have a perfect representation of the support knots by the interpolation spline function. Instead, the spline smoothness property could be enhanced to recover the pulse shape from the noisy data samples.

This can be achieved by *smoothing-splines*, which are as well realizable by a digital filter based implementation described in [UAE93a, UAE93b]. The problem of finding the corresponding smoothing spline representation $\hat{s}(k)$ of a discrete signal $s[k]$ is therefore formulated by an error function, which has to be minimized

$$(4.52) \quad \epsilon^2 = \sum_{k=-\infty}^{\infty} (s[k] - \hat{s}(k))^2 + \lambda \cdot \int_{-\infty}^{\infty} \left(\frac{\partial^r \hat{s}(x)}{\partial x^r} \right)^2 dx$$

By changing the positive parameter λ it is now possible to control the influence of the spline function smoothness on the error function. Hence, higher values correspond to a smoother spline representation. The order of the spline function is $2r - 1$, which gives for the case of cubic B-splines a value of $r = (3 + 1)/2 = 2$.

By minimizing the error function as stated in [UAE93a], a z-domain expression can be derived for the smoothing spline coefficients $C(z)$, with the input signal $S(z)$ and the impulse response for the smoothing spline filter $G(z)$:

$$(4.53) \quad \begin{aligned} C(z) &= G(z) \cdot S(z) \\ &= \frac{1}{B_1^3(z) + \lambda \cdot (-z + 2 - z^{-1})^2} \cdot S(z) \\ &= \frac{6}{z + 4 + z^{-1} + 6\lambda \cdot (z^{-2} - 4z^{-1} + 6 - 4z + z^2)} \cdot S(z). \end{aligned}$$

This describes again an IIR filter, which can be implemented by recursive equations, similar to Section 4.3.3.4. Therefore, the impulse response $G(z)$ is first factorized into the causal and anticausal term

$$(4.54) \quad \begin{aligned} G(z) &= G^+(z) \cdot G^+(z^{-1}) \\ &= \frac{1 - 2\rho \cos(\omega) + \rho^2}{1 - 2\rho \cos(\omega) \cdot z^{-1} + \rho^2 \cdot z^{-2}} \cdot \frac{1 - 2\rho \cos(\omega) + \rho^2}{1 - 2\rho \cos(\omega) \cdot z + \rho^2 \cdot z^2} \\ &= \frac{b_0}{1 - a_1 z^{-1} - a_2 z^{-2}} \cdot \frac{b_0}{1 - a_1 z^1 - a_2 z^2}. \end{aligned}$$

with ρ and ω being the magnitude and argument of the smallest complex conjugated roots of the denominator of $G(z)$. The laborious determination of ρ and ω was carried out by [UAE93a] and results in

$$(4.55) \quad \rho = \left(\frac{24\lambda - 1 - \sqrt{\xi}}{24\lambda} \right) \cdot \left(\frac{48\lambda - 24\lambda\sqrt{3 + 144\lambda}}{\xi} \right)^{1/2},$$

$$(4.56) \quad \omega = \arctan \left(\left(\frac{144\lambda - 1}{\xi} \right)^{1/2} \right),$$

with

$$(4.57) \quad \xi = 1 - 96\lambda + 24\lambda\sqrt{3 + 144\lambda}.$$

From this, a set of recursion equations for $C(z)$ can be formulated by the method described in [UAE91b]:

$$(4.58) \quad \begin{aligned} c_a[k] &= x[k] + a_1 \cdot c_a[k-1] + a_2 \cdot c_a[k-2] & \text{for } k = 3, 4, \dots, K, \\ c_b[k] &= c_a[k] + a_1 \cdot c_b[k+1] + a_2 \cdot c_b[k+2] & \text{for } k = K-2, K-3, \dots, 1, \\ c[k] &= b_0^2 \cdot c_b[k] & \text{for } k = 1, 2, \dots, K, \end{aligned}$$

with the filter factors

$$(4.59) \quad \begin{aligned} a_1 &= 1 - 2\rho \cos(\omega), \\ a_2 &= -\rho^2, \\ b_0 &= 1 - 2\rho \cos(\omega) + \rho^2. \end{aligned}$$

At the beginning of the recursion, the first vector elements have to be initialized in the following way:

$$(4.60) \quad \begin{aligned} c_a[k] &= \frac{x[k]}{1 - a_1 - a_2} & \text{for } k = 1, 2, \\ c_b[k] &= \frac{c_a[k]}{1 - a_1 - a_2} & \text{for } k = K-1, K. \end{aligned}$$

For a given smoothing parameter value λ , the filter factors in (4.59) can be determined and used as constants during the recursion (4.58). After both causal ($c_a[k]$) and anti-causal ($c_b[k]$) coefficient vectors are calculated, the final multiplication with b_0^2 gives the smoothing spline coefficients⁹. The spline coefficient description can then be handled in the same way as if obtained from the direct B-spline filter in Section 4.3.3.4. Hence, signal interpolation is also achieved in the same way as depicted in Figure 4.24.

The hardware complexity is slightly increased, compared to the three multiplications and four additions of the standard B-spline filter. The calculation of one smoothing spline coefficient value, which corresponds as well to one input sample, requires five multiplications and four additions. If the denominator in the initialization (4.60) is precalculated, it can be stored as reciprocal value, which allows to replace the four division operations by multiplications.

⁹ b_0^2 is here the filter constant b_0 squared, not the sampled B-spline kernel as given in (4.39).

4.3.4 Sensitivity to Noise

Until now, the presented methods for determination of the pulse amplitude value, were only evaluated with noiseless sample pulses. The real ADC pulse data values are however distorted by additional noise, which will degrade the results compared to the noiseless case. Therefore, the algorithms were also simulated with added Gaussian noise on the data values. Since the expected signal amplitude is usually confined to small amplitude ranges compared to the noise level, the signal to noise ratio (SNR) is defined here without the $\log()$ expression, common to other signal processing applications:

$$(4.61) \quad SNR = \frac{\text{mean signal amplitude}}{\text{std. dev. of noise}} = \frac{\overline{A_{\text{signal}}}}{\sigma_{\text{noise}}}.$$

Figures 4.27, 4.28 and 4.29 show the simulation results for the described algorithms for different signal to noise ratios. The black curve indicates thereby the result for the noiseless case with $\sigma_{\text{noise}} = 0$ and therefore $SNR = \infty$. The left-hand plots show the mean value of the calculated signal amplitude for varying sampling frequencies. The right-hand plots include the corresponding standard deviation.

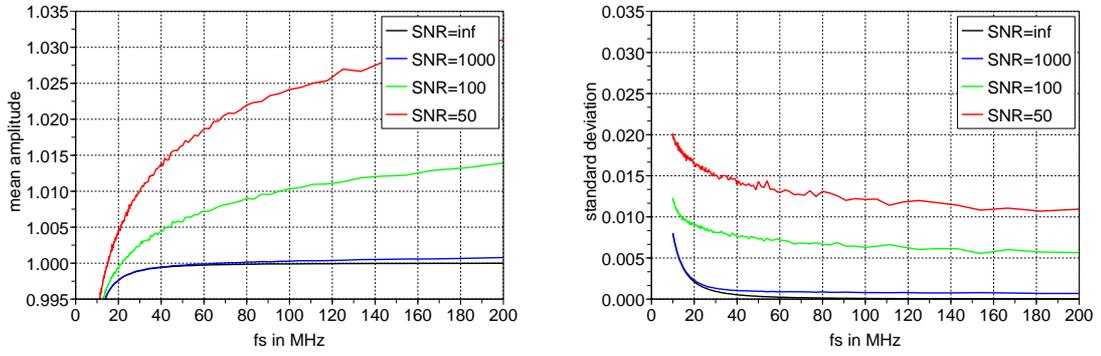
For the simulation, the normalized analog pulse form was sampled at different sampling rates conforming to (4.10). Since the time offset ϕ of the analog pulse to the ADC sampling clock is uniformly distributed (cf. Section 4.1.3.2), the ϕ value was varied in steps of 0.5 ns between 0 and t_s . Ideally, the reconstructed maximum value should always be equal to one, which is the normalized amplitude value.

Regarding the noiseless (*black*) signal first, all methods reach the expected amplitude value of 1.0 for sampling rates above 60 MHz. For smaller sampling rates, the reconstructed amplitude starts to deviate from the optimum due to undersampling effects. This also affects the standard deviation, due to the variation of the time offset ϕ . For lower signal to noise ratios, the standard deviation is increased, but shows only minor improvement once the sampling rate increases above 60–80 MHz.

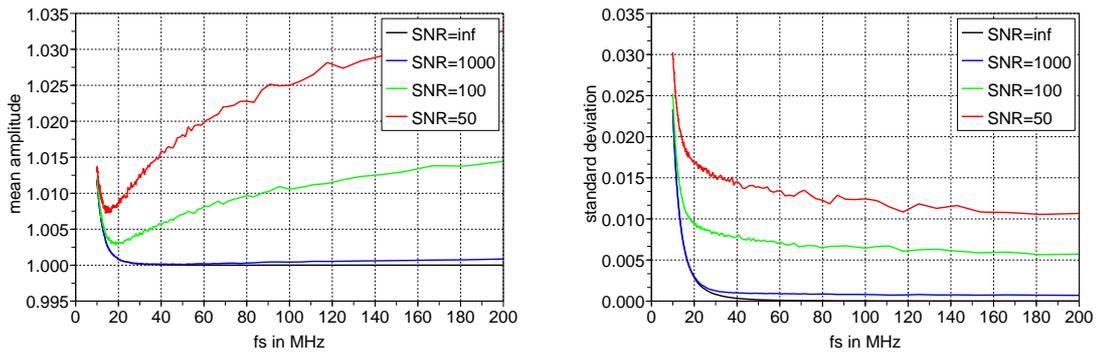
The constant increment of the mean amplitude for the noisy signals at higher sampling rates can be explained by the properties of the added Gaussian noise. Since the normalized pulse amplitude is fixed, only the added noise can influence the maximum calculation. Additionally, for an increasingly smaller distance between the sampling points, the noiseless pulse shape appears more and more flat around the maximum, which can be confirmed by Figure 4.4. Therefore, the estimated maximum value is essentially equal to the maximum of a series of Gaussian distributed values.

$$(4.62) \quad \hat{x}_{\max} = \max \{x_1, x_2, \dots, x_N\}, \quad \text{with } x_i \sim \mathcal{N}(\mu, \sigma^2).$$

peak-hold



parabola interpolation



upsampling $M = 16, L = 512$

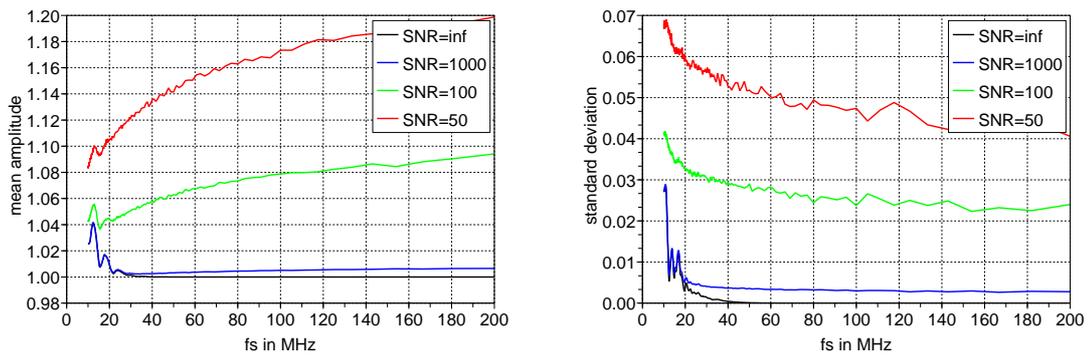
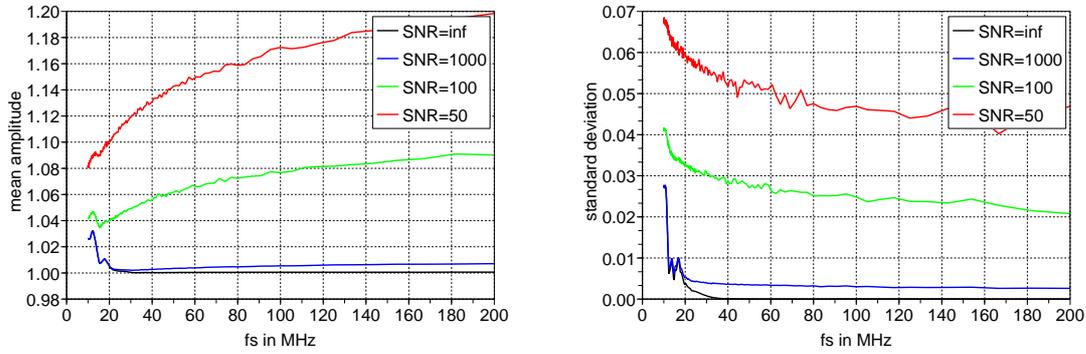
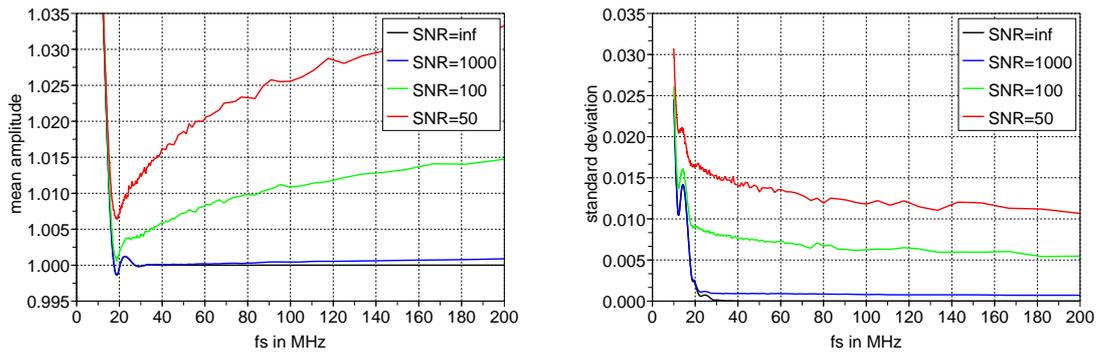


Figure 4.27: Comparison of the relative maximum error and the standard deviation of the obtained maximum value for noisy data samples at different SNR values for peak-hold (*top*), parabola interpolation (*mid*) and upsampling $M = 16, L = 512$ (*bottom*).

upsampling $M = 16, L = 256$



cubic spline interpolation $M = 8$



cubic spline interpolation $M = 16$

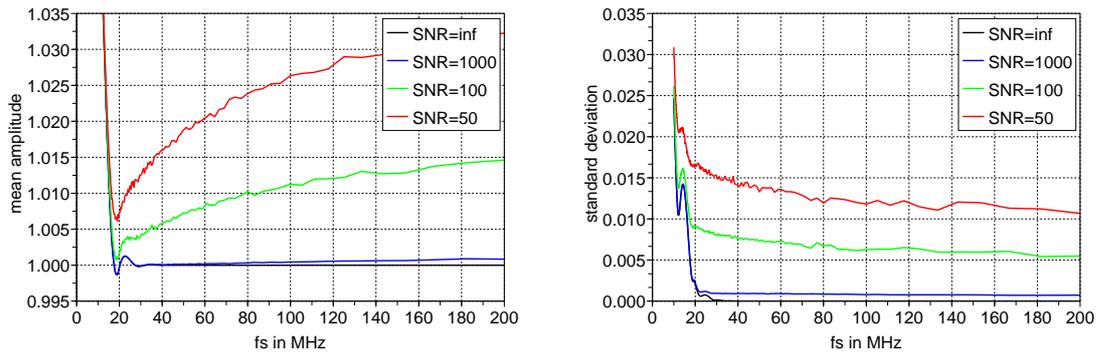
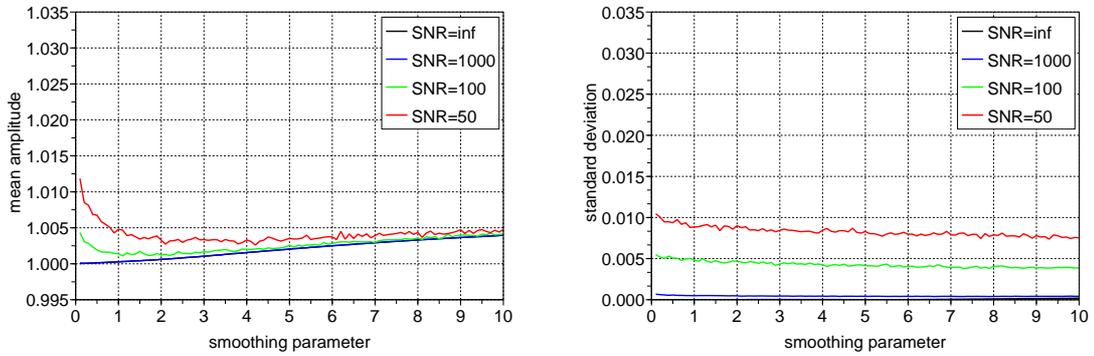
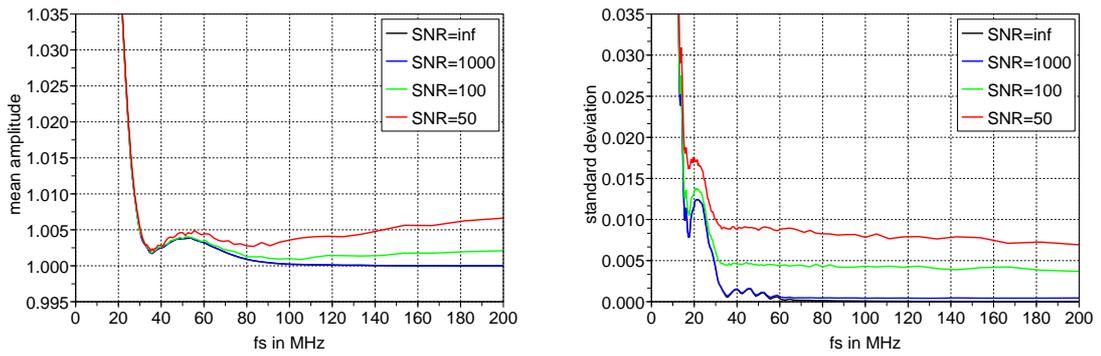


Figure 4.28: Comparison of the relative maximum error and the standard deviation of the obtained maximum value for noisy data samples at different SNR values for upsampling $M = 16, L = 256$ (top) and cubic spline interpolation with $M = 8$ (mid) and $M = 16$ (bottom).

smoothing cubic spline interpolation $M = 8, f_s = 80$ MHz
 λ scan



smoothing cubic spline interpolation $M = 1, \lambda = 3$



smoothing cubic spline interpolation $M = 8, \lambda = 3$

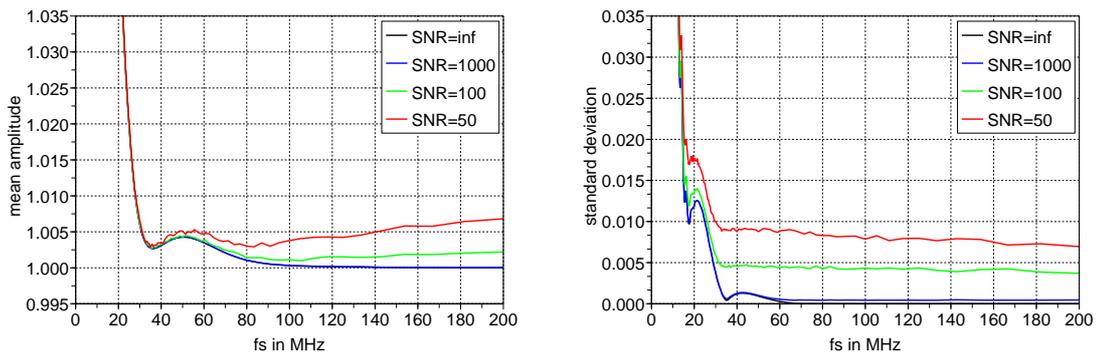


Figure 4.29: Smoothing cubic spline interpolation λ parameter scan (top) and comparison of the relative maximum error and the standard deviation of the obtained maximum value for noisy data samples at different SNR values for smoothing cubic spline interpolation $M = 1, \lambda = 3$ (mid) and $M = 8, \lambda = 3$ (bottom).

Given that the maximum value of a Gaussian distributed variable is not limited, \hat{x}_{\max} grows to infinity, for $N \rightarrow \infty$. Accordingly, the mean maximum value $\overline{\hat{x}_{\max}}$ also increases with increasing sample number N . However, once the sampling rate is fixed by the selected ADC system, this effect is no longer critical for the implementation of the algorithms. The introduced offset can then be corrected by the energy calibration of the PET system, which is anyway required due to the gain variations in the detectors and the analog readout system.

To evaluate the performance of the different algorithms, the standard deviation of the estimated maximum value can be compared. Table 4.1 lists the standard deviation of the algorithms for different sampling rates and signal to noise ratios. For the ideal, noiseless case with infinite SNR, the rather high values for the simple peak-hold algorithm are caused by the inevitable signal undersampling for this method. The interpolation methods allow to reduce this value already for the low sampling rate of 60 MHz. However, also here undersampling effects are obvious, since the standard deviation decreases further for increased sampling frequencies. The upsampling ($M = 16, L = 512$) and spline interpolation ($M = 16$) methods give similar results, with slightly better performance for the spline algorithm, and outperform the peak-hold and parabola interpolation methods. The smoothing spline method is comparable to the peak hold algorithm for an upsampling factor of $M = 1$. For $M = 8$, the performance improves comparable to the standard spline method.

For the noisy dataset, the SNR of 100 corresponds to a standard deviation of $\sigma = 0.01$ in the distribution of the added noise. The decreasing trend of the maximum standard deviation with higher sampling rates is still present, however the clear performance difference between the algorithms vanished. As already mentioned in Section 4.3.3.3, the upsampling method relies on the performance of the implemented upsampling filter. For the current specification from Figure 4.21, the filtered noise leads to large variations in the estimated maximum values. The other algorithms give almost similar results with no significant improvement for the spline interpolation. However, the smoothing spline algorithm is now superior to all other methods, as its low-pass characteristic suppresses the added noise and therefore enhances the true signal pulse. This effect is already visible for the upsampling factor $M = 1$ with no interpolation of sample values. A further increment of the upsampling factor gives no additional improvement in the standard deviation.

Regarding the implementation effort of the different algorithms, the peak-hold and parabola interpolation seem to be the most effective methods for extracting the pulse amplitude from the sampled data. For sampling rates below 60 MHz the parabola interpolation allows to recover partly from the signal undersampling. Below 40 MHz, also the parabola interpolation suffers from the missing pulse information. The smoothing spline based interpolation allows to improve the amplitude reconstruction especially for noisy input data, however, with significantly higher hardware effort compared to the parabola or peak-hold method.

Algorithm	SNR	60 MHz	80 MHz	100 MHz	140 MHz	200 MHz
peak-hold	∞	0.0002337	0.0001351	0.0000903	0.0000418	0.0000211
	100	0.0072605	0.0066947	0.0062747	0.0059072	0.0054583
parabola	∞	0.0000999	0.0000420	0.0000226	0.0000073	0.0000026
	100	0.0071644	0.0064700	0.0064509	0.0060797	0.0057237
upsampling $M = 16,$ $L = 512$	∞	0.0000171	0.0000006	0.0000004	0.0000003	0.0000003
	100	0.0280529	0.0244066	0.0237469	0.0248712	0.0239983
upsampling $M = 16,$ $L = 256$	∞	0.0000160	0.0000191	0.0000221	0.0000220	0.0000136
	100	0.0261001	0.0250981	0.0248330	0.0233758	0.0207999
spline $M = 8$	∞	0.0000036	0.0000021	0.0000014	0.0000007	0.0000004
	100	0.0073054	0.0067166	0.0062407	0.0059705	0.0054761
spline $M = 16$	∞	0.0000011	0.0000006	0.0000003	0.0000002	0.0000000
	100	0.0073142	0.0067276	0.0062475	0.0059791	0.0054851
smooth spline $M = 1, \lambda = 3$	∞	0.0002922	0.0001495	0.0000879	0.0000420	0.0000212
	100	0.0044148	0.0042363	0.0043147	0.0038927	0.0036931
smooth spline $M = 8, \lambda = 3$	∞	0.0002580	0.0000307	0.0000022	0.0000007	0.0000003
	100	0.0044403	0.0042614	0.0043344	0.0039082	0.0037087

Table 4.1: Standard deviation of the maximum estimation for selected SNR values.

4.3.5 Summary of the Amplitude Detection

Table 4.2 summarizes once more the hardware requirements of the algorithms, without optimization for pipelining and data throughput. The peak-hold method has by far the simplest demands on hardware resources. Since all other methods also include the peak-finder functionality, these resources are also added to their specific requirements. The parabola method requires only slightly more operations compared to the peak-hold, however this includes also the more complex division operation.

Besides the fact that the upsampling and spline interpolation algorithms increase the number of data words by the upsampling factor M , their requirements slightly differ. The classic upsampling is dominated by the filter implementation, which scales with the filter order L for a FIR realization. For the spline method, the requirements scale with the upsampling factor, since only the number of spline coefficients is increased. However, the two step calculation of the causal and anticausal filter terms requires more memory

Algorithm	ADD	MUL	DIV	CMP	REG
peak-hold	0	0	0	1	1
parabola	4	1	1	0 (+1)	2
upsampling	$L - 1$	L	0	0 (+1)	$L - 1 (+1)$
spline	$5 + M \cdot 8$	$5 + M \cdot 3$	0	0 (+1)	$2 \cdot N$ $+ 3 \cdot M$ $+ 6 (+1)$
smooth-spline	$4 + M \cdot 8$	$9 + M \cdot 3$	0	0 (+1)	$2 \cdot N$ $+ 3 \cdot M$ $+ 6 (+1)$

Table 4.2: Hardware requirements for the maximum extraction algorithms. L is the length of the upsampling filter, M is the upsampling factor and N is the number of input data words.

resources, since the calculation cannot be carried out directly on the sampled data stream. For the smoothing-spline method the dependence on the upsampling factor is the same as for the standard splines, with only different requirements for the coefficient initialization.

For applications with a sufficiently high sampling rate (≥ 60 MHz) the simple peak-hold algorithm gives sufficiently good results. Once the sampling rate is too low for the pulse shape, the parabola method can improve the results. If an interpolated and upsampled data stream is also advantageous for further signal processing, the spline based method provides a robust solution for interpolation of the input signal.

In addition, the peak-hold data from Table 4.1 allows to derive a constraint for the required ADC resolution. The achieved standard deviation for the input signal sampled with 80 MHz corresponds to an ADC resolution of $\log_2(1/0.0001351) \approx 12.9$ bit. Thus, a higher resolution than 13 bit would give no additional improvement, since the result is already limited by the random time shift of the signal. For the noisy signal, the resolution drops down to $\log_2(1/0.0066947) \approx 7.2$ bit. This region is easily met by current ADC families, which usually provide resolutions in the range of 8–14 bit for the given sampling rate range [TI09].

4.4 Time Reconstruction

Besides the amplitude of the detector signal, also the actual time when a signal pulse arises has to be determined. Since the positron annihilation produces two coincident 511 keV photons, both photons have to be registered in a detector in order to create a line-of-response (LOR) for the image reconstruction. To be able to identify the two coincident photon interactions in the random background, a precise time information from both interactions is required. Once two events within opposite detector pairs are sufficiently close in time, they will be considered as a coincident photon pair. If the achievable time resolution reaches values below 1 ns, the image reconstruction process can also profit from time-of-flight information which allows to confine the positron annihilation along the reconstructed LOR between two detector pairs.

A first time information may already be provided by the trigger decision, however, only with a very coarse resolution due to the limited ADC sampling frequency and the uncertainty of the trigger decision. To achieve a more precise time reconstruction, again the information of multiple sampling points of the signal waveform has to be used to interpolate the time information between the sampling points. This can be implemented using different methods with varying hardware effort and performance.

4.4.1 Leading Edge Timing

The most simple approach to derive a time information from the sampled pulse data is leading edge timing. Therefore, the pulse data are compared with a fixed threshold value A_{thr} similar to the trigger decision in (4.12). When the signal exceeds the threshold, the time index of the current sample value is then considered as the start time of the pulse.

Since the ADC sampling frequency determines the distance in time of the sampling points, it also determines the lower limit of the achievable time resolution. Unfortunately, the ADC sampling frequency can not be chosen arbitrarily high, since it is closely related to the ADC resolution, the power consumption and also the price of the ADC chip. Even though appropriate ADCs with sampling rates above 1 GHz (corresponding to a time resolution ≤ 1 ns) are available, the resulting data rate will significantly increase the requirements (and cost) of the data processing logic.

Although the achievable time resolution can be further reduced by interpolating the signal between the sampling points, a drawback of the leading edge approach is its dependence on the pulse amplitude. For higher amplitudes, the fixed threshold is exceeded earlier than for pulses with lower amplitudes, which introduces *amplitude walk* into the timing result as illustrated in Figure 4.30. To resolve this problem, the concept of constant fraction timing is used.

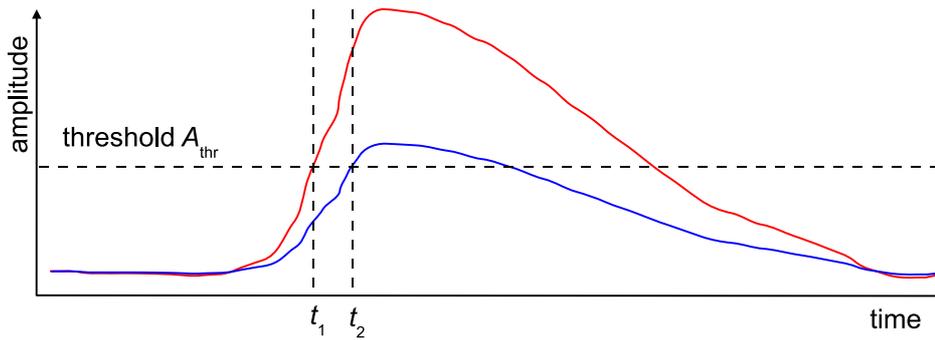


Figure 4.30: Leading edge amplitude walk: signals with high amplitude (*red*) exceed the threshold A_{thr} earlier than lower amplitude signals (*blue*), which causes a time mismatch between t_1 and t_2 .

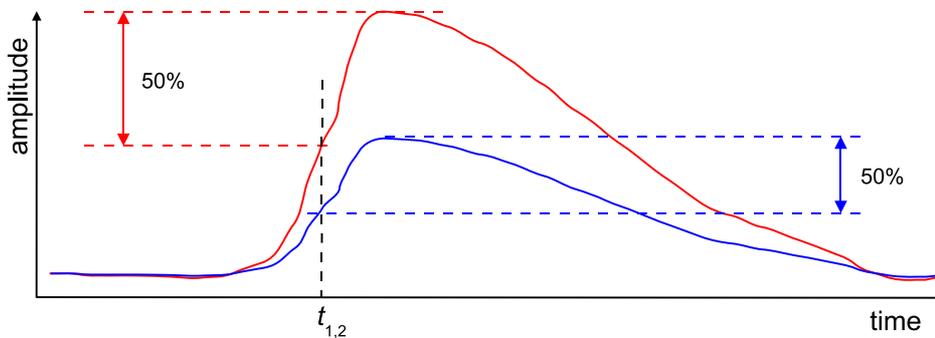


Figure 4.31: Constant fraction timing: for each signal amplitude an independent threshold value is calculated, based on the preset fraction (e.g. 50%), which results in matching time values ($t_1 = t_2$) for both signals.

4.4.2 Constant Fraction Timing

For constant fraction timing, the pulse data are compared to a variable threshold which is scaled with the pulse amplitude. Therefore, the pulse time is evaluated at a “constant fraction” of the pulse amplitude. For the assumption that the shape of the pulse does not change with different amplitude values, this gives a time information for the pulse independent of the amplitude. Figure 4.31 illustrates the constant fraction timing principle.

With analog signals, the constant fraction discriminator (CFD) is usually implemented by subtracting a delayed and scaled version of the input signal from the original signal [Kno99]. The resulting waveform then shows a zero crossing at the constant fraction threshold which can be evaluated by time-to-digital or time-to-amplitude converters. A similar approach is also implemented in the MADPET-II processing ASIC [Spa06].

If the data are available as sampled analog signals, the CFD algorithm can be implemented in a more direct way, as the amplitude value can be extracted from the data as described in the previous Section 4.3. A scaled threshold value can then be computed based on the amplitude and the specified fraction value, and used in the normal threshold decision already given in (4.12). This provides again a first time information for the

pulse, based on the ADC sampling interval. For further refinement, without increasing the sampling rate, the signal pulse must be interpolated between the sampling points.

As discussed previously (cf. Section 4.3.3), this can be accomplished by several methods with differing complexity. The easiest way is a linear interpolation (first order interpolation) between the data points. Since the CFD threshold is applied on the rising edge, the linear approximation introduces only small errors due to the steep signal slope. The principle of a linear interpolation based digital constant fraction discrimination algorithm is shown in Figure 4.32.

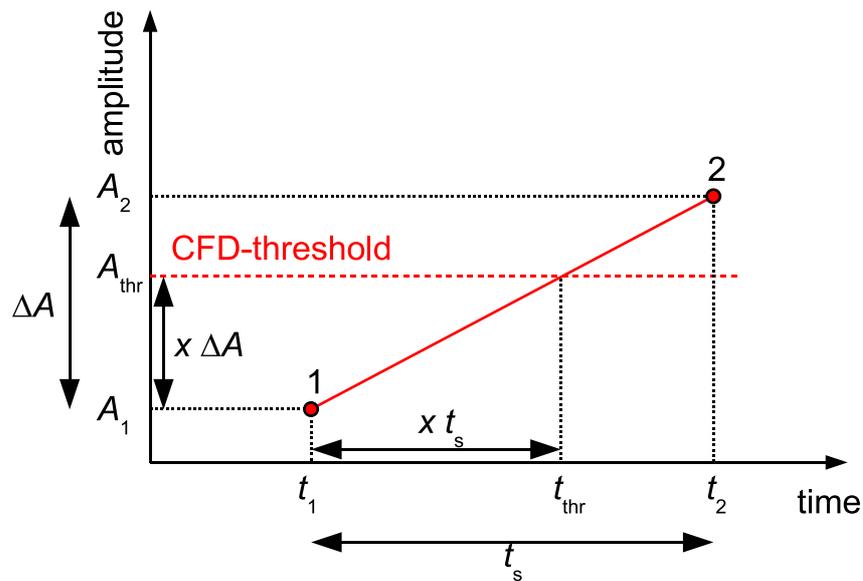


Figure 4.32: Principle of the digital constant fraction discrimination algorithm with linear interpolation between the two sampling points "1" and "2".

The CFD threshold value A_{thr} is first derived from the pulse amplitude A_{max} and the preset CFD fraction F_{CFD} by

$$(4.63) \quad A_{thr} = A_{max} \cdot F_{CFD} \quad \text{with} \quad F_{CFD} \in [0, 1].$$

In the next step, the last data sample A_1 below the CFD threshold value has to be determined by comparing the data with A_{thr} . The following data sample A_2 is then either equal to the threshold value or greater. The desired time information t_{thr} when the interpolated signal waveform crosses the CFD threshold, is now derived by simple geometric relations based on Figure 4.32. Due to the linear interpolation, A_{thr} and t_{thr} can be related in the following way:

$$(4.64) \quad \frac{A_{thr} - A_1}{A_2 - A_1} = \frac{t_{thr} - t_1}{t_s}.$$

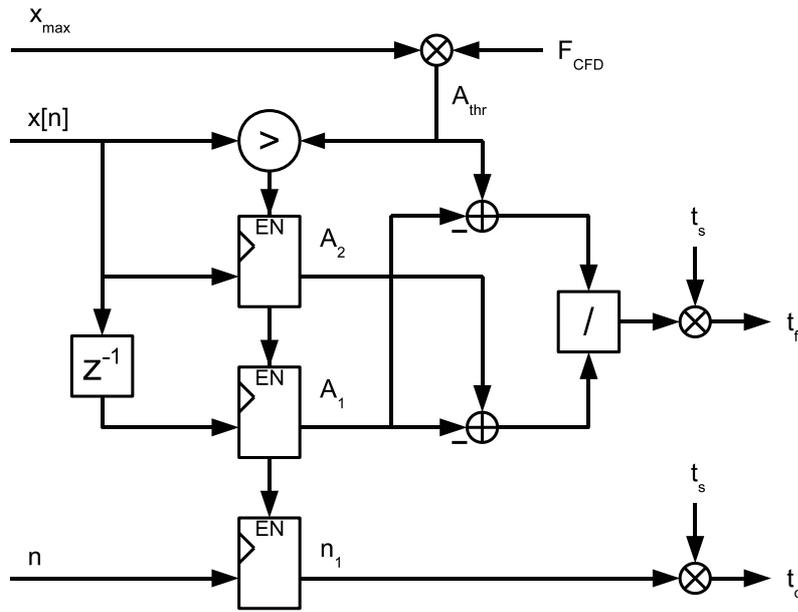


Figure 4.33: Schematic view of the CFD algorithm data flow.

The time values of the sample points are given by the corresponding sample index n which is incremented for each new sample point. This corresponds to the *coarse-time* information t_c , based on the ADC sampling rate t_s

$$(4.65) \quad t_c = n_1 \cdot t_s.$$

By solving (4.64) for $t_{thr} - t_1$, the *fine-time* information t_f can be calculated. This removes the dependence on the absolute time values of the sample points and just the sample interval remains as time variable.

$$(4.66) \quad t_f = t_{thr} - t_1 = \frac{A_{thr} - A_1}{A_2 - A_1} \cdot t_s.$$

The final time of the CFD threshold crossing is then given by the sum of both time values

$$(4.67) \quad t_{thr} = t_c + t_f.$$

Figure 4.33 shows the data flow for the digital constant fraction algorithm. The incoming data stream is continuously compared to the calculated threshold value. Once the threshold is exceeded, the A_1 and A_2 values are stored and the fine-time can be calculated. At the same time, the current sample index is stored as well, which allows to determine the

coarse time information. For the index value, it is not required to get exactly the corresponding index for the A_1 sample point. If the same offset is introduced in all channels, the relative time difference between different channels does not change. This allows to simplify the structure which detects the threshold position to a simple storage register.

From the implementation complexity, the algorithm is quite efficient. Once the signal amplitude is provided, the threshold calculation requires one multiplication to determine the threshold value and one compare operation per sample point until the first n_1 sample is found. The calculation of the t_f value involves two additions (subtractions) and a division operation. Since the multiplication with t_s is required for both t_c and t_f , only one multiplication is needed after the combination of both values. Since the sampling rate is usually fixed by the hardware setup, the t_s value can in principle take on any value, which allows once again to simplify the multiplications either by binary shift operations or remove them completely by setting $t_s = 1$. The calculated time values then have to be corrected later on, when a real time value, for example in units of ns, is needed.

4.4.2.1 CFD Performance Evaluation

To evaluate the performance of the digital CFD algorithm, similar simulations as for the amplitude estimation can be carried out. Therefore, the exemplary pulse shape from Section 4.1.1 is again sampled at varying sampling rates, to derive the optimal ADC clock frequency setting. Additionally, since the algorithm focuses on achieving a sub-sampling-interval time resolution, the generated pulse data includes also the uniformly distributed time offset ϕ according to (4.10) and (4.11). To achieve a constant output value for the given input parameters ϕ and t_s , the simulated CFD time estimation function is given by

$$(4.68) \quad \hat{t}_{\text{thr}} = (t_c + t_f) + \phi - t_s = \text{const.} \quad \forall \phi, t_s.$$

Figure 4.34 shows the simulation results for different signal to noise ratios. As (4.68) postulates a constant time value for varying sampling rates, the mean amplitude should settle at a constant value, which is determined by the pulse shape and the fraction setting F_{CFD} . For the actual simulation, the CFD fraction was set to $F_{\text{CFD}} = 40\%$ and the input pulse amplitude was normalized to one.

The plots show several features: First, the timing algorithm is obviously not working with sampling rates below about 30 MHz, as the standard deviation exhibits a steep slope. This can be explained by referring back to the time structure of the sample signal given in Figure 4.4. Since 30 MHz corresponds to a 33 ns spacing of the sample points, this agrees with the time position of the 40% trigger point on the rising signal edge. Therefore, for sampling rates below about 30 MHz, no sample point below the threshold value can be obtained from the pulse waveform. Instead, the zero baseline value is used in the

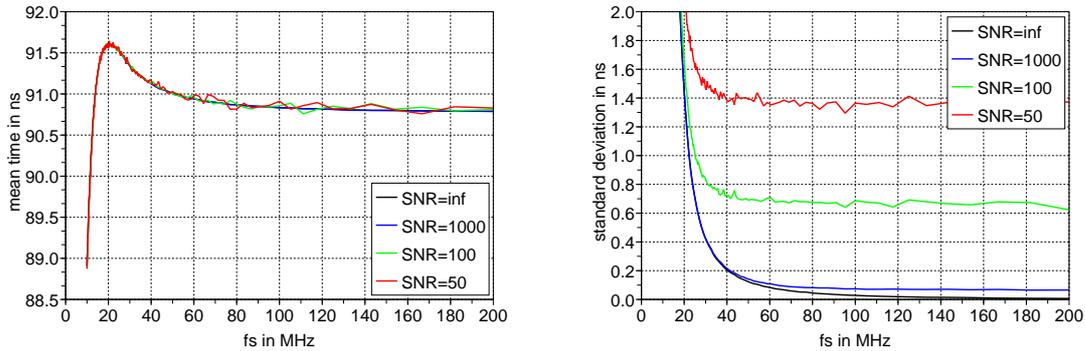


Figure 4.34: Simulation of the digital CFD algorithm with linear interpolation at varying sampling rates. Plotted is the mean of the reconstructed time (*left*) and the corresponding standard deviation (*right*) for $F_{CFD} = 40\%$.

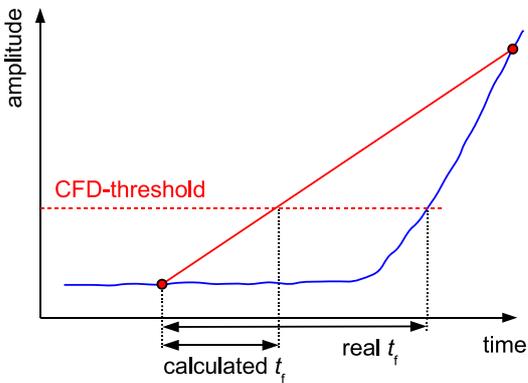


Figure 4.35: Effect of the interpolation with baseline values due to undersampling.

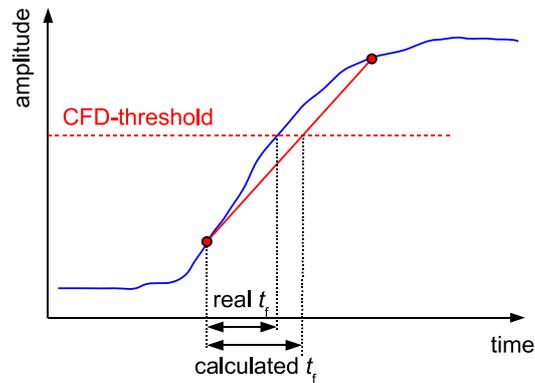


Figure 4.36: Rising edge linear interpolation mismatch due to undersampling.

t_f calculation, which shifts the reconstructed time to smaller values, as demonstrated in Figure 4.35.

For sampling rates between 30 MHz and about 60 MHz, two waveform sampling points are available for the calculation, but the linear interpolation does not yet agree with the shape of the rising edge comparable to Figure 4.36. Once the sampling rate increases, the linear interpolation more and more becomes an adequate approximation for the rising edge. This leads to a reduction of the estimated time value, which starts to settle around 90.8 ns for higher sampling rates. The signal to noise ratio has no significant effect on the mean value of the reconstructed time.

For the standard deviation, the black curve for the noiseless case only asymptotically reaches the zero value for increasing sampling rates. This is as well caused by the mismatch of the linear interpolation on the rising signal edge, but, since the absolute value is well below 100 ps, this causes only minor errors. Higher deviations are essentially caused by noisy input data. For the case of noisy data the standard deviation reaches a stable value already for sampling rates around 80 MHz. Therefore, this frequency can

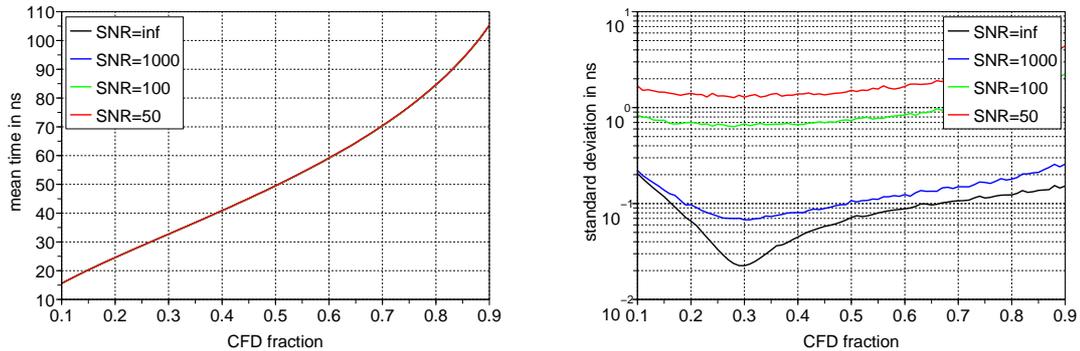


Figure 4.37: Simulation of the digital CFD algorithm with linear interpolation for varying fraction settings at $f_s = 80$ MHz. Plotted is the mean of the reconstructed time (*left*) and the corresponding standard deviation (*right*).

be assumed a good compromise between implementation effort and achievable timing performance.

For further optimization, the fraction value F_{CFD} can be varied to find the point with minimal time errors. Theoretically, this should be around the steepest slope on the rising edge, as changes in amplitude translate to the smallest impact on the reconstructed time value at this position. Figure 4.37 shows the mean reconstructed time and its standard deviation for different fraction settings with input data sampled at a fixed rate of 80 MHz. As expected, the mean reconstructed time is continuously increasing as the fraction approaches 100% and moves along the rising edge of the pulse. Additional noise causes no significant degradation of the mean time.

The standard deviation indicates an optimal F_{CFD} setting of about 29% for the given pulse shape and sampling rate. For smaller fraction values, the undersampling effect from Figure 4.35 arises again and rapidly increases the error. For higher fractions, the steep slope of the rising edge slowly vanishes which leads to a steady increase of the error until the signal maximum is reached.

Apart from the “by thumb” explanation of the CFD performance, it can also be addressed by the underlying mathematical properties of the analyzed signal. Finally, a value for the optimum CFD fraction setting can be obtained.

4.4.2.2 Determination of the CFD Fraction Settings

The underlying problem—which necessitates an optimal CFD fraction setting—is the estimation of the arrival time of the sampled detector signal. As already used in the previous sections, the input pulse $h(t)$ is modeled by a sum of three exponentials, corresponding to the three time constants in the analog signal chain (4.6):

$$h(t) = \alpha \cdot e^{-t/\tau_{\text{HP}}} + \beta \cdot e^{-t/\tau_{\text{LP}}} + \gamma \cdot e^{-t/\tau_{\text{LSO}}}.$$

Since the algorithm operates on sampled input data, a discrete representation can be obtained by using (4.10):

$$(4.69) \quad h[n; \phi] = \alpha \cdot e^{-(n \cdot t_s + \phi) / \tau_{HP}} + \beta \cdot e^{-(n \cdot t_s + \phi) / \tau_{LP}} + \gamma \cdot e^{-(n \cdot t_s + \phi) / \tau_{LSO}}.$$

For the time estimation problem, ϕ is now the free parameter that has to be determined, whereas the τ_i and t_s are constant values which are determined by the circuit setup. It should be remembered that the remaining variables α , β and γ were only introduced to simplify the appearance of Equation (4.5), and are as well only depending on the τ_i . As the ideal sampled pulse shape from (4.69) will in reality be distorted by noise, the sampled data from the ADC have to include also an additive term modeling the white Gaussian noise:

$$(4.70) \quad x[n] = h[n] + w[n] \quad \text{with} \quad w[n] \sim \mathcal{N}(0, \sigma^2).$$

According to the specific τ_i values, the correlation between the free parameter ϕ and the pulse shape $h(t)$ is either stronger or weaker. Therefore, a measure for the information content of each sample value about the free parameter can be formulated [Kay93]. This is given by the *Fisher information*¹⁰, based on the probability density function $p(\mathbf{x}; \phi)$ of the analyzed sample values $\mathbf{x} = \{x[0], x[1], \dots, x[N-1]\}$

$$(4.71) \quad I(\phi) = -\mathbb{E} \left\{ \frac{\partial^2 \ln p(\mathbf{x}; \phi)}{\partial \phi^2} \right\}.$$

Thereby, the expression $\mathbb{E} \{ \bullet \}$ is the expectation value taken with respect to the probability density function (PDF) of the sampled data. Since every data sample is independently disturbed by additive Gaussian noise, the PDF of the full sampled pulse data vector is the product of the independent distributions of all samples

$$(4.72) \quad \begin{aligned} p(\mathbf{x}; \phi) &= \prod_{n=0}^{N-1} \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp \left(-\frac{1}{2\sigma^2} \cdot (x[n] - h[n; \phi])^2 \right) \\ &= \frac{1}{(2\pi\sigma^2)^{N/2}} \cdot \exp \left(-\frac{1}{2\sigma^2} \sum_{n=0}^{N-1} (x[n] - h[n; \phi])^2 \right). \end{aligned}$$

Due to its dependence on the parameter ϕ , $p(\mathbf{x}; \phi)$ is also termed the *likelihood function*. Taking the natural logarithm and deriving twice to the free parameter ϕ , gives the argument for the expectation operator in (4.71)

¹⁰Named after Sir Ronald Aylmer Fisher (1890–1962).

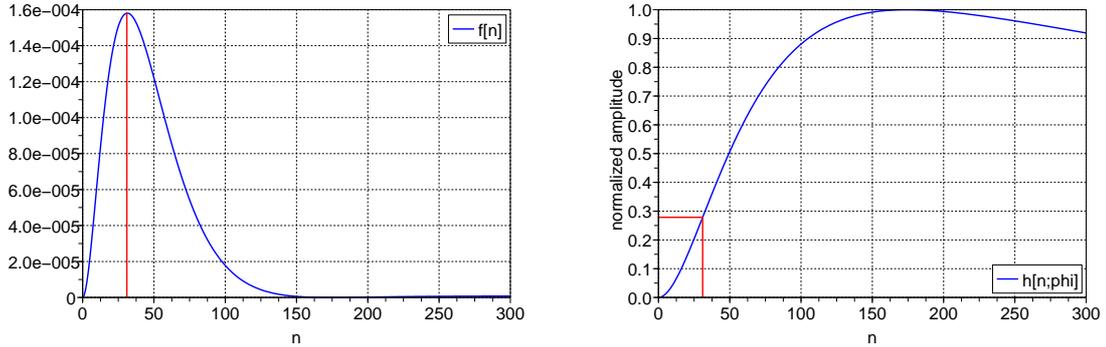


Figure 4.38: Calculation of the Fisher information for individual samples of $h[n; \phi]$ (left). The peak position with maximum information content (red) maps to the optimal fraction setting of the CFD algorithm, when projected on the normalized pulse shape of $h[n; \phi]$ (right). The sample number n corresponds to the pulse time in nanoseconds ($t_s = 1$ ns).

$$(4.73) \quad \frac{\partial^2 \ln p(\mathbf{x}; \phi)}{\partial \phi^2} = \frac{1}{\sigma^2} \sum_{n=0}^{N-1} \left((x[n] - h[n; \phi]) \cdot \frac{\partial^2 h[n; \phi]}{\partial \phi^2} - \left(\frac{\partial h[n; \phi]}{\partial \phi} \right)^2 \right).$$

By taking the negative expectation value of (4.73) the Fisher information for the pulse parameterization is obtained. As the expected value of $x[n]$ is the noiseless signal $h[n; \phi]$, the expectation operator forces $x[n] = h[n; \phi]$. Therefore, the expression $(x[n] - h[n; \phi])$ vanishes and the Fisher information simplifies to

$$(4.74) \quad I(\phi) = -E \left\{ \frac{\partial^2 \ln p(\mathbf{x}; \phi)}{\partial \phi^2} \right\} = \frac{1}{\sigma^2} \sum_{n=0}^{N-1} \underbrace{\left(\frac{\partial h[n; \phi]}{\partial \phi} \right)^2}_{f[n]},$$

with the derivative of $h[n; \phi]$ having the form

$$(4.75) \quad \frac{\partial h[n; \phi]}{\partial \phi} = \frac{\alpha}{\tau_{HP}} \cdot e^{-(n \cdot t_s + \phi)/\tau_{HP}} + \frac{\beta}{\tau_{LP}} \cdot e^{-(n \cdot t_s + \phi)/\tau_{LP}} + \frac{\gamma}{\tau_{LSO}} \cdot e^{-(n \cdot t_s + \phi)/\tau_{LSO}}.$$

Figure 4.38 shows a calculation of the squared sum argument $f[n]$ from (4.74), for $t_s = 1$ ns. This can be interpreted as the information content of the single sample points of $h[n; \phi]$ about the time offset ϕ . It is evident, that the major information is carried by the rising edge of the pulse, with its maximum at the steepest slope, as already assumed previously. The maximum position of the left plot in Figure 4.38 allows now to determine the optimal fraction setting for the CFD algorithm. Therefore, the normalized pulse shape $h[n; \phi]$ is evaluated in the right plot of Figure 4.38 at the same n_{\max} -value as the

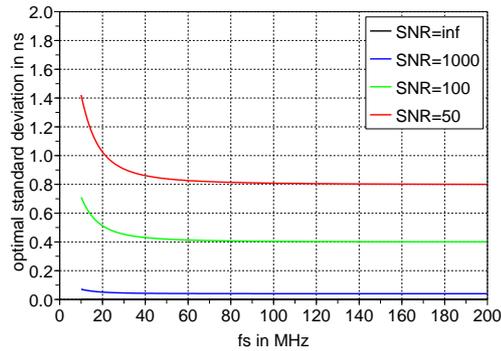


Figure 4.39: Calculation of the Cramér-Rao lower bound for the CFD algorithm at different sampling rates.

maximum of the Fisher information. This yields a value of $F_{\text{CFD}} \approx 28\%$ which is in perfect agreement with the value found in Figure 4.37.

Since the CFD algorithm uses two sampling points for the interpolation, the available Fisher information is the sum of both independent information values. By placing the CFD fraction at the maximum value, it is guaranteed that the total information is maximized as the position of the sample values is bound to $(n_{\text{max}} - 1) < n_{\text{max}} < (n_{\text{max}} + 1)$.

Based on the Fisher information $I(\phi)$, the *Cramér-Rao lower bound* can be determined, which states a lower bound for the variance of the estimated parameter. This allows to compare the algorithm performance with the best achievable physical limit. The Cramér-Rao lower bound (CRLB) is given by

$$(4.76) \quad \text{var}(\hat{\phi}) \geq \frac{1}{I(\phi)} = \frac{\sigma^2}{\sum_{n=0}^{N-1} \left(\frac{\partial h[n; \phi]}{\partial \phi} \right)^2}.$$

For the CFD algorithm, the total number of samples used is $N = 2$. The two sampling points are referenced by the two index values n_1 and n_2 which are spaced by the sampling interval t_s . Therefore, $n_2 = n_1 + 1$. The expression which has to be evaluated for the CRLB is thus

$$(4.77) \quad \text{var}(\hat{\phi}) \geq \frac{\sigma^2}{\sum_{n=n_1}^{n_1+1} \left(\frac{\partial h[n; \phi]}{\partial \phi} \right)^2}.$$

Figure 4.39 shows a calculation of the best achievable CRLB for different sampling frequencies. To be able to compare with the previous simulations of the standard deviation, the square root of the CRLB is plotted. For an infinite signal to noise ratio, the σ of the

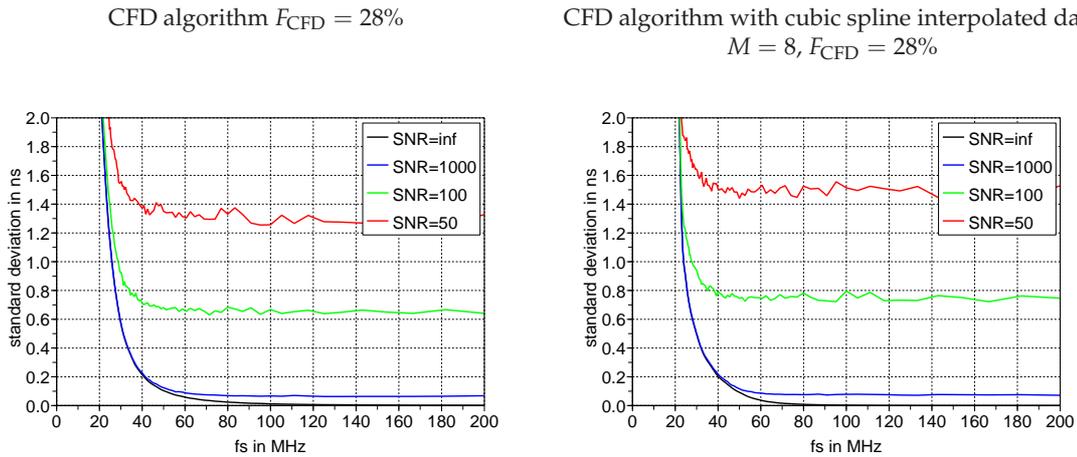


Figure 4.40: Comparison of the CFD algorithm performance, simulated on sampled data (*left*) and spline interpolated data with an interpolation factor of $M = 8$ (*right*).

Gaussian noise is zero. Therefore, the CRLB attains also the ideal zero value. For decreased SNR values, the CRLB is increased, but reaches stable values, once the sampling rate reaches 80–100 MHz. When compared with the simulation in Figure 4.34, the simulated standard deviation is about 1.7-times higher than the CRLB.

4.4.2.3 CFD with Interpolated Input Data

Since some options for the amplitude estimation include a more advanced interpolation scheme of the input samples, the CFD algorithm may also operate on the interpolated data samples. By creating an interpolated data stream with an upsampling factor of M , the sampling interval t_s for the CFD algorithm is as well scaled by M . This requires on one hand more operations to find the first threshold value, on the other hand, the linear interpolation in the CFD algorithm is only required for an M -fold smaller interval, which should match to the pulse shape in a better way.

Figure 4.40 compares simulation results for the CFD algorithm, preceded by a spline interpolation with $M = 8$ (*right*) with the performance on the original data (*left*). The fraction setting is for both cases at $F_{CFD} = 28\%$. As the signal to noise ratio decreases, the standard deviation of the reconstructed time is slightly higher for the interpolated data. This is due to the fact that the spline interpolation operates on the noisy data samples, with the additional noise causing larger changes on the interpolated pulse shape than the linear CFD algorithm. Hence, the exact spline based interpolation gives no advantage over the simple CFD algorithm.

Figures 4.41 and 4.42 show the results for the same simulations including a smoothing spline data preprocessing. To find the optimal value for the smoothing parameter λ , Figure 4.41 evaluates the standard deviation of the reconstructed time for different λ values and interpolation factors M . It can be seen, that the achieved standard deviation stabilizes for $\lambda > 2$. For high signal to noise ratios, an increment in the standard deviation

4 DATA ACQUISITION BASED ON SAMPLING ADCS

$$M = 1, F_{\text{CFD}} = 28\%, f_s = 80 \text{ MHz}$$

$$M = 2, F_{\text{CFD}} = 28\%, f_s = 80 \text{ MHz}$$

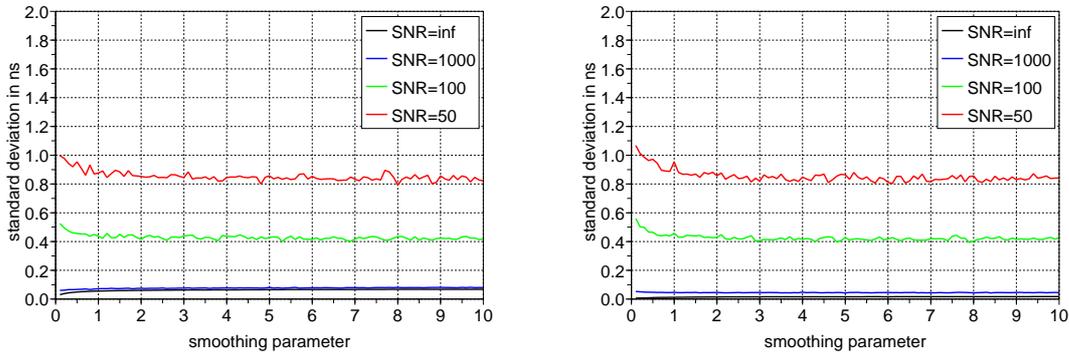


Figure 4.41: Smoothing parameter (λ) scans for the CFD algorithm with smoothing spline interpolated data at different SNR values. The used interpolation factors are $M = 1$ (left) and $M = 2$ (right).

CFD algorithm with smoothing cubic spline interpolated data $M = 2, \lambda = 3, F_{\text{CFD}} = 28\%$

CFD algorithm with smoothing cubic spline interpolated data $M = 8, \lambda = 3, F_{\text{CFD}} = 28\%$

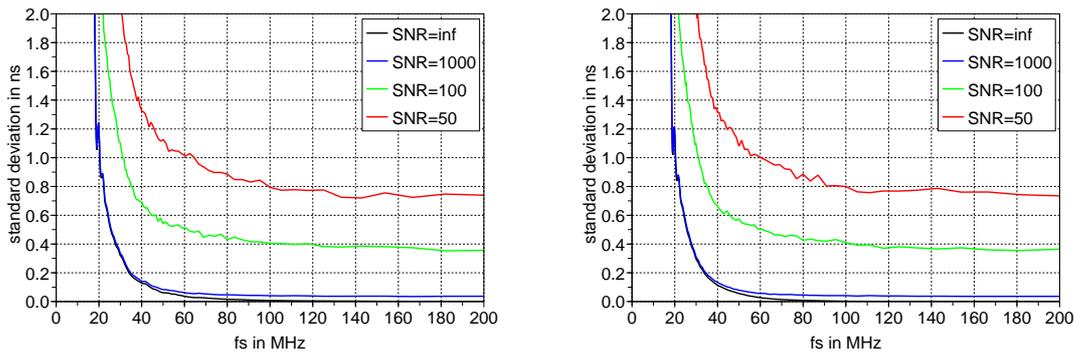


Figure 4.42: Comparison of the CFD algorithm performance, simulated on smoothing cubic spline interpolated data with $M = 2, \lambda = 3$ (left) and $M = 8, \lambda = 3$ (right).

is visible for increasing λ values, which is caused by the smoothing of the noiseless pulse shape. For lower SNR, this effect is compensated by the improvement due to the reduced noise on the smoothed data.

Figure 4.42 shows the CFD performance for different sampling rates and spline interpolation factors. The smoothing parameter is thereby fixed at $\lambda = 3$. The standard deviation indicates that the performance of the algorithm decreases significantly for sampling rates below 60 MHz. From 100 MHz upwards, the standard deviation reaches almost constant values comparable to the CRLB in Figure 4.39 for the CFD with two sampling points. The fact that the standard deviation becomes lower than the calculated CRLB for high sampling rates can be explained since the CRLB in Figure 4.39 is calculated from the original pulse model. Due to the smoothing operation—which takes into account multiple signal samples at once—the pulse shape is modified, which also leads to different CRLB values. Similarly, the smoothing spline preconditioning of the input data can be interpreted

based on the spline recursion equations (4.58) as an algorithm which uses the information from five input data values to derive the smoothed output. The smoothed CFD implementation thus is based on the input of in total six data values, which would lead to a lower CRLB than given in Figure 4.39.

4.4.3 Least Squares Timing

Another method for the determination of the pulse arrival time can be based on the least square method [Kay93]. Therefore, the incoming data $x[n]$ are compared with predefined pulse images $h[n; \phi]$, which correspond to a known time offset ϕ . By minimizing the resulting error, the best matching value for the time offset can be found. The least squares error function is given by

$$(4.78) \quad J(\phi) = \sum_{n=0}^{N-1} (x[n] - h[n; \phi])^2.$$

The optimal value for the parameter ϕ is then found by minimizing $J(\phi)$ for a given sampled data set. This may be achieved in two different ways: For the grid search method, the input data are explicitly tested against all possible pulse images, arising from variations of the ϕ parameter. Although this “brute-force” approach requires a significant number of operations—depending on the required search range for the parameter—it may be suitable for the implementation in highly parallel FPGA devices. The second method treats the least squares problem analytically and has to derive a calculation procedure which determines the unknown parameter ϕ , based on the input data.

It is useful to describe the linear least squares problem in matrix notation. Therefore, the signal model vector is given by $\mathbf{h} = [h[0, \phi] \ h[1, \phi] \ \cdots \ h[N-1, \phi]]^T$ and can be determined from the parameter vector $\boldsymbol{\phi} = [\phi_0 \ \phi_1 \ \cdots \ \phi_{P-1}]^T$ by the matrix multiplication $\mathbf{h} = \mathbf{H}\boldsymbol{\phi}$. The *observation matrix* \mathbf{H} thus represents the signal model which can be parameterized by $\boldsymbol{\phi}$. The least squares error function (4.78) can then be written as

$$(4.79) \quad \begin{aligned} J(\boldsymbol{\phi}) &= (\mathbf{x} - \mathbf{h})^T (\mathbf{x} - \mathbf{h}) \\ &= (\mathbf{x} - \mathbf{H}\boldsymbol{\phi})^T (\mathbf{x} - \mathbf{H}\boldsymbol{\phi}). \end{aligned}$$

To minimize $J(\boldsymbol{\phi})$, the product in (4.79) is evaluated and derived with respect to $\boldsymbol{\phi}$. This gives

$$(4.80) \quad \frac{\partial J(\boldsymbol{\phi})}{\partial \boldsymbol{\phi}} = -2\mathbf{H}^T \mathbf{x} + 2\mathbf{H}^T \mathbf{H} \boldsymbol{\phi}.$$

By setting this to zero, the parameter vector can be determined

$$(4.81) \quad \hat{\boldsymbol{\phi}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x}.$$

As the full rank P of the matrix \mathbf{H} is guaranteed—otherwise parameters could be removed from the model until full rank is achieved—it is also guaranteed that $\mathbf{H}^T \mathbf{H}$ can be inverted. The expression $(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T$ is a $P \times N$ matrix which can be precomputed. The calculation of each element of the parameter vector $\hat{\boldsymbol{\phi}}$ then simplifies to a product of the matrix line with the input data vector.

4.4.3.1 Grid Search

With the matrix based least squares error function in (4.79), only linear signal models are tractable. Unfortunately, for the signal model of the sampled detector pulse (4.69) the unknown time offset parameter is located in the exponent, therefore describing a nonlinear least squares problem. However, the model vector \mathbf{h} can always be precalculated, based on the signal model and a chosen parameter value. For the grid search method, several model vectors are determined in order to scan the desired parameter space. The minimum least squares error can then be found, by evaluating $J(\boldsymbol{\phi})$ for all available model vectors \mathbf{h} .

Since the continuity condition of the square operation is not required for evaluating the error function in this way, the squared error (4.78) can also be simplified, e.g. by using a *sum of absolute differences* (SAD), which allows an easier hardware implementation without multiplications

$$(4.82) \quad J(\phi) = \sum_{n=0}^{N-1} |x[n] - h[n; \phi]|.$$

Figure 4.43 gives a schematic overview of the grid search method. The input data $x[n]$ are stored in a shift register and successively compared to the pulse model data. By identifying the minimum error value, the corresponding index of the pulse model gives the unknown ϕ parameter. Since the amplitude of the input data changes with the energy of the detected photon, the pulse model must be adapted to the current amplitude x_{\max} of the data. Hence, only normalized pulse model data are stored and then multiplied with the pulse amplitude, which is determined beforehand.

The grid search can be applied as well, if the position of the pulse within the data samples is unknown. Therefore, the data samples are shifted and the comparison with the pulse model data is repeatedly evaluated for all possible pulse positions. The minimum error

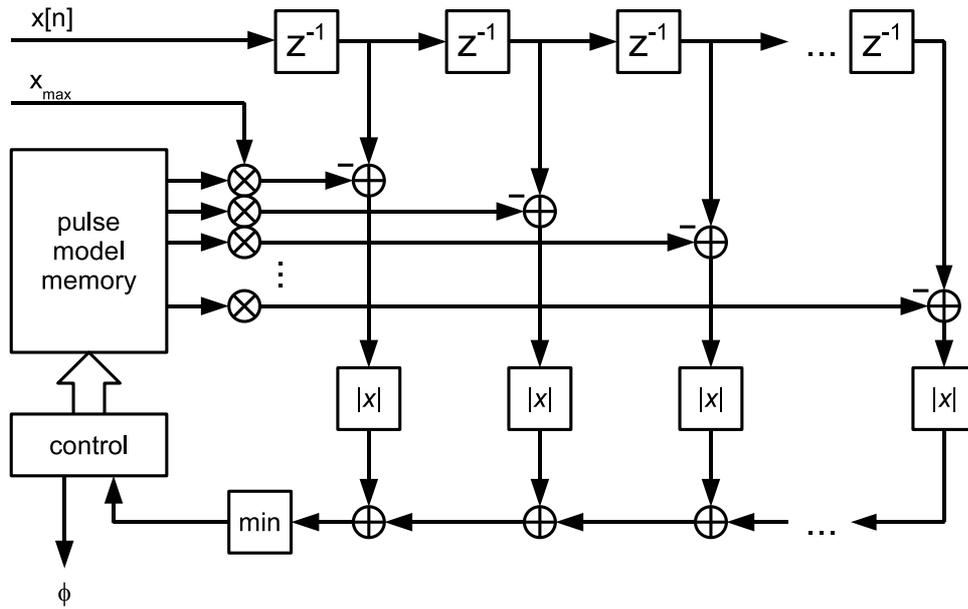


Figure 4.43: Schematic view of the least squares grid search method.

then indicates the ϕ value and as well the position of the shifted pulse data, similar to the fine- and coarse-time values from the CFD algorithm in Section 4.4.2.

The grid search procedure can be parameterized by two parameters, namely the number of evaluated sampling points N_s in the error function (4.82) and the number of different pulse model realizations N_ϕ . The best achievable time resolution is then determined by

$$(4.83) \quad t_{\min} = \frac{t_s}{N_\phi}.$$

Figure 4.44 shows the simulation results for the grid search method. Again, no useful results can be obtained for sampling rates below 20 MHz as not enough information is available. With increasing frequency, the standard deviation continuously decreases. When compared to the constant fraction algorithm in Figure 4.34 and 4.40, it can be seen that the grid search can not reach the low standard deviation values for the noiseless case, which is due to the minimum achievable time resolution (4.83), limited by N_ϕ . However, for lower signal to noise ratios, the grid search gains from the fact that multiple input data samples contribute to the determination of the time value. Therefore, the performance is superior to the CFD algorithm for sampling rates above 40 MHz (cf. Figure 4.42). Nevertheless, the CFD algorithm can be improved by preprocessing the data with a smoothing spline interpolation. As the smoothing operation affects also the pulse shape, it can not be applied directly to the grid search method since this would cause a mismatch between the input data and the stored pulse model.

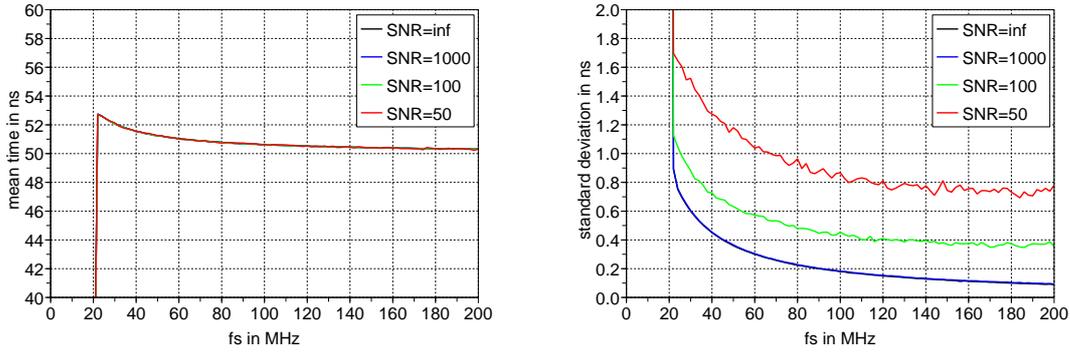


Figure 4.44: Simulation of the grid search method at varying sampling rates. Plotted is the mean of the reconstructed time (*left*) and the corresponding standard deviation (*right*) for $N_s = 10$ and $N_\phi = 16$.

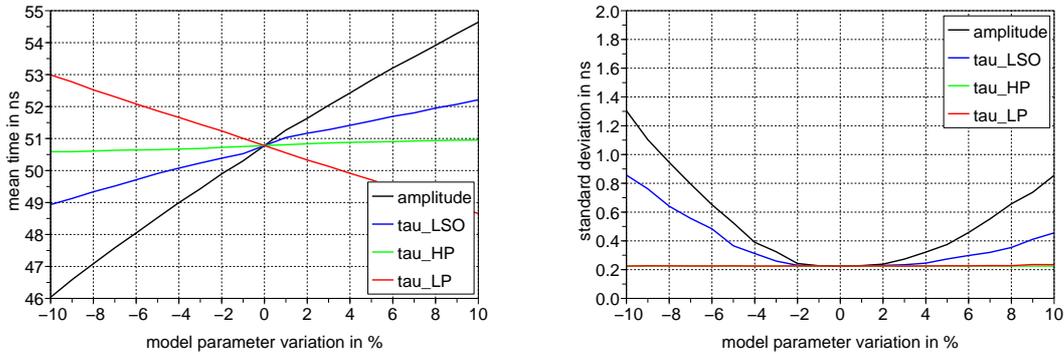


Figure 4.45: Simulation of the sensitivity to model parameter variations for the grid search method. Plotted is the mean of the reconstructed time (*left*) and the corresponding standard deviation (*right*) for $N_s = 10$, $N_\phi = 16$ and $f_s = 80$ MHz.

Since the grid search is based on the knowledge of the expected pulse shape of the input signals, either a mathematical description which allows to calculate the different pulse models, or an extracted pulse model from the sampled input data is required. Figure 4.45 shows the dependence of the reconstructed time on variations of the four parameters of the mathematical pulse model in (4.69). The two most sensitive parameters are the pulse amplitude and the decay time of the LSO scintillator, whereas the rise and fall time of the preamplifier have only minor influence. Since the properties of the LSO scintillator are influenced by the production process, the decay time may vary between detectors assembled from different scintillator lots. This may require an additional calibration/optimization procedure for the real detector setup, if the scintillator parameters differ significantly.

To find the optimal settings for the N_s and N_ϕ parameters, Figure 4.46 shows the simulation results for the corresponding parameter scan. Comparable to the CFD algorithm, two sample points are already sufficient to reach the optimal result for noiseless signals. For noisy signals, no significant improvement is visible for more than 10 sampling

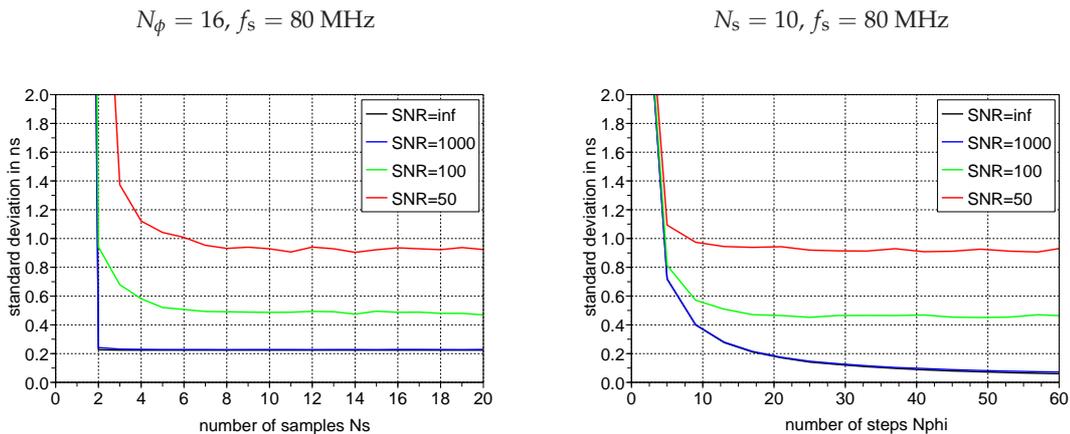


Figure 4.46: Parameter scan simulations for the number of samples N_s (left) and the number of pulse model realizations N_ϕ (right). Plotted is the standard deviation of the reconstructed time value.

points. This corresponds to the duration of the rising signal edge, which is in the order of $10 \cdot (1/80 \text{ MHz}) = 125 \text{ ns}$ and, as already shown in Section 4.4.2.2, carries most of the information about the start time of the pulse.

Since the number of evaluated pulse model realizations determines the achievable time resolution given in (4.83), the standard deviation decreases with an increased number of realizations. For noisy input data, a stable result is obtained for values greater than ~ 20 . However, for a finer sampling of the pulse model grid, more computational effort is required to find the start time of the input pulse.

For the calculation of one error value, the grid search method requires N_s multiplications and absolute value operations, and $2 \cdot N_s$ additions. For the evaluation of all stored pulse model data, this requirement is scaled by N_ϕ . If, in addition, also the pulse position within the input data set is unknown, the search procedure has to be repeated multiple times until the global minimum is found. More advanced realizations can also include a prescan of all possible pulse positions with just one model realization to first determine a coarse position of the error minimum. The following fine-grained scan for the precise ϕ parameter can then be confined to a smaller region of the input data, which can reduce the total number of required operations.

4.4.3.2 Analytic Least Squares Method

Although the sampled pulse model (4.69) has a nonlinear dependence on the time offset parameter ϕ , an analytical solution of the least squares problem is still possible. Therefore, the nonlinear problem of minimizing $J(\phi)$ can be transformed into a linear least squares problem $J(\theta)$, followed by a back-transformation of the obtained linear parameters into the nonlinear parameter space. This allows to obtain the unknown time offset directly from the input dataset without the high computational effort of the grid search method.

To be able to convert the nonlinear parameters $\boldsymbol{\phi}$ into a linear parameter vector $\boldsymbol{\theta}$, an invertible transformation function has to be found. For the general case of a P -dimensional parameter vector the transformation becomes as well a P -dimensional function

$$(4.84) \quad \boldsymbol{\theta} = \mathbf{g}(\boldsymbol{\phi}),$$

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_{P-1} \end{bmatrix} = \begin{bmatrix} g_0(\phi_0) \\ g_1(\phi_1) \\ \vdots \\ g_{P-1}(\phi_{P-1}) \end{bmatrix}.$$

This allows to formulate the linear least squares problem as $\mathbf{h}_{\text{lin}} = \mathbf{H}_{\text{lin}}\boldsymbol{\theta}$. By the inverse transformation of (4.84), the linear solution vector $\hat{\boldsymbol{\theta}}$ can be transformed back into the desired nonlinear solution

$$(4.85) \quad \hat{\boldsymbol{\phi}} = \mathbf{g}^{-1}(\hat{\boldsymbol{\theta}}).$$

The analytical least squares method requires again a mathematical description of the expected pulse shape, which was already given by (4.6). For the sampled data input, the discrete form is given by (4.69) with ϕ as the unknown time offset parameter. By separating the exp-functions with ϕ in the exponent, the parameters for the linear least squares model can be derived. For simplification of the least squares calculation, the α , β and γ constants are included in the θ_i parameters as well

$$(4.86) \quad h[n; \phi] = \alpha \cdot e^{-(n \cdot t_s + \phi) / \tau_{\text{HP}}} + \beta \cdot e^{-(n \cdot t_s + \phi) / \tau_{\text{LP}}} + \gamma \cdot e^{-(n \cdot t_s + \phi) / \tau_{\text{LSO}}}$$

$$= e^{-n \cdot t_s / \tau_{\text{HP}}} \cdot \underbrace{\alpha \cdot e^{-\phi / \tau_{\text{HP}}}}_{\theta_0} + e^{-n \cdot t_s / \tau_{\text{LP}}} \cdot \underbrace{\beta \cdot e^{-\phi / \tau_{\text{LP}}}}_{\theta_1} + e^{-n \cdot t_s / \tau_{\text{LSO}}} \cdot \underbrace{\gamma \cdot e^{-\phi / \tau_{\text{LSO}}}}_{\theta_2}.$$

The linear least squares problem is then given by $\mathbf{h}_{\text{lin}} = \mathbf{H}_{\text{lin}}\boldsymbol{\theta}$

$$(4.87) \quad \mathbf{h}_{\text{lin}} = \begin{bmatrix} 1 & 1 & 1 \\ e^{-\frac{1 \cdot t_s}{\tau_{\text{HP}}}} & e^{-\frac{1 \cdot t_s}{\tau_{\text{LP}}}} & e^{-\frac{1 \cdot t_s}{\tau_{\text{LSO}}}} \\ e^{-\frac{2 \cdot t_s}{\tau_{\text{HP}}}} & e^{-\frac{2 \cdot t_s}{\tau_{\text{LP}}}} & e^{-\frac{2 \cdot t_s}{\tau_{\text{LSO}}}} \\ \vdots & \vdots & \vdots \\ e^{-\frac{(N-1) \cdot t_s}{\tau_{\text{HP}}}} & e^{-\frac{(N-1) \cdot t_s}{\tau_{\text{LP}}}} & e^{-\frac{(N-1) \cdot t_s}{\tau_{\text{LSO}}}} \end{bmatrix} \cdot \begin{bmatrix} \alpha \cdot e^{-\phi/\tau_{\text{HP}}} \\ \beta \cdot e^{-\phi/\tau_{\text{LP}}} \\ \gamma \cdot e^{-\phi/\tau_{\text{LSO}}} \end{bmatrix}.$$

The solution of this problem is determined by using (4.81), where the expression $(\mathbf{H}_{\text{lin}}^T \mathbf{H}_{\text{lin}})^{-1} \mathbf{H}_{\text{lin}}^T$ can be calculated beforehand and stored as a constant matrix. Once the linear parameters are obtained, they have to be transformed back into the nonlinear parameter space via the inverse transformation (4.85). Since the transformation into the linear least squares model required to split the single ϕ parameter into three linear parameters $\theta_{0,1,2}$, the three results have to be combined again. For further simplification, the amplitude of the signal model may be split from the three constants α , β and γ , introducing the normalized constants α' , β' and γ' :

$$(4.88) \quad \begin{aligned} \alpha &= A \cdot \alpha', \\ \beta &= A \cdot \beta', \\ \gamma &= A \cdot \gamma'. \end{aligned}$$

The three equations for the back-transformation then become

$$(4.89) \quad \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} A \cdot \alpha' \cdot e^{-\phi/\tau_{\text{HP}}} \\ A \cdot \beta' \cdot e^{-\phi/\tau_{\text{LP}}} \\ A \cdot \gamma' \cdot e^{-\phi/\tau_{\text{LSO}}} \end{bmatrix}.$$

By e.g. multiplying the three equations, the ϕ parameter can be isolated and extracted from the exponent by taking the natural logarithm

$$(4.90) \quad \begin{aligned} \theta_0 \cdot \theta_1 \cdot \theta_2 &= A^3 \cdot \alpha' \beta' \gamma' \cdot e^{\phi \cdot \left(-\frac{1}{\tau_{\text{HP}}} - \frac{1}{\tau_{\text{LP}}} - \frac{1}{\tau_{\text{LSO}}} \right)}, \\ \phi &= -\frac{\tau_{\text{HP}} \tau_{\text{LP}} \tau_{\text{LSO}} \cdot \ln \left(\frac{\theta_0 \cdot \theta_1 \cdot \theta_2}{A^3 \cdot \alpha' \cdot \beta' \cdot \gamma'} \right)}{(\tau_{\text{LP}} + \tau_{\text{HP}}) \cdot \tau_{\text{LSO}} + \tau_{\text{HP}} \tau_{\text{LP}}}. \end{aligned}$$

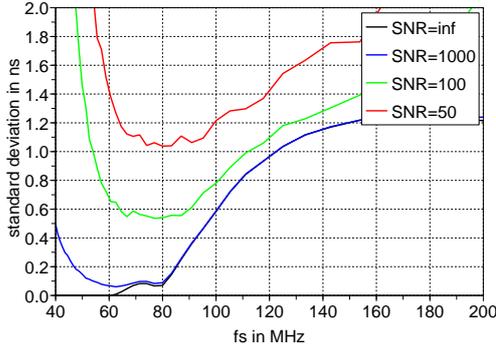


Figure 4.47: Simulation of the analytic least squares method at varying sampling rates for $N_s = 10$.

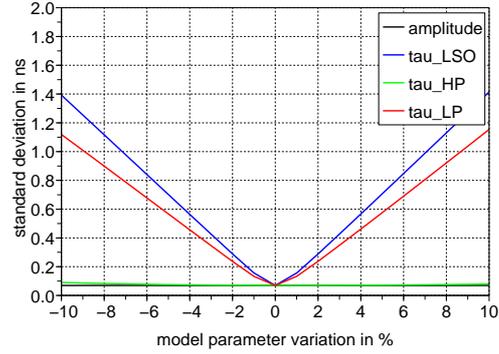


Figure 4.48: Simulation of the sensitivity to model parameter variations for the analytic least squares method. The settings are $N_s = 10$ and $f_s = 80$ MHz.

The problem with (4.90) is its amplitude dependency. Therefore, errors in the amplitude determination propagate also to the calculated time. By a different combination of the back-transformation equations (4.89), the amplitude dependence can be removed

$$\frac{\theta_0 \cdot \theta_1}{\theta_2^2} = \frac{\alpha' \beta'}{\gamma'^2} \cdot e^{\phi \cdot \left(-\frac{1}{\tau_{HP}} - \frac{1}{\tau_{LP}} + \frac{2}{\tau_{LSO}}\right)},$$

$$(4.91) \quad \phi = -\frac{\tau_{HP} \tau_{LP} \tau_{LSO} \cdot \ln\left(\frac{\theta_0 \cdot \theta_1 \cdot \gamma'^2}{\alpha' \cdot \beta' \cdot \theta_2^2}\right)}{(\tau_{LP} + \tau_{HP}) \cdot \tau_{LSO} + 2 \cdot \tau_{HP} \tau_{LP}}.$$

Similar to the grid search approach, the analytical least squares method is parameterized by the number of evaluated sample points N_s . Since the calculation involves no discretization of the unknown ϕ parameter, there is no algorithm based lower limit for the time resolution. However, noisy input data introduces errors and will therefore limit the achievable time resolution.

Figure 4.47 shows the simulation results for the analytical least squares algorithm. It can be seen that the optimal performance is only reached for sampling frequencies between 60 and 80 MHz. For this region, the input signal starts at the same index, as expected by the linear model. For higher sampling rates, the signal offset is increased, since more sampling points are obtained before the pulse. Since the pulse then does not match the assumed start index of the model, the error on the resulting time value increases. If the pulse position is unknown, the algorithm requires additional effort to align the model before the evaluation. Once the input data and the model are aligned, the achievable performance is comparable to the grid search method.

The sensitivity to model parameter variations is shown in Figure 4.48. In contrast to the grid search, the dependence on the signal amplitude is completely removed by using

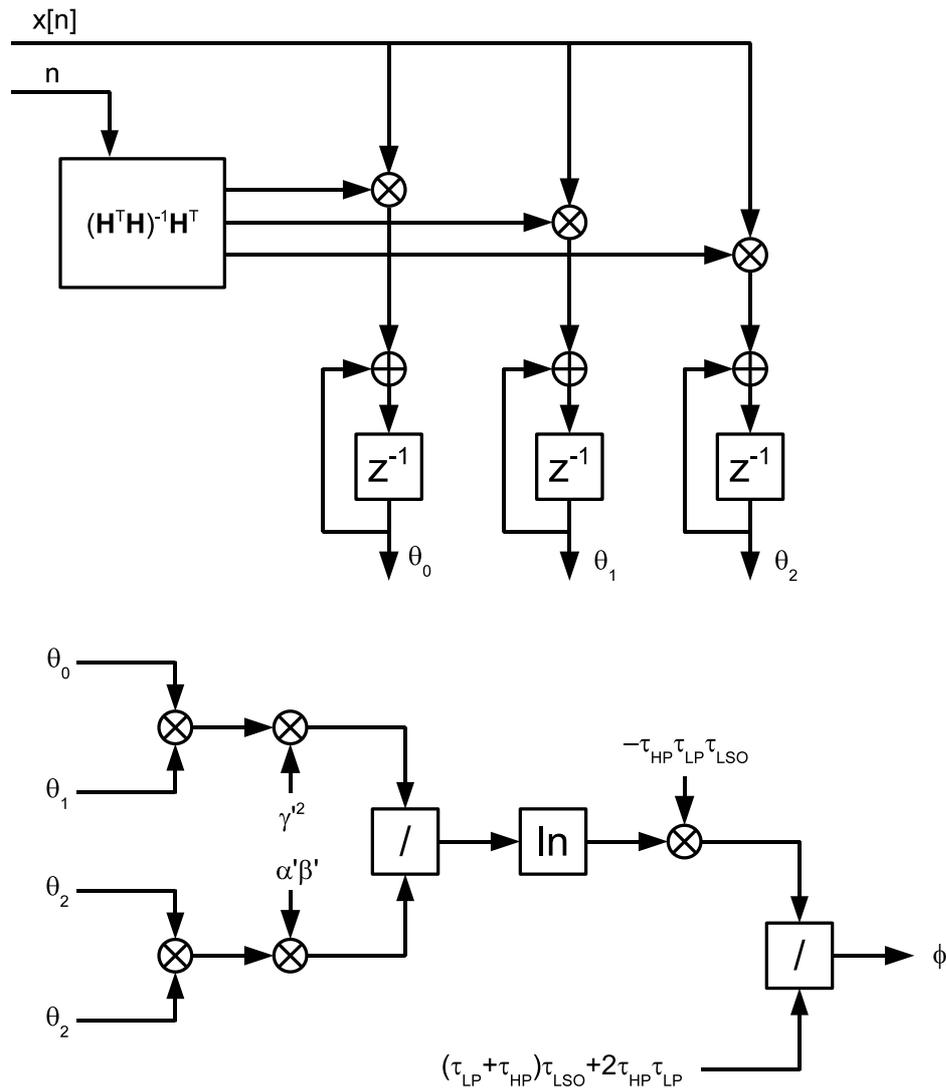


Figure 4.49: Schematic view of the analytic least squares method.

the back-transformation in (4.91). However, the sensitivity on the LSO decay constant remains and the rising signal edge has now increased influence via the τ_{LP} parameter.

Figure 4.49 gives an overview of a possible algorithm implementation. Based on the signal model, all constant values and the matrix entries can be precomputed and used as multiplication coefficients. The calculation of the linear θ_i parameters can be realized by three multiply-accumulate branches which are successively supplied with the input data samples. After the evaluation of all matrix entries, the accumulation registers hold the linear parameters which are then combined to the nonlinear ϕ parameter. The required natural logarithm operation can be implemented either by a stored lookup table or by using a CORDIC¹¹ algorithm implementation [MB04, Xil04]. In total, $3 \cdot N_s$ multiplication and addition operations are required to obtain the linear parameter vector. In addition,

¹¹CORDIC: COordinate Rotation DIgital Computer

the final determination of the ϕ parameter requires five multiplications and two division operations.

4.4.4 Summary of the Time Reconstruction

When the time reconstruction performance of the described algorithms is compared, they show almost equal performance values. Although, the least squares based algorithms are able to use the information from multiple data samples, most of the pulse information is carried by the first 2–8 sample values. The digital constant fraction based algorithm allows a hardware efficient use of the two sampling points with the highest information content. The best result for the CFD algorithm is obtained with a smoothing spline interpolation of the input data, with an upsampling factor of only $M = 2$.

The least squares methods suffer mainly from the fact, that the start position of the pulse data has to be known, especially for the analytical calculation of the time offset parameter. The grid search approach can compensate this by systematically evaluating all possible pulse positions and searching for the global minimal error. Though, this implementation requires a significantly higher computation effort, compared to the CFD algorithm. In addition, the least squares methods require a precise model of the expected pulse shape and have a strong dependency on variations of certain model parameters. An evaluation of the grid search algorithm with test pulse data is also presented in [MGK⁺06].

For the CFD algorithm, the pulse model is not necessarily required. The optimal constant fraction setting for the CFD can be determined from sampled data or as well from a pulse model realization. Therefore, the CFD algorithm offers currently the most efficient implementation for the extraction of the sampled pulse time offset parameter. Remaining variations in the reconstructed time due to differing pulse shapes between channels can be compensated in a subsequent time calibration procedure.

4.5 Pulse Shape Optimization

Up to now, the performance evaluation is only based on the pulse shape model from the MADPET-II preamplifier specifications in [Pic01b] and Section 4.1.1. However, the given preamplifier pulse shape, defined by τ_{LP} and τ_{HP} , might not be optimal for the feature extraction with sampling ADCs. To find the optimal pulse shape for different ADC sampling rates, the Cramér-Rao lower bound from Section 4.4.2.2 can be used, since it defines a limit for the best possible amplitude or time resolution, based on the signal parameters. For the constant fraction based time extraction, the CRLB is given by (4.77)

$$\text{var}(\hat{\phi}) \geq \frac{\sigma^2}{\sum_{n=n_1}^{n_1+1} \left(\frac{\partial h[n;\phi]}{\partial \phi} \right)^2},$$

with the signal pulse samples described by (4.69)

$$h[n; \phi] = \alpha \cdot e^{-(n \cdot t_s + \phi) / \tau_{HP}} + \beta \cdot e^{-(n \cdot t_s + \phi) / \tau_{LP}} + \gamma \cdot e^{-(n \cdot t_s + \phi) / \tau_{LSO}}.$$

To improve the achievable time resolution, the CRLB (4.77) must be minimized. For a fixed noise variance σ^2 , this corresponds to the maximization of the CRLB denominator for the free pulse shape parameters τ_{LP} and τ_{HP}

$$(4.92) \quad \max_{\tau_{HP}, \tau_{LP}} \sum_{n=n_1}^{n_1+1} \underbrace{\left(\frac{\partial h[n; \phi]}{\partial \phi} \right)^2}_{f[n]}.$$

The CFD algorithm applies two boundary conditions to the optimization: Since the algorithm operates only on the rising signal edge, both sampling points for the linear interpolation must be placed on the rising signal edge, which gives a condition for the signal rise time $t_r > 2t_s$. Further on, the optimum CFD threshold setting is determined by the position of the peak in the derivation $(\partial h[n_{\max}; \phi] / \partial \phi)^2 = f[n_{\max}]$, as this ensures that for the linear interpolation both data points, n_1 and $n_1 + 1$, give the highest possible contribution to (4.92) for all values of ϕ . Thus the optimization simplifies to the maximization of the peak in the derivation (4.92)

$$(4.93) \quad \max_{\tau_{HP}, \tau_{LP}} \left(\frac{\partial h[n_{\max}; \phi]}{\partial \phi} \right)^2.$$

The numerical evaluation of (4.93) gives as optimal pulse shape parameters

$$(4.94) \quad \tau_{HP} = \tau_{LP} = \tau_{LSO}.$$

The LSO light decay time τ_{LSO} was already discussed in Section 4.1.1 and is usually in the range of 40 ns to 50 ns [MS92, Lud95]. For the simulations in the previous sections, the decay time was fixed at $\tau_{LSO} = 41$ ns. With the condition from (4.94), the pulse shape description in the Laplace domain $H_{LSO}(s) \cdot H_{APD}(s)$ from (4.4) simplifies to

$$(4.95) \quad H(s) = \frac{\tau_{LSO}^2}{(1 + s \cdot \tau_{LSO})^3}.$$

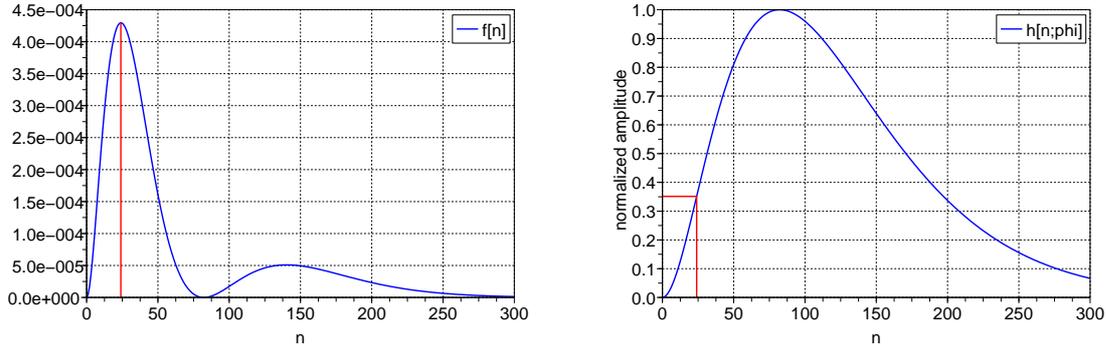


Figure 4.50: Calculation of the sampled Fisher information $f[n]$ (left) to determine the optimal CFD threshold setting (cf. Figure 4.38). Mapping the value of $n_{\max} = 24$ to the normalized pulse shape (right), gives the optimal threshold value of $F_{\text{CFD}} = 35\%$. Since $t_s = 1$ ns, the sample number n corresponds to time in nanoseconds.

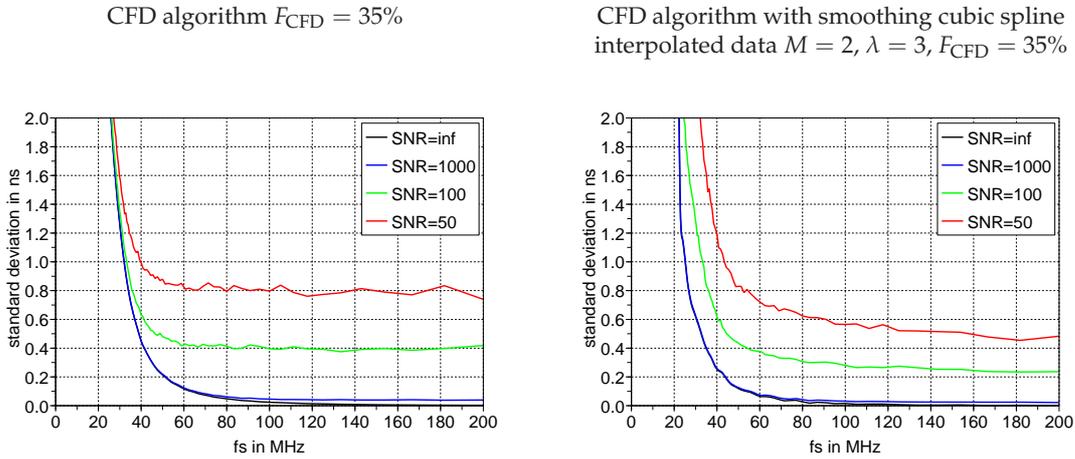


Figure 4.51: Comparison of the CFD algorithm performance for the optimized pulse shape, simulated on sampled data (left) and smoothing spline interpolated data with $M = 2, \lambda = 3$ (right) at different SNR values.

The time domain transformation of $H(s)$ is then given by

$$(4.96) \quad h(t) = \mathcal{L}^{-1}(H(s)) = \frac{t^2 \cdot e^{-t/\tau_{\text{LSO}}}}{2\tau_{\text{LSO}}}.$$

Figure 4.50 shows the new pulse shape, as well as the calculation to determine the optimal CFD threshold, resulting in a threshold value of $F_{\text{CFD}} = 35\%$. Since $t_s = 1$ ns, the sample number n is equal to the time in nanoseconds. As the sampled Fisher information $f[n]$ has its maximum at $n_{\max} = 24$, this corresponds to 24 ns from the start of the pulse to the CFD threshold crossing. To ensure a valid signal sample below the CFD threshold for the linear interpolation, the lowest ADC sampling rate is therefore limited to $f_s = 1/24 \text{ ns} \approx 42 \text{ MHz}$.

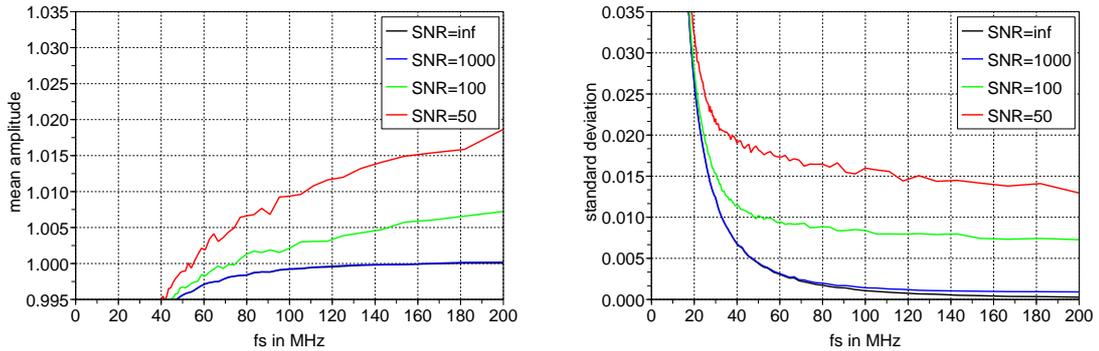


Figure 4.52: Comparison of the relative maximum error (*left*) and the standard deviation (*right*) of the obtained maximum value at different SNR values, using the peak-hold method.

A performance evaluation of the CFD time extraction algorithm operating on the optimized pulse shape is given in Figure 4.51. In the left plot, the algorithm is operated on raw simulated data samples, which shows an improvement of about 40%, compared to the time resolution evaluated with the current pulse shape in Figure 4.40. For sampling rates below 40 MHz, the performance degrades rapidly, since the first sample point for the interpolation is reaching the baseline. The plot on the right hand side shows the time resolution obtained with a smoothing cubic spline preprocessing of the data. For this purpose, the smoothing factor was set to $\lambda = 3$ and the signal was upsampled by a factor of $M = 2$. This improves the time resolution further, resulting in about 30% improvement when compared to the previous smoothing spline data in Figure 4.42.

Since the optimized pulse shape now shows a more distinct peak, compared to the rather flat plateau for the current preamplifier pulse, the simple peak-hold amplitude extraction leads to larger variations in the resulting amplitude. Figure 4.52 shows the corresponding mean amplitude and standard deviation values for different sampling rates. A comparison with the initial peak-hold simulation in Figure 4.27 indicates a degradation of about 20% which results in an increased standard deviation.

Due to the reduced extent of the peak, the effect of accumulating for increasing sampling rates also an increased number of Gaussian distributed noise values is now reduced (cf. Section 4.3.4). This leads to a slower rising of the mean amplitude curve compared to Figure 4.27. However, for the noiseless case, the normalized amplitude value is now significantly underestimated up to sampling rates of about 140 MHz.

By using additional interpolation methods, the amplitude error can again be improved. Figure 4.53 shows the simulation results for the parabola interpolation and smoothing cubic spline functions. The parabola interpolation improves the standard deviation for lower sampling rates, but shows no significant improvement for noisy data. Since the smoothing splines method requires several support knots on the signal pulse to follow the pulse shape, undersampling effects appear for sampling rates below 70 MHz. Above 70 MHz, the achievable precision remains constant, which is comparable to the initial results obtained with the current preamplifier pulse shape given in Section 4.3.4.

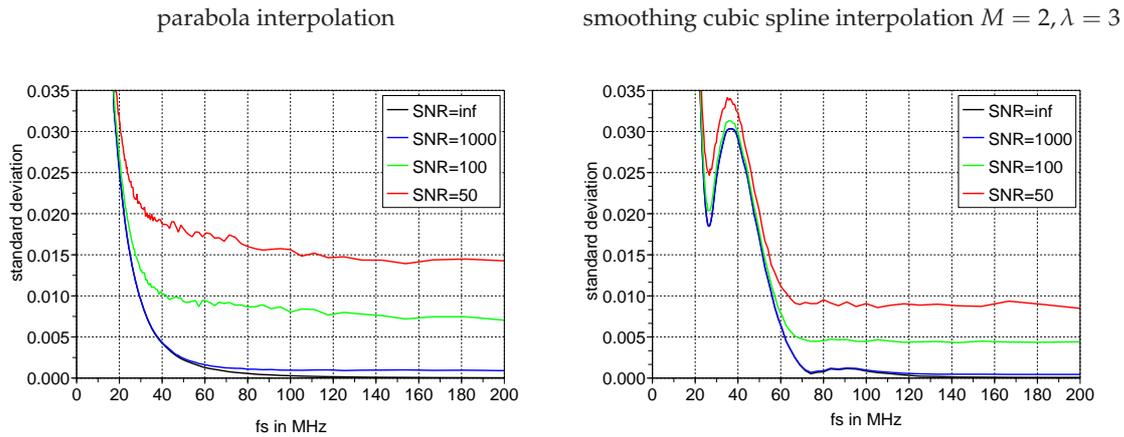


Figure 4.53: Comparison of the standard deviation for the extracted pulse maximum with parabola interpolation (*left*) and smoothing cubic spline interpolation $M = 2, \lambda = 3$ (*right*).

Although the smoothing spline interpolation gives the best result for the maximum extraction, the errors for noiseless signals is for all algorithms still below 1% for sampling rates above 40 MHz. Therefore, the simple peak-hold method is still sufficient since the amplitude/energy resolution for the MADPET-II system is currently mainly limited by the scintillator/photodetector assembly.

Chapter 5

Sampling ADC Based Electronics

To realize the data processing schemes described in the previous chapter, the individual digitization of each front-end channel is necessary. Although the initial idea was inspired by the concept of digital oscilloscopes, the required number of readout channels demand for a compact and cost effective acquisition system, which obviously can not be realized by commodity oscilloscopes. Therefore, dedicated readout electronic modules are required to construct a cost effective and scalable data acquisition system for multichannel applications. Since the sampling ADC based approach can also be applied for detector systems in other fields, like high energy physics or neutron physics, special care had to be taken to achieve as much flexibility for current and future applications. Additionally, the demand for readout channels can range from a few channels for tabletop applications in the laboratory, up to several thousand channels in high energy physics. Ideally, both cases should be manageable by the described data acquisition system.

Depending on the sampling rate, a significant amount of data is generated by the ADCs in the acquisition system. Therefore, the trigger decision and further processing algorithms have to be implemented close to the sampling stage. As the processing tasks can be parallelized in a multichannel design, they can be ideally realized within modern field programmable gate arrays (FPGAs). This allows to extract only the interesting features from the sampled analog data, thus reducing the data rate remaining for later analysis operations.

5.1 The COMPASS Sampling ADC System

The first evaluation of a sampling ADC based data acquisition system (DAQ) for MADPET-II was based on existing electronic modules for the COMPASS¹ experiment at CERN² [M⁺07]. This system consists of several components, which serve the tasks of

¹COMPASS: COmmon Muon Proton Apparatus for Structure and Spectroscopy

²CERN: Conseil Européen pour la Recherche Nucléaire, now *European Organization for Nuclear Research*

acquiring, multiplexing and buffering data. Additionally, the system also allows to distribute a common reference clock to all DAQ modules which provides common timing information on all modules [S⁺04].

5.1.1 The Sampling ADC Module (SADC)

The sampling ADC module (SADC) is realized as a 6U VME card and provides 32 differential analog input channels. All 32 channels are independently digitized by 10 bit ADCs with sampling rates up to 80 MHz. The ADCs are preceded by differential operational amplifiers (OpAmp), which allows the adaptation of the dynamic range and impedance of the input during manufacturing. The signal baseline can be adjusted by two offset voltage sources which have to be configured during the manufacturing process as well.

The digital data from the ADCs are directly read by two Xilinx Virtex-II FPGAs which perform data buffering and signal processing for 16 channels each. The processed data of both FPGAs are then combined in a third link control FPGA and transmitted to the next readout stage via either a serial front panel interface or a parallel backplane interface. Figure 5.1 shows the top side of the SADC card which holds 16 ADC channels. The remaining 16 channels are mounted on the back side of the module.

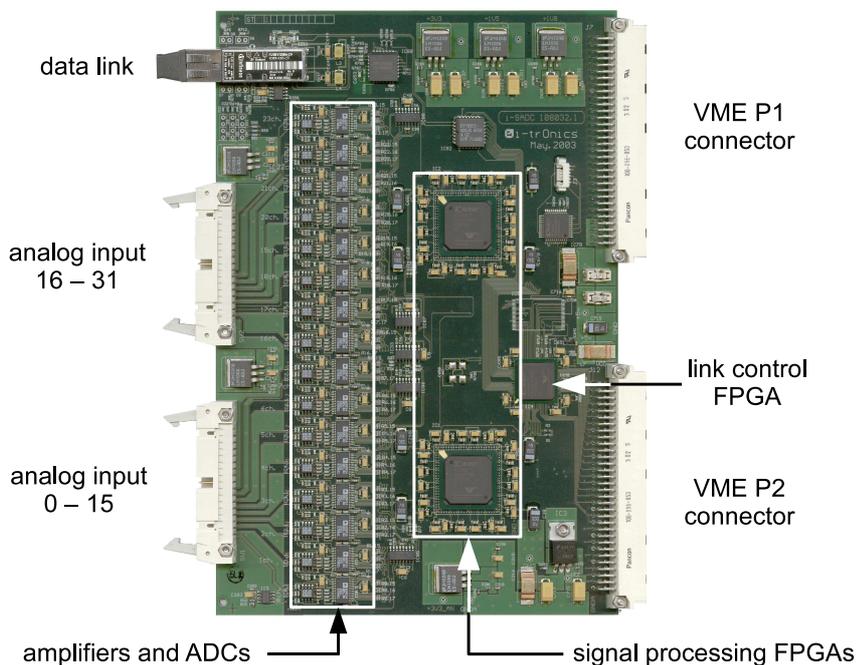


Figure 5.1: Sampling ADC VME module.

The implemented data link can either be based on an optical fiber connection—which ensures galvanic decoupling from the uplink modules—or simple CAT6³ twisted pair cop-

³CAT6: Cable standard (Category 6) commonly used for Gigabit Ethernet and specified up to 250 MHz per twisted cable pair.

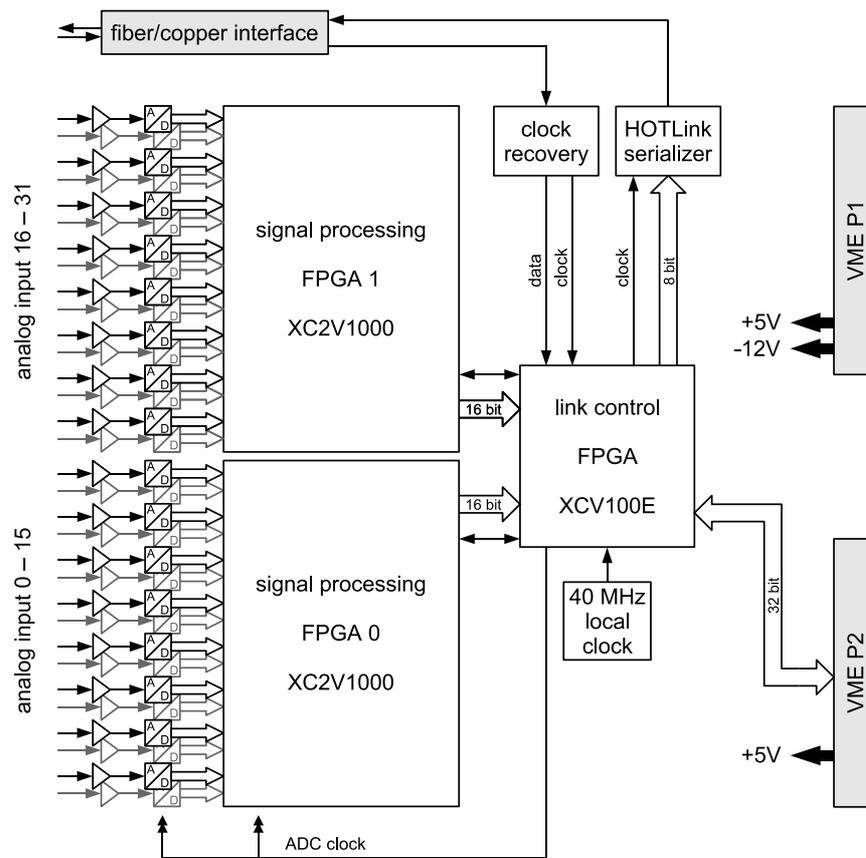


Figure 5.2: Overview of the sampling ADC data flow.

per cable, which allows to reduce the system cost if no galvanic decoupling is required. The raw uplink data rate can be up to 320 Mbit/s via a serializer/deserializer solution from Cypress Semiconductor (HOTLink) [CY05]. The downlink to the ADC module implements a biphase mark encoded signal, which allows to recover a common clock signal, as well as a configuration data and trigger channel. The recovered clock signal is routed to the control FPGA which then generates the clock signal for all ADCs and the processing FPGAs. The firmware of the two processing FPGAs is reprogrammable via the configuration data channel. In addition, this channel provides access to configuration registers within the FPGA firmware to parametrize and control the firmware functionality. For applications in high energy physics which realize an external trigger decision, the trigger channel allows to distribute the resulting trigger with low latency to the connected SADCs and to initiate the readout of the input channels. Figure 5.2 gives an overview of the components and the data flow on the SADC module.

Depending on the implemented FPGA firmware functionality and ADC sampling rate, the SADC power consumption varies between 20–35 W. Power is supplied via the P1/P2 VME connector and requires +5 V and –12 V. Further supply voltages are derived on the SADC module with linear regulators to avoid additional noise in the analog signal chain.

5.1.2 The Multiplexer Module (MUX8)

If several SADC modules have to be read, the individual data streams can be combined by a multiplexer module. Up to eight SADC cards can therefore be connected via the serial HOTLink connection to the MUX8 module. This 6U VME card then merges the incoming data streams and fans out the clock, trigger and configuration signals to the SADCs. The multiplexer is as well equipped with a Xilinx Virtex-II FPGA which can be reprogrammed remotely. To buffer large datasets, the card may also hold standard SDRAM⁴ DIMM⁵ modules up to 512 MB. The configuration registers for loading and configuring the FPGA firmware are accessible via the VME bus which is interfaced by a complex programmable logic device (CPLD). Figure 5.3 shows the top side of a MUX8 module which is equipped with optical fiber interfaces to the front-end SADCs. Not mounted are the socket for the RAM module and an optional uplink port.

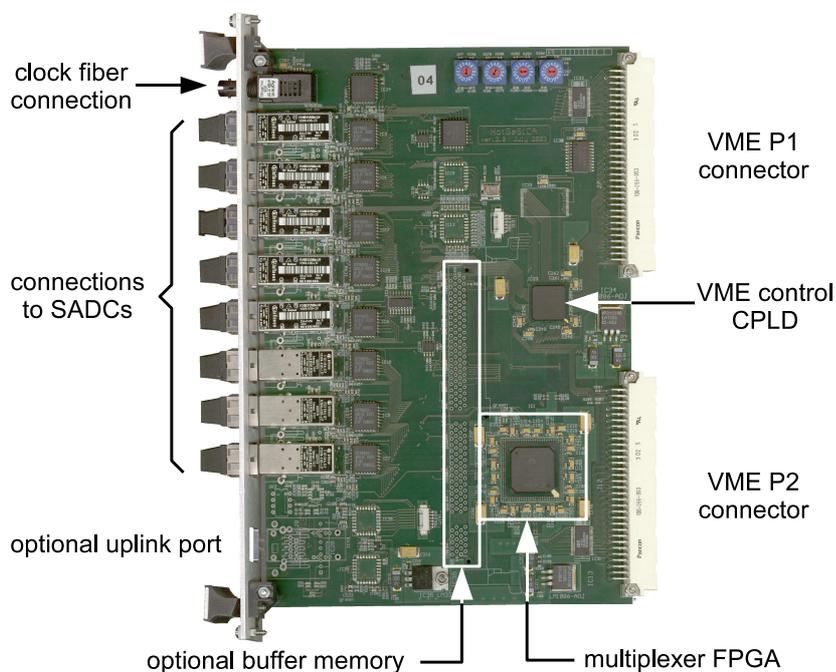


Figure 5.3: Multiplexer VME module.

The multiplexed data stream can be transmitted further in two different ways. One possibility employs the same serial HOTLink connection as already used by the SADC interface to realize an uplink port to subsequent multiplexer stages. This allows to combine data from multiple SADCs in a tree-like fashion as illustrated in Figure 5.4. However, the maximum data rate is thereby limited to the 320 Mbit/s provided by the HOTLink interface. If higher data rates have to be handled, each multiplexer can connect to an interface card via the P2 VME connector which allows e.g. data rates up to 1.28 Gbit/s with the S-Link interface standard [vdBMBR97] developed at CERN.

⁴SDRAM: Synchronous Dynamic Random Access Memory

⁵DIMM: Dual In-line Memory Module

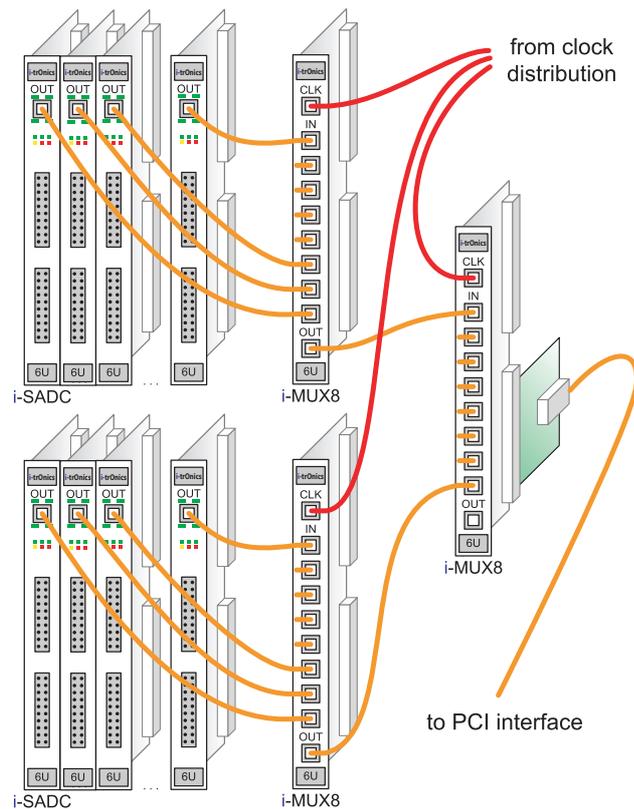


Figure 5.4: Example configuration of an SADC readout network.

The configuration access to the multiplexer and all connected SADCs is implemented via the VME backplane interface. Thus a crate conforming to the VME standard and a corresponding VME-CPU module is required. This allows to upload firmware files into the FPGAs and to access the configuration registers in the loaded FPGA firmware. Similarly, the upload functionality is also provided for the SADC modules via the HOTLink connections.

5.1.3 The Clock Distribution

To ensure a consistent timing information on all front-end SADC modules, all clock signals are derived from a central clock source. Therefore, the master clock signal is distributed by a passive optical fiber network to all multiplexer modules. The clock signal for the SADCs is then forwarded via the bidirectional data connection which also transmits the configuration data to the front-end modules. For applications which require the distribution of a central trigger information, this task is also realized by the clock distribution network. Thus, this building block is named *Trigger Control System (TCS)* [Gru01, Gru06] within the COMPASS experiment at CERN.

The achievable precision of the local clock reference on the front-end modules is determined by the jitter of the distributed clock signal. For the first clock recovery stage on

the multiplexer modules the clock jitter is in the order of 50 ps [Gru01]. From there, the clock signal is re-encoded with the data from the configuration channel and distributed to the front-end modules. On the SADC card the clock signal has to be recovered again and is then used in the FPGA firmware to derive the actual ADC clock signal. These multiple encoding and recovery stages cause some degradation in the jitter properties of the clock signal. For two SADC modules, connected to different multiplexers the clock jitter is therefore in the order of 400 ps.

Since the low jitter requirements also demand a precise master clock signal, the TCS system uses a master oscillator rated for the SONET⁶ OC-3 telecommunication standard. The SONET standard therefore defines the initial clock frequency of 155.520 MHz from which all further clock signals in the system are derived. Therefore, the actual SADC sampling frequency is $155.520 \text{ MHz}/2 = 77.76 \text{ MHz}$ if the clock is supplied via the TCS system. The encoding scheme of the data transmission via the TCS signal is based on biphasic mark encoding which allows to recover both clock and data at the destination. Since the system is based on standardized bitrates, commercial clock recovery integrated circuits can be used.

5.1.4 The PCI Interface/Buffer Module

The interface of the DAQ system to a standard personal computer based infrastructure is realized by PCI⁷ buffer cards, which provide a mounting option for S-Link cards as a piggyback module. To avoid data loss due to bursts in the incoming data stream or delays on the PC driver level, the buffer module can access up to 512 MB of registered SDRAM in a standard DIMM package. The firmware of the Xilinx Virtex FPGA then uses the random access memory (RAM) to implement one big first-in first-out (FIFO) memory to buffer the data stream. The output data of the FIFO buffer can then be transferred via direct memory access (DMA) to the main memory of the PC. From there on, the data handling is implemented by PC based software, which may store the data to disk or do further online processing. Measurements in [S⁺04] showed, that for proper DMA settings the readout rate via the PCI interface can reach up to 100 MB/s once the transfer block size is larger than 40 kB. Figure 5.5 shows the PCI buffer card with mounted SDRAM memory. On the bottom, an S-Link card is shown which plugs top-down to the marked space on the buffer card.

As an alternative to the S-Link interface, the PCI buffer may also be directly connected to an SADC by a dedicated piggyback module. Thus, for small scale applications, the SADC readout may be realized without multiplexer modules and the clock distribution system, which allows a simplified setup. This is especially useful in experiments or lab setups which require only a small number of readout channels, but still want to benefit from the FPGA based data processing in the SADC. Examples of such applications are

⁶SONET: Synchronous Optical NETWORKing

⁷PCI: Peripheral Component Interconnect

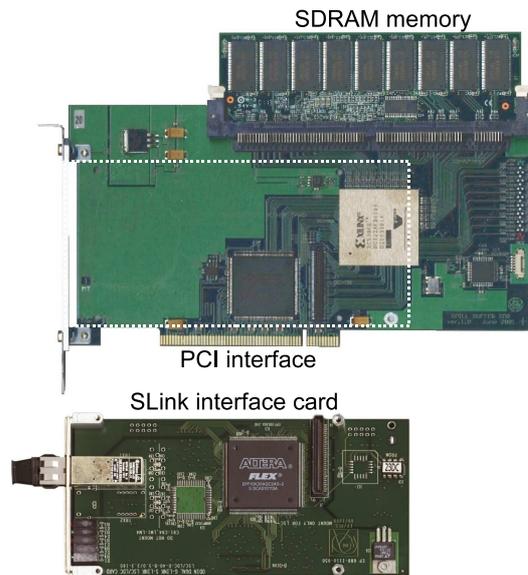


Figure 5.5: PCI bus interface and buffer module with unmounted S-Link card.

the *a*SPECT spectrometer [Pet07] and AbEx [Pic08], which used the system to acquire data from proton and neutron detectors.

Since the parallel PCI bus is successively replaced by the serial PCI Express (PCIe) standard, a redesign of the interface card was required to retain compatibility with up-to-date computing equipment. To ease a later transition to the emerging Advanced Telecom Computing Architecture standard (ATCA, see Section 5.4), a PCI Express evaluation card for the Xilinx Virtex 5 FPGA family [XV509, IV508] was used as a first development platform. The integration of high-speed gigabit-capable transceivers into the FPGA fabric allows then to integrate the PCI Express interface and multiple S-Link interfaces within the same FPGA. Similarly, the buffer memory can be extended with recent memory modules up to several gigabytes.

5.1.5 SADC USB Interface

To simplify the setup of small-scale sampling ADC systems for laboratory use, the full readout chain with MUX8, TCS and PCI interface can as well be substituted by a standardized computer interface. Therefore, one SADC module is connected via the P2 VME connector to an extension card which provides a USB⁸ 2.0 interface. The host PC then allows to configure the SADC firmware and to read the output data stream similar to the full DAQ system. The required SADC clock is provided by a local 40 MHz oscillator on the SADC, which is doubled in the link control FPGA to reach the 80 MHz sampling clock. Hence, this results in a slightly higher sampling frequency compared to the

⁸USB: Universal Serial Bus

77.76 MHz clock derived from the TCS clock source, which has to be accounted during the data analysis.

5.2 Introduction to Field Programmable Gate Arrays (FPGAs)

The main functionality of the modules described in the previous Section 5.1 is implemented in field programmable gate arrays (FPGAs). FPGAs are reprogrammable integrated circuits, which can be configured to realize complex logic functions directly on the hardware level. This is in contrast to a processor based design, where a software program implements the functionality. Whereas the program is sequentially executed by the processor, an FPGA may implement similar functionality by connecting parallel functional blocks. Hence, FPGAs are advantageous in environments where parallel data processing is required with high throughput. In addition, the functionality of a “normal” processor design can as well be realized within an FPGA. This requires that the processor itself is described on the hardware level and has to be realized by the FPGA configuration.

5.2.1 Internal Structure

The structure of modern FPGAs consists of a regular array of so called *configurable logic blocks* (CLB), which are surrounded by an interconnection network. The logic blocks allow to realize logic functions of the input signals (like AND, OR, NOT or other boolean functions) and provide optional storage registers for the output signals. The CLB input and output signals can then be connected by the interconnection network to form more complex logic functions than possible within one logic block. Figure 5.6 gives a simplified overview of an FPGA fabric.

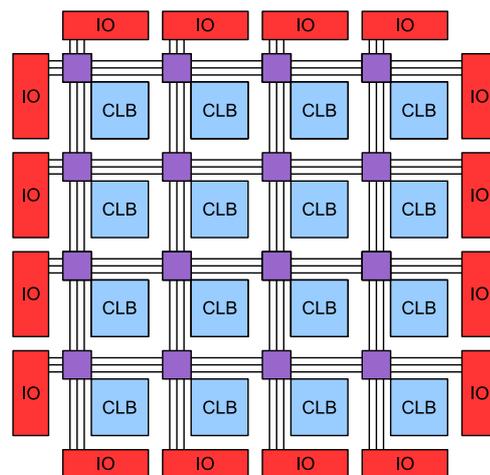


Figure 5.6: Simplified view of an FPGA fabric with configurable logic blocks (*blue*), input/output blocks (*red*) and routing switch boxes (*purple*).

Usually, the external input and output signals of the chip are interfaced to the fabric by dedicated IO-blocks, which allow to interface to various IO-standards and voltage levels. Additionally, the IO-blocks may include buffer registers and more complex interface logic to adapt to modern high speed memory (e.g. DDR-RAM⁹) or chip interconnection standards (e.g. HyperTransportTM). For high speed serial interface standards, some FPGA devices include also serializer/deserializer blocks which can handle up to several Gbit/s. This allows for example to interface directly to PCI Express or Gigabit Ethernet.

The configurable logic blocks include for example lookup-tables (LUT) to realize boolean logic equations and storage registers to realize synchronous logic, like counters or finite state machines (FSM). These components are interconnected by several multiplexers to provide a reasonable amount of flexibility in the configuration. By loading the lookup-tables with appropriate bit-patterns, any logic function of the input signals can be generated. To simplify the design of high level functions—like for example adders—the CLBs may have additional interconnections between neighboring CLBs to speed up e.g. the carry signal path in a ripple carry adder architecture.

For signal processing intense applications, the FPGA can provide in addition hardware for multipliers, first-in first-out buffers (FIFO) or random access memory blocks (RAM), which otherwise would require significant resources when implemented via the CLBs. This may even include complete processors which are realized as so called *hard-cores* directly on the FPGA silicon. In contrast, the same functionality can also be realized by configuring the FPGA logic elements and switch network, by utilizing a *soft-core* based design. However, the resource usage for the soft-core design usually is higher with a smaller maximum possible clock frequency. The exact layout of the FPGA and the internal options highly depend on the vendor and the chosen FPGA family. In principle, all vendors and devices offer comparable features, however with different flavors and realizations [XV207, XV408, AS308, AS408, LEC09, LSC08].

5.2.2 Design Process

To implement the desired functionality into the FPGA device, a software framework is usually provided by the vendor. This framework then allows to enter the design on various abstraction levels, ranging from the individual configuration of CLBs up to schematic entry or high level hardware description languages (HDL). As modern FPGA devices easily include more than 10000 CLBs, a hardware description language—like VHDL¹⁰ or Verilog—offers a convenient way to structure the firmware design. Via simulation programs, the HDL description of an FPGA firmware design can also be simulated prior to the actual implementation into the FPGA fabric. This is an essential advantage for complex firmware designs, since all internal FPGA signals can be monitored without the existence of real physical hardware probes.

⁹DDR: Double Data Rate

¹⁰VHDL: VHSIC Hardware Description Language, VHSIC: Very High Speed Integrated Circuit

Once the FPGA functionality is specified and simulated successfully, a synthesis step compiles the HDL design into a description, based on hardware primitives. In a second step, the primitives have to be mapped to the logic resources inside the target FPGA. The final steps then consist of the resource placement inside the FPGA fabric and the routing of the interconnection signals. Since the arrangement of logic and routing paths inside the FPGA results in specific delay values for logic and routing, this process has to be guided by specifying additional timing constraints for the design. This may be for example the target clock frequency or minimal delay requirements between logic blocks.

Although the HDL based design approach is in principle portable between different FPGA types, the usage of vendor/FPGA specific soft-cores usually binds the design to a specific device and vendor. For the presented ADC system, the designs are based on the Virtex-II [XV207] and Virtex-4 [XV408] series from Xilinx Inc.

5.3 Firmware Development

Since all components of the COMPASS SADC system are based on field programmable gate arrays, a dedicated firmware is required for each FPGA to implement the required functionality of the module. For an implementation in the field of positron emission tomography, some parts of the firmware design can be reused from the initial realization within COMPASS. However, the data acquisition task for PET applications differs from the structure within the COMPASS experiment, where data are acquired upon a central trigger decision. In contrast, as the emerging photons within PET may interact in any detector, independently of all other detector channels, the trigger decision has to be localized within each front-end channel instead.

Therefore, the data from different front-end channels appears completely uncorrelated, which requires a continuous handling of the data flow in the multiplexer level, in contrast to the trigger/event based approach within COMPASS. In fact, some correlations do exist within the data stream due to the coincident photon pair which is emitted and eventually detected simultaneously. Though, the extraction of this feature requires information from opposing detector elements to determine the corresponding line of response. Therefore, this feature is more easily implemented by post-acquisition software, especially if the detector geometry and the channel number may still change.

The main part of the firmware development was therefore focused on the SADC module, including the trigger decision and pulse data processing. Some modifications were also required for the MUX8, to handle the continuous input data streams. The clock distribution and the PCI-interface could be used without modifications.

5.3.1 SADC Firmware

Different firmware architectures were realized for the SADC module, targeting the restrictions of previous versions. The first implementations for the SADC pulse data pro-

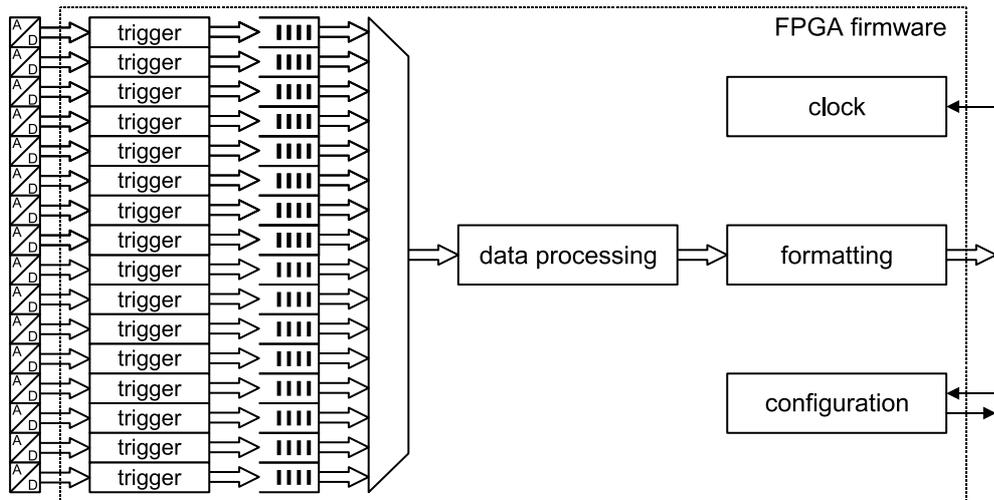


Figure 5.7: Overview of the firmware data flow in the processing FPGA on the SADC.

cessing are described in [Man04, MGK⁺06]. Since the limited FPGA resources do not allow a fully parallel implementation of the data processing for all 16 channels which are attached to one FPGA, some multiplexing scheme of the data has to be implemented. Figure 5.7 shows the basic structure of the processing FPGAs on the SADC module.

From the left side of Figure 5.7 the ADCs are sending the sampled input data directly into the FPGA firmware. Since every channel has to decide independently whether a valid signal is in the data stream, each of the 16 input channels implements its own trigger functional block. Upon a valid trigger decision, an event packet with timestamp and a certain number of data words is created and stored into an event buffer. The 16 buffer memories are then continuously polled by a multiplexer block, which merges the event packets into one single event data stream. Each event is then individually processed by the data processing unit(s) and finally a formatting stage creates an output packet, which is then transmitted to the link control FPGA.

Besides the main data processing, the firmware additionally implements a configuration register block, which allows to parametrize the firmware during operation. The FPGA is supplied with the common ADC clock frequency, which is directly distributed to the different firmware functional blocks to allow a synchronous operation with the incoming ADC data.

5.3.1.1 Trigger Block

The trigger block has to discriminate valid detector pulses from the background noise. As described in Section 4.2, the simplest implementation may realize a threshold comparison of each input data word and issue a trigger once the amplitude exceeds the threshold. However, the robustness of the trigger decision in presence of noise can be enhanced if

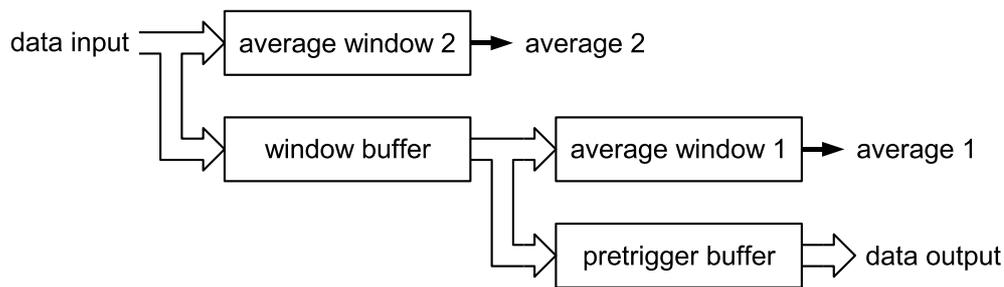


Figure 5.8: Overview of the data flow in the trigger block.

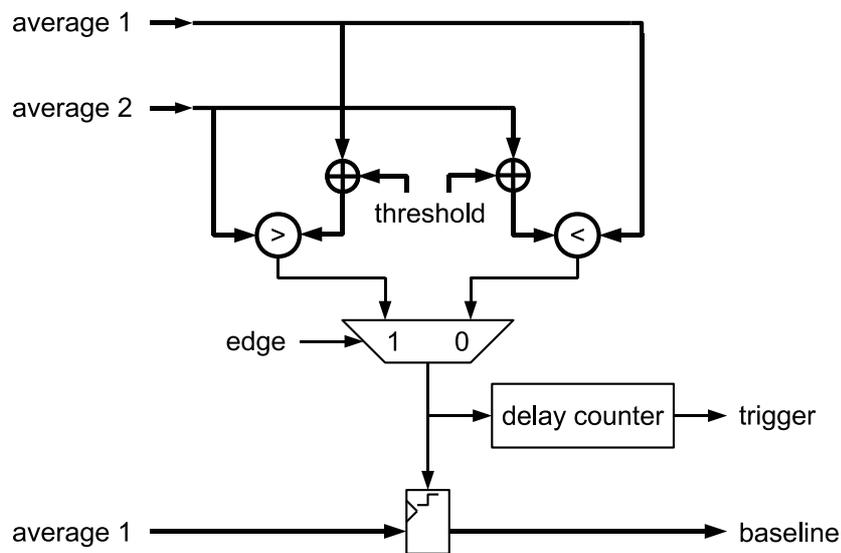


Figure 5.9: Overview of the trigger logic.

the threshold decision is applied to averaged input data values. To allow a flexible adaptation to different environments and pulse shapes, the average and trigger parameters have to be configurable. Nevertheless, the trigger block is repeated 16 times in the FPGA fabric. Therefore, a resource efficient realization is mandatory. Figure 5.8 and Figure 5.9 give an overview of the data flow and the trigger decision logic within each trigger block.

The trigger decision is based on two average values which are calculated from the input data stream based on Equation (4.19) described in Section 4.2.1. The number of averaged values can be configured in units of 2^x , which simplifies the division operation in the averaging to a binary bit shift. The first average value thereby gives an estimation for the current baseline value, whereas the second average includes the evolving pulse data. Both average windows are separated by a configurable number of samples, realized by the *window buffer*. To be able to acquire also data prior to the trigger event, a configurable *pretrigger buffer* delays the data stream before it is transferred to the event buffer block.

The trigger logic can be configured by the average difference threshold, which must be exceeded to initiate a trigger signal. The *edge* signal allows to select the trigger between

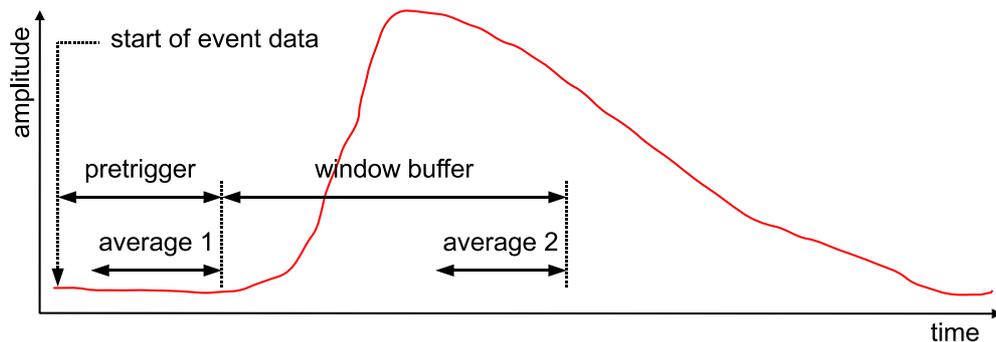


Figure 5.10: Overview of the average window and buffer relations, defining the starting point of the event data.

rising ($edge = 1$) and falling ($edge = 0$) signal edges. Once the threshold value is exceeded, the first average value is stored in addition as baseline information. The “real” trigger signal is only generated if the threshold is exceeded for a certain number of data samples or clock signals respectively, which is realized by an additional *delay counter*. If the input signal vanishes before the trigger delay is reached, the delay counter is reset without issuing any trigger. This allows to suppress short spikes on the input data, which show shorter rise and fall times compared to the valid pulse data.

Figure 5.10 illustrates again the relations between the average windows and the buffer memories in the data stream. The end of the pretrigger buffer defines the starting point of the acquired event data. By increasing the trigger delay, the pulse data are shifted to the left of Figure 5.10, with the buffer and window position fixed. Therefore, a higher trigger delay setting may be compensated with a correspondingly increased pretrigger buffer size.

5.3.1.2 Event Buffer Block

Upon a valid trigger signal, the event buffer block has to store the emerging data samples from the trigger block for further processing. Therefore, an event packet is generated, which contains all required information for this specific signal pulse. This includes a timestamp for the trigger decision, the baseline value and the acquired ADC samples. Figure 5.11 shows an overview of the data flow in the event buffer.

The 30 bit time counter is continuously incremented by the sampling clock of the ADC. Once a trigger signal is detected, a snapshot of the time counter is stored. Therefore, the time value is split into three 10 bit segments, which fit into the normal width of the data path, which is either 10 bit or 12 bit, depending on the ADC type. After the three time words, the baseline value is stored, followed by a configurable number of data samples.

Since the sampling frequency f_s is known, the time counter value can be converted into a real time value by $t = n_{\text{counter}} \cdot (1/f_s)$. Due to the limited width of the time word, the

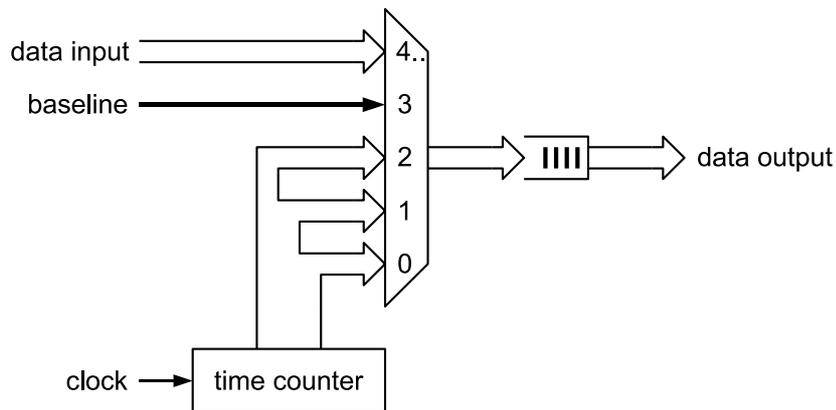


Figure 5.11: Overview of the data flow in the event buffer block.

counter value overflows after $t_{\text{overflow}} = 2^{30} \cdot (1/f_s)$ and restarts counting from zero. This effect has to be corrected later on during the analysis of the acquired data.

Since the first-in first-out (FIFO) buffer memory can only hold a limited number of data words (e.g. 1 kWord), the buffer may reach an overflow condition at some point in time. To avoid the uncontrolled loss of data, this condition is handled separately. If upon a trigger attempt, the remaining free space in the buffer is smaller than the maximum possible event size, special overflow events are stored, which include only the three time words. This allows to retain at least the trigger time information of the detected pulse and which channel went into the overflow mode. Once the buffer FIFO is completely full, all trigger attempts are ignored.

5.3.1.3 Channel Multiplexer Block

The channel multiplexer combines the event data from all 16 input channels and provides a single data stream for the subsequent processing blocks. Additionally, it has to add an additional word to the event packet, which designates the corresponding input channel number. Depending on the available hardware resources, two implementations are possible, including either a time sorting of the merged event data or not. The simplest realization just probes all channels for data in a round robin scheme and successively reads a single event from each requesting event buffer. The FPGA resource usage for this method is very small, since only a counter and a state machine is required to implement the round robin.

However, the simple multiplexing scheme has the effect that, depending on the current position of the channel readout pointer, events may be mixed in time at the multiplexer output. If further data processing relies on time ordered event data, a more complex multiplexing model is necessary. Therefore, a dedicated schedule memory can track the time history of the trigger decisions and thus determine the channel multiplexing sequence. An overview of the dataflow for this case is given in Figure 5.12.

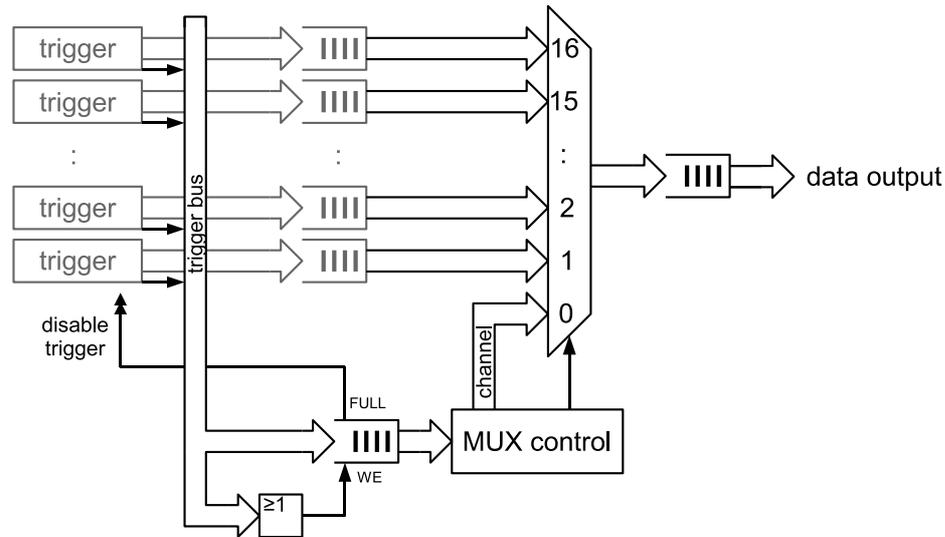


Figure 5.12: Overview of the data flow in the channel multiplexer with time-order scheduling.

To avoid the loss of scheduling data, the minimum required schedule buffer size $N_{\text{schedule}}^{(\min)}$ is determined by the maximum number of event packets $N_{\text{event buffer}}^{(\max)}$ in the individual channel event buffers and the number of total channels K by

$$(5.1) \quad N_{\text{schedule}}^{(\min)} = K \cdot N_{\text{event buffer}}^{(\max)}$$

Additionally, the trigger logic can be disabled once the schedule buffer becomes full. Thus a compromise between the buffer requirements for normal event data and overflow events (which include only the three time words) can be realized, ensuring the consistency of the schedule buffer.

5.3.1.4 Data Processing Block(s)

The multiplexed event data from the input channels can now be analyzed by several processing stages. To allow the flexible arrangement of different processing units in varying application requirements, all data processing blocks follow a common pipelined data flow architecture. Therefore, one full event packet is read into an intermediate buffer FIFO, while the processing logic analyzes the included data samples. The result of the data processing is then added to the buffered event packet, prepending the ADC data samples. This enables the following processing stage to use the computed results already for further operations on the data samples. The processing results are added to the event while the buffered packet is moved to the output buffer FIFO. Since all processing blocks, as well as the channel multiplexer, implement the same output buffer, the blocks may be

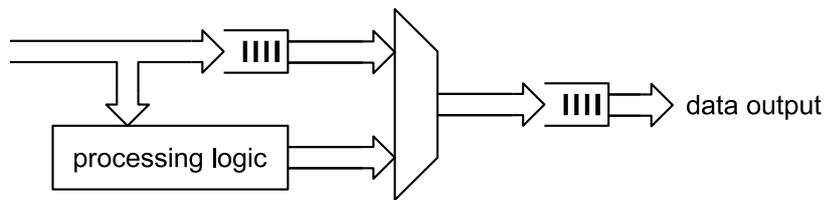


Figure 5.13: Overview of the data flow in the processing blocks.

chained in any order, as required by the processing task. Due to the double buffering scheme, the processing of a new event may already begin, while the current data are moved to the output FIFO. Figure 5.13 illustrates the data flow in the processing block.

For the MADPET-II readout application, the processing logic has to implement the amplitude and time extraction tasks, as already discussed in Chapter 4. Since the time extraction requires also information about the amplitude of the sampled signal, the two tasks can be ideally implemented as two chained processor blocks. The first processor block extracts the amplitude and adds the determined value to the data stream. The second block then analyzes the pulse samples to determine the precise time of the pulse, for example by the CFD algorithm described in Section 4.4.2. The various methods to interpolate additional sample values fit to this scheme as well and can be placed at the beginning of the processing pipeline. Figure 5.14 shows the data flow through the chained processing blocks.

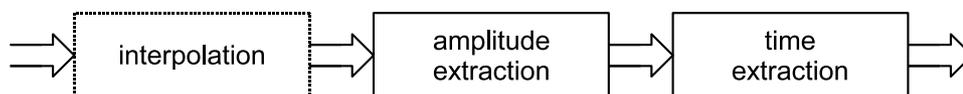


Figure 5.14: Processing chain for the MADPET-II application.

5.3.1.5 Formatter Block

After all processing stages are finally passed, the formatter block combines all relevant information into an output packet. The final packet is stored in an output FIFO buffer which is then accessed by the link control FPGA. If no further off-line processing is foreseen, the data samples may be discarded at this stage to reduce the amount of output data. For compatibility with the existing COMPASS environment, the output packets are formatted in 32 bit words, with two reserved words marking the packet boundaries.

The generic output format is listed in Table 5.1 for the SADC configuration with 10 bit ADCs. For SADCs with 12 bit ADCs, the format of the data word changes as shown in Table 5.2. The *packet size* field indicates the number of 32 bit data words, not including the header and trailer.

31						0
header: 0x00000000						
0000 0000 0000 0000 0000					packet size (12 bit)	
10	status (6 bit)	000	channel (5 bit)	0000	packet size (12 bit)	
01	timestamp (30 bit)					
01	data 3 (10 bit)	data 2 (10 bit)			data 1 (10 bit)	
⋮						
01	data N (10 bit)	data $N - 1$ (10 bit)			data $N - 2$ (10 bit)	
trailer: 0xFFFFFFFF						

Table 5.1: Format of the SADC output packet for 10 bit ADCs. The first packet size word (*dashed*) is added by the link control FPGA.

31				0
01	00 0000	data 2 (12 bit)	data 1 (12 bit)	
⋮				
01	00 0000	data N (12 bit)	data $N - 1$ (12 bit)	

Table 5.2: Data word format for 12 bit ADCs.

Up to now, the FPGA firmware includes only the amplitude and time extraction functionality without any interpolation, since the available logic resources in the Virtex-II based COMPASS SADC module are limited. As each processing stage is independently pipelined, the maximum data throughput is determined by the slowest processing task. Table 5.3 lists the required number of clock cycles for each processing pipeline stage and the corresponding time delay for a clock rate of 80 MHz. The number of acquired data samples for each event is therefore $N = 32$.

Pipeline stage	Required clock cycles	Time delay for $f = 80 \text{ MHz}, N = 32$
Channel multiplexer	$N + 7$	487.5 ns
Amplitude extraction	$N + 16$	600 ns
Time extraction	$(N + 22)$ to $(2N + 22)$	675–1075 ns
Output formatter	$N + 14$	575 ns

Table 5.3: Required clock cycles for each stage of the SADC firmware processing pipeline and resulting time delays for 80 MHz clock frequency. N is the number of samples that are acquired for every trigger event.

The highest processing time demand—which is also varying with the signal properties of the processed pulse—is determined by the time reconstruction block. If the worst case time delay of 1075 ns is assumed, the pipeline can handle about 58000 events per second for each input channel, which is well above the expected average trigger rate of 10000 s^{-1} . However, as the input buffer for each channel can only hold a limited number of events, the trigger rate has to be roughly equal on all input channels to avoid buffer overflows. Since the detector elements are usually mounted at an almost equal distance to the radioactive source, this condition can be easily fulfilled. In addition, the trigger threshold setting allows to adjust the signal rate for each channel by suppressing Compton scattered events with low signal amplitudes.

5.3.1.6 Clock and Configuration

The two remaining blocks in Figure 5.7 are responsible for providing the global clock to the firmware modules and to access the configuration registers which allow to parametrize the firmware functionality. Since the processing FPGAs are supplied with the same clock signal as the ADCs, a fully synchronous design is achieved by distributing the clock to all firmware modules. Additionally, also other clock frequencies may be generated by the FPGA clock resources. This allows for example lower clock rates for processing stages which do not meet the timing requirements or increased processing clocks for higher data throughput. By the use of asynchronous FIFO buffers at the output of the firmware blocks (cf. Figure 5.12 and Figure 5.13), the transition between different clock domains is possible.

The configuration block implements a simple I²C bus¹¹ interface to the link control FPGA, providing a readable and writable register bank. The registers then directly interface to the corresponding ports on the firmware modules, to set e.g. thresholds or to read status information.

5.3.1.7 Link Control FPGA

After the data formatting, the link control FPGA multiplexes the two data streams from both processing FPGAs. Currently, only a simple round-robin multiplexing scheme is implemented. Therefore, this may cause again a disordering in time of the event packets from both processing FPGAs. A guaranteed time ordered multiplexing can be realized by a similar method as described in the channel multiplexer Section 5.3.1.3. This requires in addition to the data bus connection an additional “trigger” signal between the FPGAs, to indicate that a new event packet is written to the output FIFO of the processing FPGA.

In addition, the size of the data packet is reevaluated and prepended to the data packet as an additional packet size word. The multiplexed data stream is again buffered in a FIFO memory and is then transmitted to the next readout stage. This may be realized

¹¹I²C: Inter-Integrated Circuit

either by the serial HOTLink data link, which connects to the MUX8 module or via the P2 VME backplane connector, interfacing e.g. to the SADC USB extension card described in Section 5.1.5.

5.3.2 MUX8 Firmware

If arranged in a tree-like topology as shown in Figure 5.4, the MUX8 module has to multiplex the data streams from up to eight upstream SADC or MUX8 modules. Again, a simple round-robin scheduling creates the problem of a possible time disordering of the event packets. Additionally, the HOTLink connection to the upstream modules can not easily provide the trigger signal, as required by the scheduled multiplexer from Figure 5.12. Instead, the multiplexer stage has for example to evaluate the received packet data and compare the included timestamp values. This allows to select always the channel with smallest time index. However, due to the 30 bit width of the time information, the evaluation step is expensive in terms of logic requirements. Additionally, the overflow of the timestamp value has to be handled properly. The first firmware realizations were therefore only implementing the round-robin scheduling, since all events could be sorted after the acquisition by the analysis software.

The MUX8 module also receives the clock signal from the TCS system (cf. Section 5.1.3) and distributes it to all upstream modules. In addition, control sequences can be send to the register banks in the SADC modules via the MUX8 VME interface. For this functionality, most parts of the firmware could be reused from the COMPASS environment.

5.4 ATCA Based Electronics

The described COMPASS readout electronics is quite flexible, due to the online FPGA firmware reprogrammability and the existing software framework for configuration and data readout. However, some limitations are present for the readout of PET applications, similar to the MADPET-II system. At first, the available logic resources in the various FPGA stages is just sufficient for the described processing tasks, with only limited safety margin for future extensions. Moreover, the embedded FPGA family is slowly reaching the end of life status, with increasing unit prices and decreasing support in recent versions of the development tools. Therefore, either a replacement of the current FPGA family with up-to-date technology or a complete redesign of the DAQ modules is required.

In addition, the current design uses the VME backplane connection only for power supply and slow control communication. The high bandwidth data transfer requires the routing of additional cables via the module front panel as the VME bus cannot provide enough communication bandwidth. Therefore, a redesign could also address improvements in backplane and crate technology, to reduce module wiring and increase reliability. Furthermore, the power consumption of the current SADC module can exceed the

VME specification for supply currents and—even more critical—system cooling capabilities, depending on the implemented firmware complexity. Hence, an upgrade of the module form factor is also desirable.

The Advanced Telecom Computing Architecture (ATCA or AdvancedTCA) [PIC06a] is a new system standard, which originates in the telecom industry. Thus, it provides several advantages over the existing VME specification, namely in terms of system reliability and fault tolerance. In addition, the standard is accepted by a large industry which results in a broad range of compatible equipment and tools at competitive prices.

The ATCA standard defines the mechanics of a new crate type (called *shelf*), as well as electrical and thermal properties and management procedures. The form factor of the ATCA *Front Boards* provides a height of 8U with a maximum power dissipation of typical 200 W per board. In addition to the front panel, an optional *Rear Transition Module* (RTM) allows to feed signal inputs from the back side of the shelf. Front boards may either implement directly the functionality or may provide additional sockets for up to eight *Advanced Mezzanine Card* (AMC) [PIC06b] modules. The communication links between all boards in a shelf are provided via the shelf backplane, which realizes differential point-to-point connections for high speed serial links. Depending on the communication pattern and data flow, the backplane may be implemented in either a star or full mesh topology, reaching a total communication bandwidth from several Gbit/s up to Tbit/s.

The AMC modules are also covered by the MicroTCA [PIC06c] standard which allows to create small scale systems, compatible to the full ATCA specification. Therefore, DAQ systems for laboratory use can be build using MicroTCA components which can scale later on to the full required number of channels by migrating to ATCA. In summary, the AdvancedTCA, MicroTCA and AMC standards are currently gaining increasing acceptance in the physics community, especially for experiments with high data rate applications and high reliability demands. Current emerging experiments focusing on a complete ATCA based readout or control infrastructure are for example the PANDA¹² experiment [PAN05] at GSI¹³, the XFEL¹⁴ [XFE07] at DESY¹⁵ and the ILC¹⁶ [ILC07]. Thus, updates of the existing DAQ components, described in the previous Section 5.1 will successively migrate toward the ATCA standard.

5.4.1 The Mezzanine Sampling ADC

The first redesign of the COMPASS DAQ system was focused on the SADC module. Ideally, the 6U VME card should be replaced by an AMC or ATCA card with similar or increased performance and channel number. However, the concept of crate/shelf based ADCs causes some distance between the detector and the ADC, which has to be bridged

¹²PANDA: antiProton ANnihilations at DArmstadt

¹³GSI: Gesellschaft für Schwerionenforschung, now GSI Helmholtzzentrum für Schwerionenforschung GmbH

¹⁴XFEL: X-ray Free-Electron Laser

¹⁵DESY: Deutsches Elektronen-SYnchrotron

¹⁶ILC: International Linear Collider

by analog signal cables. To reduce the cabling effort, sometimes an integration of detector and ADC at short distance may be favorable. This lead to the design of a small form factor mezzanine card, the Mezzanine Sampling ADC (MSADC) [MKP07a]. Figure 5.15 shows the MSADC module with its dimensions and key components.

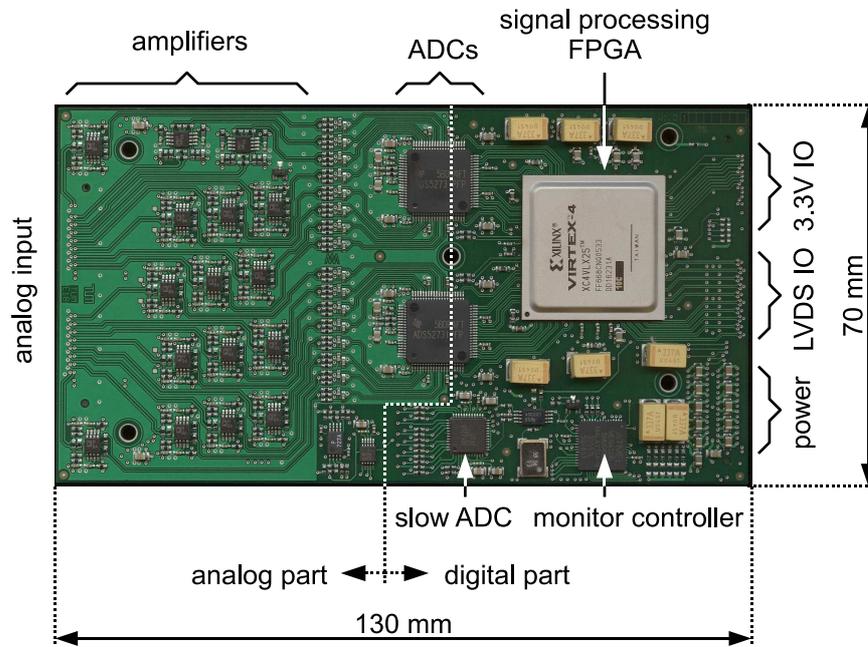


Figure 5.15: Mezzanine sampling ADC module.

The MSADC card separates the basic DAQ core—namely the analog front-end with ADCs and the signal processing FPGA—from all supporting circuitry, like power supplies and the digital readout interface. This gives the advantage that the system can be easily adapted to different readout requirements in a laboratory or the final detector setup. An increment in sampling frequency or ADC resolution does not require to change the readout interface and, vice versa, the same mezzanine card can work with different readout configurations. Thus, the number of spare parts can be reduced and existing systems may be easily upgraded once more ADC features are required. To be compatible with modules conforming to the AMC standard, the size of the MSADC card is 130 mm × 70 mm. This allows to mount the MSADC onto an AMC carrier card, which then may be deployed within a MicroTCA or ATCA system. As well, the MSADC can be directly mounted on a dedicated ATCA or VME based carrier card to optimize the system cost.

5.4.1.1 Analog Part

In total, the MSADC card provides up to 32 differential analog channels, which are independently buffered by an operational amplifier stage for impedance matching and amplification, similar to the SADC design from Section 5.1.1. Due to the bipolar power

supply, signals of both polarities can be accepted up to 5 V signal amplitude. To compensate for common offset shifts of the signal baseline, all 32 channels can be biased by a 16 bit digital-to-analog converter (DAC). As the DAC can drive the full differential input amplifier range, it can also be used to generate test input signals for the ADCs.

To be able to fit 32 channels on the mezzanine card form factor, a multichannel ADC family with serial low-voltage differential signaling (LVDS) lines on the digital side is used. The ADS527x family [TI09] from Texas Instruments provides 8 differential ADC channels with up to 70 MHz sampling rate in a compact 80 pin housing, which allows to mount two ADC chips on both sides of the printed circuit board (PCB). By choosing between the different ADC types with either 10 bit or 12 bit resolution and varying sampling frequency, a trade-off between performance and system cost can be achieved.

Since some applications may require a higher sampling rate than the 70 MHz provided by the ADS527x ADC family, the PCB design allows to change the channel configuration in a way that two ADCs can work in an interleaved mode. This allows in principle to double the sampling rate up 140 MHz, however with increased effort in the FPGA firmware to correct for channel mismatches between both ADCs. The change to interleaved operation is realized by a modification in the component placement on the PCB, which reduces as well the number of usable input channels down to 16.

In addition to the differential analog inputs for detector signals, the MSADC provides also 8 single ended analog inputs for slow control functionality. These channels may be used for general purpose monitoring tasks on the detector front-end, like temperatures, supply voltages or currents which are sampled by a “slow” multichannel ADC with 24 bit resolution. All analog input signals are connected via a 120 pin high frequency connector, to ensure the analog signal quality and proper grounding of the mezzanine card.

5.4.1.2 Digital Part

The digital core of the MSADC module is based on a Xilinx FPGA from the Virtex 4 LX family, which interfaces the ADCs to the subsequent DAQ stages and implements all data processing algorithms. An important feature of this FPGA family is an included serializer/deserializer combination for every IO-pin. Therefore, the FPGA can directly connect to the serial interface of the ADCs and deserialize the LVDS data stream into parallel data words which are required for further processing in the FPGA fabric. The communication with subsequent stages in the DAQ system can be implemented in a similar way, via 20 free configurable LVDS signal pairs which are routed directly to the mezzanine connector on the digital side of the MSADC. In addition to the 2.5 V LVDS lines, 25 single ended IO lines allow to connect to equipment based on 3.3 V IO voltage. Both interfaces may be either configured for upstream data transfer or control of detector components and front-end ASICs preceding the MSADC card. In addition, dedicated clock and trigger lines ensure the proper synchronization of multiple MSADC cards which are working in parallel.

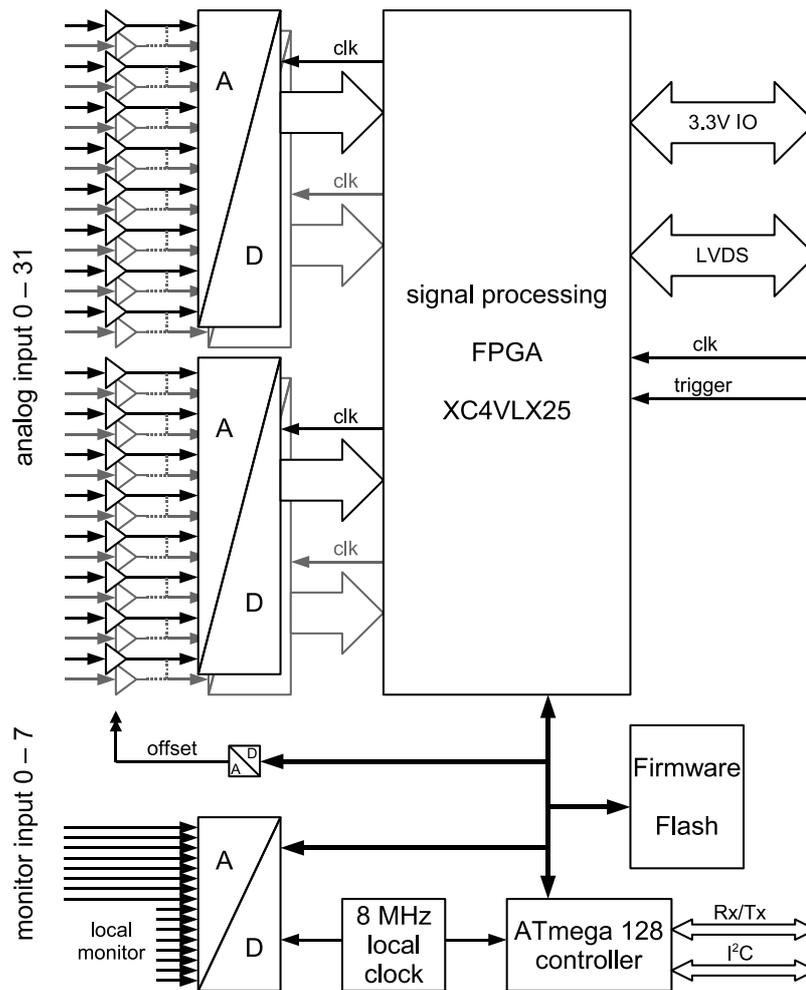


Figure 5.16: Overview of the data flow on the mezzanine sampling ADC module.

To increase the reliability of DAQ systems including a large number of MSADC modules, each MSADC is monitored by an independent microcontroller from Atmel. As all supply voltages are provided externally to the card, all six voltage lines are connected to the 24 bit slow ADC to ensure the required voltage ranges. In addition the PCB temperature is measured at three different points to ensure proper cooling. The microcontroller can interface to a carrier card via serial Rx/Tx lines and an I²C interface, which is also compatible to the management bus type of ATCA based systems.

Furthermore, the microcontroller is responsible for the initial loading of the FPGA firmware after power-up. Therefore, an on-card flash memory allows to store up to two different firmware configurations. This enables as well the verification of a loaded firmware during the normal operation. Since the internal FPGA configuration memory is susceptible to random bit-errors in environments with increased radiation level [CGJ⁺02], a continuous verification of the running firmware allows to detect such errors and to reconfigure the FPGA [Xi108] if necessary. Figure 5.16 summarizes again the functional blocks and data paths of the MSADC module.

The major part of the MSADC firmware can be reused from the SADC development described in Section 5.3.1. Since all 32 input channels are now processed within one FPGA, the functionality of all three FPGA devices on the SADC module is combined. Additional firmware blocks are implemented for the deserialization of the ADC interface and on-board communication on the MSADC card. Furthermore, the firmware for the supervisor microcontroller had to be implemented from scratch. Currently, more than 190 MSADC cards are already used in the readout system for the COMPASS ECAL2 electromagnetic calorimeter [M⁺09].

5.4.2 MSADC USB Interface

Since the MSADC card itself is not functional without an additional power supply and readout interface, a matching interface board was designed for initial tests and laboratory setups. This board provides all voltage regulators for the MSADC and implements an USB interface similar to the SADC USB interface from Section 5.1.5. To simplify the setup, the board operates from a single +12 V power supply. In addition, all LVDS lines and the JTAG¹⁷ interface for the microcontroller and the FPGA are available for initial configuration and debugging. Figure 5.17 shows the USB interface card, with the mounted MSADC module.

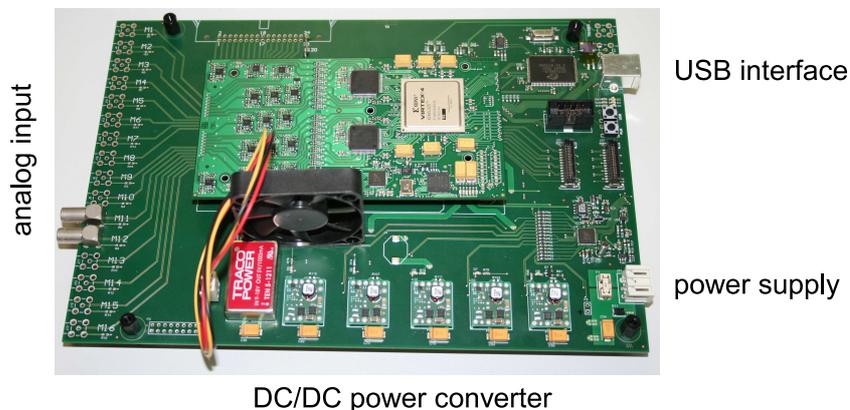


Figure 5.17: USB interface for the MSADC module.

Since the ATCA standard is based on DC/DC power converters to increase efficiency, a similar concept is realized on the USB interface. Thus, the module allows also to evaluate the influence of the switching power supplies on the signal quality of the sampled analog signal. The power consumption of a 32 channel MSADC is in the range of 12–17 W, depending on the implemented FPGA firmware.

¹⁷JTAG: Joint Test Action Group

Chapter 6

Sampling ADC Measurements

To evaluate the described sampling ADC algorithms from Chapter 4, real detector data were acquired with various detector setups. Initially, a prototype installation of the MADPET-II system was used to obtain first results with real detector data. The acquired data included the time and amplitude information, extracted within the FPGA firmware, as well as the raw data samples, to allow the evaluation of different algorithms by offline data reprocessing after the acquisition.

Besides the activities on MADPET-II, the predecessor system, MADPET-I [Zie01] (cf. Section 3.2), was reactivated for a student lab course [Sch09]. This allowed to base the new data acquisition system on the described sampling ADC modules. Further on, the system employs as well the energy and time calibration procedures presented in Section 3.3.1 and Section 3.3.3.

In addition to the existing MADPET detectors, the sampling ADC approach was also evaluated with silicon photomultiplier detectors (SiPM) as next generation photodetectors for a successor system [Spa08]. This includes also the possibility for multimodality imaging in a combined PET/MR scanner, with additional challenges for the data acquisition regarding the sensitivity to noise arising from the MRI system.

6.1 MADPET-II Measurement Setup

The MADPET-II measurement setup included 128 APD channels from a 384 channel prototype system which were read by four SADC modules. The data acquisition system was based on the COMPASS readout chain already described in Section 5.1, including the multiplexer stage, PCI-interface and clock distribution. The 128 APD detector channels were split into two opposing 64 channel segments, each including two 32 channel APD modules from the front detector layer. All detectors were illuminated by an ^{18}F -FDG plane source, placed in the center of the field of view as illustrated in Figure 6.1.

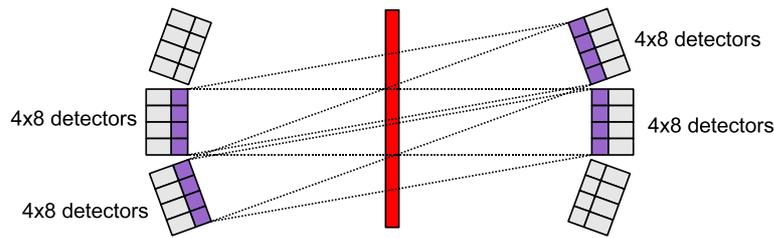


Figure 6.1: Measurement setup with 2×2 opposing MADPET-II front layer detector modules (*purple*) and the ^{18}F -FDG plane source (*red*) in the center of the field of view.

The implemented data processing in the SADC FPGA firmware included amplitude extraction and time extraction without any interpolation or upsampling of the sampled ADC data. The amplitude extraction was realized by the peak-hold method (cf. Section 4.3.3.1) and the time extraction was realized by the digital CFD algorithm (cf. Section 4.4.2) with the default hardware CFD threshold set to 40%. The processing results were included in the output data stream, together with the initial sampled ADC data and stored to disk. In total, 14×10^6 events with raw ADC data samples included were acquired.

6.2 Data Analysis

From the acquired data, the performance of the firmware algorithms can be evaluated by determining the achievable energy and time resolution of the system. In addition, the acquired raw data samples allow to optimize the CFD threshold settings to further improve the achievable time resolution.

6.2.1 Amplitude Extraction

As a first step, the achievable energy resolution can be extracted from the acquired data. This is realized by the same procedure as used for the energy calibration in Section 3.3.1, which histograms the baseline corrected amplitude values and fits two Gaussian functions for the Compton and photopeak contributions. The calculation formula for the energy resolution of the photopeak is given by (3.2) and (3.3).

The baseline correction is realized for each event individually by subtracting the baseline value, which is determined by the SADC trigger logic (cf. Section 5.3.1.1), from the corresponding amplitude value in the event. Figure 6.2 shows the energy resolution for each APD detector channel, obtained from the extracted amplitude information by the peak-hold FPGA firmware. The achieved values are comparable with results from the ASIC based data acquisition system reported in [Tap08] and [Spa08]. The resulting mean energy resolution is 14.8% FWHM. Figure 6.3 gives the positions of the 511 keV photopeaks in ADC units for each APD channel. These values correspond to the energy calibration coefficients, which are used later on for applying energy thresholds to the data.

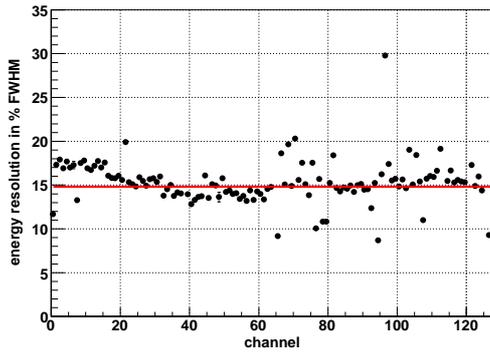


Figure 6.2: Energy resolution of the 128 MADPET-II channels acquired, determined with the peak-hold amplitude extraction algorithm in the SADC firmware. The mean energy resolution is 14.8% FWHM.

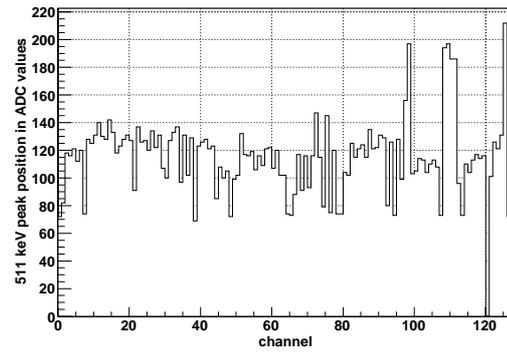


Figure 6.3: Position of the 511 keV photopeak in ADC values for each channel, after subtraction of the baseline value from the peak-hold value.

The variation in the photopeak position is mainly caused by the differing amplification of the APDs, which has a strong dependence on temperature and the applied bias voltage. Channel 120 is a completely dead channel, where no useful energy information could be extracted from the data. As the prototype setup was reusing SADC modules initially produced for the electromagnetic calorimeter of COMPASS, the amplification settings on the SADC allowed to use only up to 20% of the dynamic range of the 10 bit ADC.

To compare the performance of the other algorithms for the amplitude extraction, which were described in Section 4.3.3, the acquired raw ADC samples were reprocessed with a software version of the algorithms. The resulting data files were then analyzed in the same way as the data acquired from the SADC firmware processing. Table 6.1 lists the results for the parabola and spline based interpolation methods.

Algorithm	Mean energy resolution (FWHM)
peak-hold	14.8%
parabola	14.9%
spline $M = 8$	14.9%
spline $M = 16$	14.8%
smooth spline $M = 1, \lambda = 3$	14.9%
smooth spline $M = 8, \lambda = 3$	14.8%

Table 6.1: Mean energy resolution of the MADPET-II setup for different amplitude extraction/interpolation algorithms.

Obviously, the extracted energy resolution can not be improved by using more advanced interpolation algorithms, which indicates that it is mainly limited by the detector performance. Thus, the implementation of the resource-efficient peak-hold algorithm provides a sufficiently precise value for the amplitudes of the incoming signal pulses.

6.2.2 Time Extraction

Similar to the amplitude extraction, the time extraction algorithms can also be simulated based on the sampled pulse data. This allows to compare the different algorithm options and to tune the parameter settings for optimal performance. The proper configuration of the firmware parameters can also be monitored by the behavior of the extracted time values.

The digital CFD algorithm which is implemented in the FPGA firmware provides two different time informations: a *coarse-time* value t_c and a *fine-time* value t_f , corresponding to the data-flow schematic in Figure 4.33 and Equations (4.65) and (4.66). Both values are available as 10 bit numbers in the output data stream of the SADC modules. In addition, a 30 bit timestamp t_{trig} is available from the trigger decision of the corresponding analog input channel. Since the average-window based trigger decision results in slightly varying timestamps, depending on the signal amplitude (cf. Section 4.2.1 and Section 5.3.1.1), the coarse-time t_c corrects for this shift of the sampled signal. To further increase the time resolution, the fine-time t_f gives the interpolated time between two sampling points.

To simplify the data handling, all values are represented in integer values and normalized to the ADC sampling interval. The fine-time value therefore uses the full 10 bit range to represent the interpolated time between two sampling points. The “real” time information of an event can then be calculated with the knowledge of the ADC sampling interval t_s by

$$(6.1) \quad t = \left(t_{\text{trig}} + t_c + \frac{t_f}{2^{10}} \right) \cdot t_s.$$

Additionally, the limited range of the trigger timestamp t_{trig} must be considered. For a sampling interval of $t_s = 12.86 \text{ ns}$ (which is derived from the TCS clock frequency as depicted in Section 5.1.3), the timestamp value overflows after $12.86 \text{ ns} \cdot 2^{30} \approx 13.8 \text{ s}$. This overflow must be detected by the analysis software to achieve a continuously increasing time value for the measurement data.

Figure 6.4 and 6.5 show the histogrammed coarse-time and fine-time values for all acquired events. With a proper selection of the pretrigger buffer (cf. Section 5.3.1.1) the acquired signal pulse is shifted to a position which results in all extracted t_c values within the acquired data window, indicated by $t_c > 0$. For the given default CFD threshold value of 40% this condition is fulfilled, but the safety margin in Figure 6.4 is quite small, which causes problems for smaller F_{CFD} values.

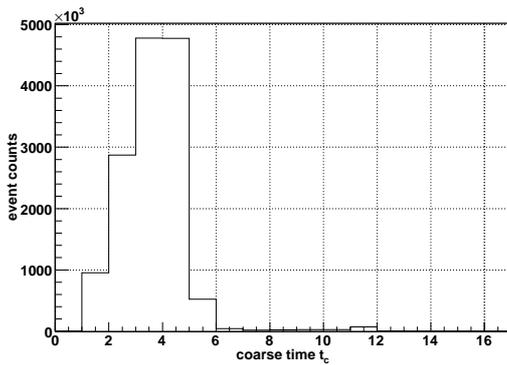


Figure 6.4: Typical distribution of the coarse-time t_c values ($F_{CFD} = 40\%$). All-most all extracted time values are within the sampled data window ($t_c > 0$).

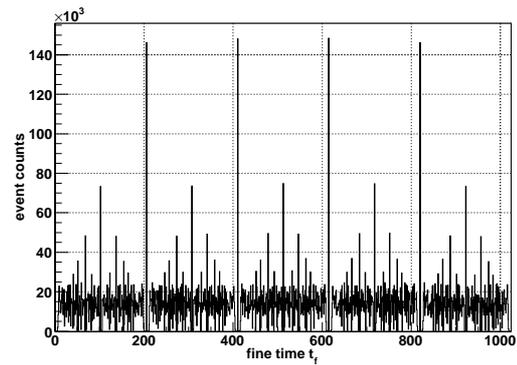


Figure 6.5: Typical distribution of the fine-time t_f values ($F_{CFD} = 40\%$). The peak structure is caused by the integer quotient in the interpolation algorithm.

The determined fine-time values t_f in Figure 6.5 show a distinct peak structure, which is caused by the integer division of the amplitude values in the algorithm (cf. Equation (4.66)). However, this effect can be neglected, since the time width of each histogram bin is sufficiently small. For the given sampling interval of $t_s = 12.86$ ns, each bin corresponds to $t_s/2^{10} \approx 12.56$ ps, which is by far lower than the minimum achievable time resolution of the LSO scintillator.

Another important parameter is the uniformity of the fine-time distribution. As motivated in Section 4.1.3.2, the extracted t_f values should be uniformly distributed over the full sampling interval, since there is no correlation between the initial radioactive decay and the ADC sampling clock. Nevertheless, the acquired t_f values can differ from this prerequisite, due to the pulse shape and baseline effects described in Section 4.4.2. Depending on the constant fraction threshold F_{CFD} , these effects will either shift the distribution to lower or higher t_f values, resulting in an underestimated time resolution due to degradation of the global time-difference histogram. This effect is illustrated in Figure 6.6 which shows the t_f distribution for different constant fraction threshold settings.

This non-uniformity effect can be corrected by a dynamic rebinning of the t_f values, with the aim to achieve again a uniform t_f distribution. The rebinned mapping of the t_f values can then be used as a lookup-table for the determined t_f values. By grouping multiple t_f values together, the minimum time resolution of the digital CFD algorithm is reduced. However, since a time resolution below 100 ps is hardly possible for LSO based detectors—which is due to the long decay constant of the scintillation light (see Table 2.2)—a rebinning factor of 10–15 is acceptable. Figure 6.7 shows the rebinned fine-time distribution for different CFD threshold settings, which is more uniform compared to the uncorrected data in Figure 6.6.

The rebinning procedure calculates the average a of all histogram bin entries and determines the new average value \tilde{a} after the rebinning operation. Since the fine-time histogram is squeezed by the rebinning, the new average value is given by

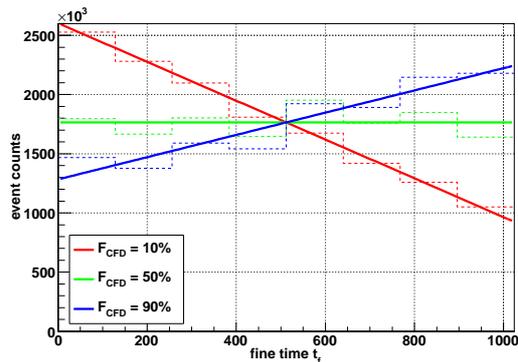


Figure 6.6: Distribution of the fine-time t_f values for different constant-fraction thresholds. Extreme CFD threshold values cause a severe non-uniformity of the distribution. (To suppress the peak structure from Figure 6.5 the bin-width is set to 128.)

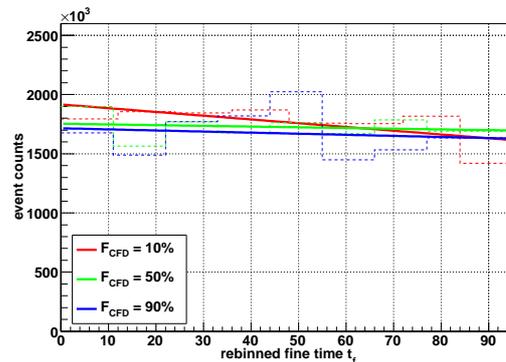


Figure 6.7: Rebinned distribution of the fine-time t_f values for different constant-fraction thresholds. The non-uniformity from Figure 6.6 is almost removed.

$$(6.2) \quad \check{a} = a \cdot \frac{t_f^{(\max)}}{\check{t}_f^{(\max)'}}$$

with $\check{t}_f^{(\max)'}$ being the new range for the rebinned t_f values. In a second step, a rebin lookup-table is generated by successively accumulating histogram bin entries until the new average value is exceeded. This process is executed until the full t_f input range is mapped to the squeezed \check{t}_f number range. Due to the distinct peak structure in the distribution of the initial t_f values, it is usually not possible to meet exactly a specified $\check{t}_f^{(\max)'}$ value. Instead, the final range is slightly smaller than specified, which must be considered in the calculation of the “real” time value according to (6.1). This causes also the different bin widths noticeable in Figure 6.7.

Figure 6.8 repeats the CFD threshold optimization from Figure 4.37 in Section 4.4.2. Therefore, the available 32 raw ADC sampling points are processed by a software implementation of the CFD algorithm to create a similar event-dataset as if acquired directly with the SADC based system. The event data for each specific CFD threshold setting is then processed with the fine-time rebinning method to guarantee a uniform t_f distribution. Finally the dataset is evaluated by the time calibration software (cf. Section 3.3.3.2), which gives a measure for the width of the global time difference histogram before and after calibration.

The problem of the too small safety margin for the pretrigger buffer is noticeable in Figure 6.9, which shows the fit quality (as the χ^2/ndf value¹) for the estimation of the time resolution based on the global time difference histogram. Ideally, this value should approach unity for a perfectly matching fit result. For an increased disagreement between

¹**ndf:** Number of Degrees of Freedom

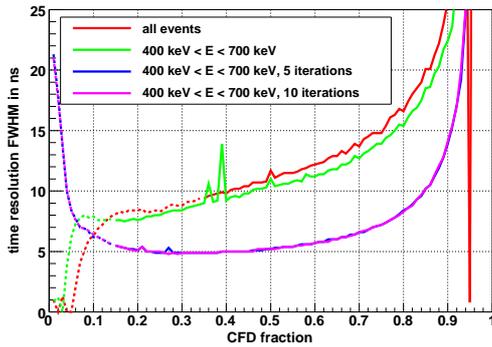


Figure 6.8: Time resolution for different constant fraction settings. Evident is the difference before (*red, green*) and after (*blue, purple*) time calibration. The dotted regions are not useful, since the fit quality is degraded, as shown in Figure 6.9.

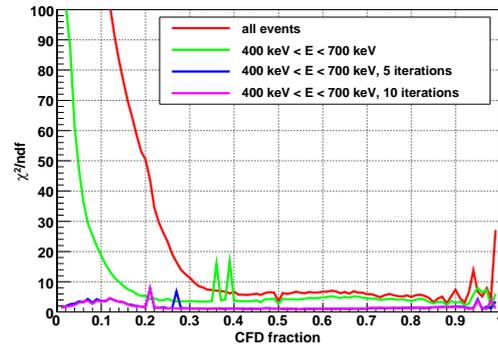


Figure 6.9: Fit quality for the determination of the time resolution from the global time difference histogram. For low F_{CFD} values the performance degrades due to missing pulse data.

the shape of the time difference histogram and the assumed shape in the fit-function the value increases.

If the first part of the signal waveform is missing due to a too small pretrigger buffer, the fit quality decreases for small F_{CFD} values. This effect is especially obvious if events with low amplitude are included in the analysis, since these events are usually closer to the beginning of the sampled data window (cf. Section 4.2 and Section 5.3.1.1). Therefore, CFD threshold settings with $F_{\text{CFD}} < 35\%$ are not useful for the analyzed dataset. If the low amplitude events are suppressed by an applied energy cut ($400 \text{ keV} < E < 700 \text{ keV}$), this limit can be reduced down to about $F_{\text{CFD}} < 15\%$.

After the time calibration of the system, the fit quality and the time resolution improve significantly. From Figure 6.8, a minimum time resolution of about 4.9 ns FWHM can be determined after five time calibration iterations, which cannot be improved by further iterations. Table 6.2 summarizes the achieved time resolutions for the CFD algorithm with different interpolation options. Therefore, the FWHM time resolution, extracted from the time difference histogram, and the corresponding standard deviation as $\sigma \approx \text{FWHM}/2.35$ is listed. As the time difference histogram combines the measurement data from two detectors, the time resolution for a single detector channel is determined by $\sigma_{\text{single}} = \sigma/\sqrt{2}$.

To compare the results from Table 6.2 with the simulations in Chapter 4, the signal to noise ratio (SNR) of the measurement data can be calculated by using Equation (4.61)

$$\text{SNR} = \frac{\text{mean signal amplitude}}{\text{std. dev. of noise}} = \frac{\overline{A_{\text{signal}}}}{\sigma_{\text{noise}}}$$

Since the time reconstruction is only operating on a selected energy range around the 511 keV photopeak ($400 \text{ keV} < E < 700 \text{ keV}$), the mean signal amplitude $\overline{A_{\text{signal}}}$ is given

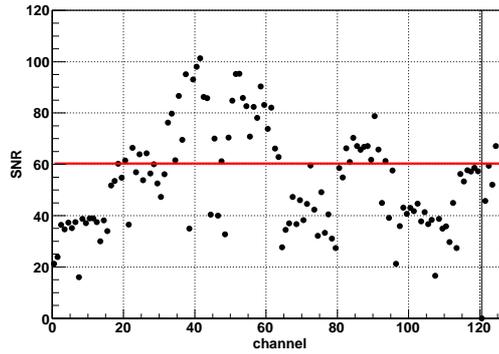


Figure 6.10: Signal to noise ratio for each channel of the MADPET-II setup. The average value is $\overline{SNR} \approx 60$.

Algorithm	Minimum time resolution			F_{CFD}
	FWHM	σ	$\sigma/\sqrt{2}$	
raw data	4.9 ns	2.1 ns	1.5 ns	$\sim 30\%$
spline $M = 8$	4.8 ns	2.0 ns	1.4 ns	$\sim 32\%$
spline $M = 16$	4.8 ns	2.0 ns	1.4 ns	$\sim 30\%$
smooth spline $M = 1, \lambda = 3$	5.3 ns	2.3 ns	1.6 ns	$\sim 44\%$
smooth spline $M = 8, \lambda = 3$	5.2 ns	2.2 ns	1.6 ns	$\sim 40\%$

Table 6.2: Minimum time resolution of the MADPET-II setup for the CFD based time extraction with different interpolation options. The F_{CFD} column indicates the constant fraction threshold setting for the best time resolution.

by the photopeak position, which was already determined for the energy calibration. The standard deviation of the signal noise can be extracted by histogramming and fitting the first raw signal samples. For the given data with very small t_c values, only the first signal sample was analyzed for each channel, as the signal pulse is rising very close to the first data sample. Figure 6.10 shows the measured SNR values for all acquired channels. The average signal to noise ratio is $\overline{SNR} \approx 60$.

When compared to the simulation results in Section 4.4.2, the measured time resolution is not as good, as expected from the simulation. This can be explained by two reasons: Due to the small t_c values, some events may not include the full rising edge of the signal, since the event recording started too late. This influences especially the spline interpolation routines, as the first interpolation points cannot be properly initialized, which leads to a degradation of the rising edge. A second source for the degradation of the time resolution is the jitter of the SADC clock distribution, which can reach up to 0.4 ns (cf. Section 5.1.3).

6.3 MADPET-I Measurement Setup

Besides the first evaluation measurements with the MADPET-II detector, the predecessor system, MADPET-I, was reactivated to create a lab course for students as an introduction into positron emission tomography [Sch09]. As the initial CAMAC/NIM based readout system of MADPET-I was no longer existing, a new data acquisition system was required. Thus, an SADC based readout system was realized to digitize the 96 APD channels of MADPET-I. Figure 6.11 gives an overview of the MADPET-I detector system.

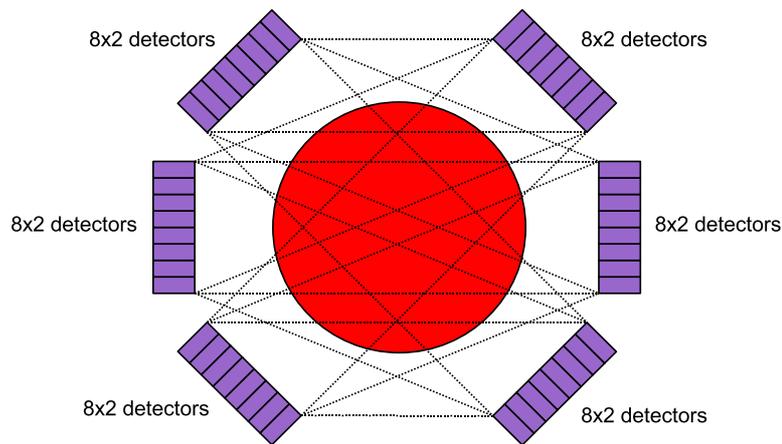


Figure 6.11: Detector setup with the 3×3 opposing MADPET-I detector modules (*purple*). The lab course uses different source geometries, which are placed in the marked field of view region (*red*).

6.3.1 MADPET-I Hardware Overview

The MADPET-I scanner consists of six detector modules, which are arranged at the sides of an octagon. Each detector module includes an 8×2 APD array with an area of $4 \times 4 \text{ mm}^2$ for each APD channel. The APDs are individually coupled to LSO scintillator crystals with a size of $3.7 \times 3.7 \times 12.0 \text{ mm}^3$. To enhance the light collection, the scintillator crystals are embedded in a PTFE² matrix with high reflectivity. Each APD channel is amplified by a discrete preamplifier circuit and then connected to the readout electronics. Since there are no ASICs in the signal chain, the setup does not reach the integration level of MADPET-II and has also a higher power consumption. The scanner gantry with the six detector modules is shown in Figure 6.12. In the center of the scanner the radioactive source is placed on a rotating table, to cover missing projections for the image reconstruction.

The readout system is shown in Figure 6.13. It includes the low-voltage power supplies for the preamplifiers and the SADC system, as well as a controllable high-voltage power supply to generate the bias voltage for the APDs. The 96 APD channels are digitized by

²PTFE: Polytetrafluoroethylene, “Teflon®”

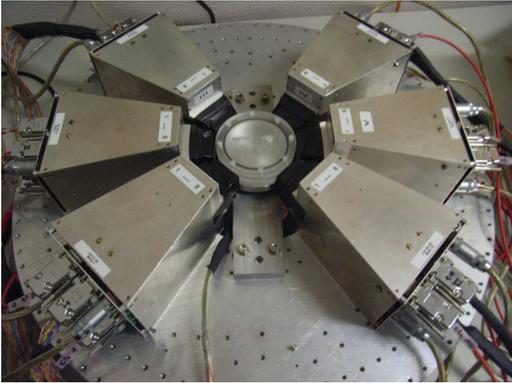


Figure 6.12: MADPET-I detector front-end with radioactive source placed on a rotating table in the field of view. [Sch09]



Figure 6.13: Electronic system for the MADPET-I lab course, including all power supplies (low-voltage, high-voltage), rotating table control and three SADC modules with USB readout (*right*). [Sch09]

three SADC modules, which are connected to the readout PC via the SADC USB interface (cf. Section 5.1.5). A common 40 MHz clock is distributed by the first SADC to all other modules via a backplane connection to achieve synchronous operation between all SADCs. The 80 MHz ADC sampling clock is then derived from this master clock on each module.

6.3.2 Data Analysis

Similar measurements as for the MADPET-II setup were obtained with the new data acquisition system during the commissioning phase [Sch09]. For this comparison, 25×10^6 events were acquired with an ^{18}F -FDG source. As the MADPET-I detectors use a different preamplifier module, the pulse shape is different compared to the simulated MADPET-II system. Figure 6.14 shows the normalized and time aligned pulse shapes from all detectors. The pulse rise and fall times are about $t_r \approx 55$ ns and $t_f \approx 150$ ns. The signal to noise ratio for all channels is shown in Figure 6.15, with an average value of $\overline{SNR} \approx 94$. The mean energy resolution for the different extraction algorithms is listed in Table 6.3.

For the shorter MADPET-I signal, only the smoothing spline interpolation algorithm allows to enhance the energy resolution. Since some preamplifier signals include oscillating noise in the frequency range of 20–65 MHz, the direct interpolation methods are not sufficient. The oscillation of the preamplifiers is partly caused by a non-optimal grounding scheme of the detector front-end and the high capacitance of the APDs. The trigger logic in the SADC is able to suppress this noise and allows to trigger only on valid signal pulses, however, the acquired signal pulse still includes the noise. This effect is also visible as the regular structure in the pulse shape of Figure 6.14.

The results for the time extraction measurements are listed in Table 6.4. Again, due to the specific noise properties, only the smoothing spline interpolation allows some im-

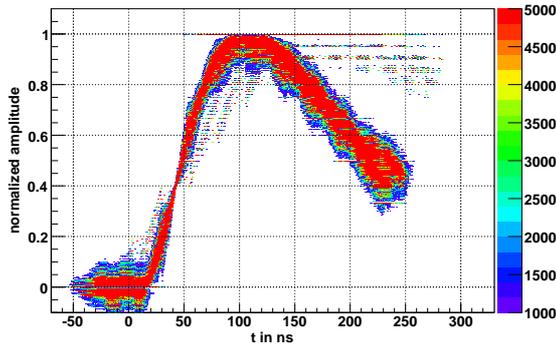


Figure 6.14: Normalized and time aligned MADPET-I detector pulses, acquired with the SADC system.

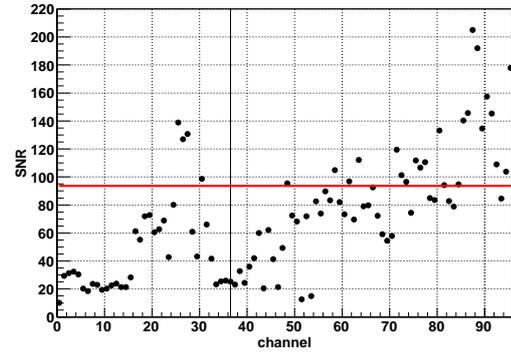


Figure 6.15: Signal to noise ratio for the MADPET-I detector front-end. The average value is $SNR \approx 94$.

Algorithm	Mean energy resolution (FWHM)
peak-hold	11.8%
parabola	11.9%
spline $M = 8$	12.0%
spline $M = 16$	12.1%
smooth spline $M = 1, \lambda = 3$	9.5%
smooth spline $M = 8, \lambda = 3$	10.3%

Table 6.3: MADPET-I mean energy resolution for different amplitude extraction/interpolation algorithms.

Algorithm	Minimum time resolution			F_{CFD}
	FWHM	σ	$\sigma/\sqrt{2}$	
raw data	4.4 ns	1.9 ns	1.3 ns	$\sim 30\%$
spline $M = 8$	4.8 ns	2.0 ns	1.4 ns	$\sim 29\%$
spline $M = 16$	4.8 ns	2.0 ns	1.4 ns	$\sim 28\%$
smooth spline $M = 1, \lambda = 3$	4.1 ns	1.7 ns	1.2 ns	$\sim 32\%$
smooth spline $M = 8, \lambda = 3$	4.2 ns	1.8 ns	1.3 ns	$\sim 30\%$

Table 6.4: MADPET-I minimum time resolution for the CFD based time extraction with different interpolation options. The F_{CFD} column indicates the constant fraction threshold setting for the best time resolution.



Figure 6.16: Sample source for the lab course with the TUM logo as active region. The source is milled from plexiglas and can be filled with ^{18}F -FDG. [Sch09]

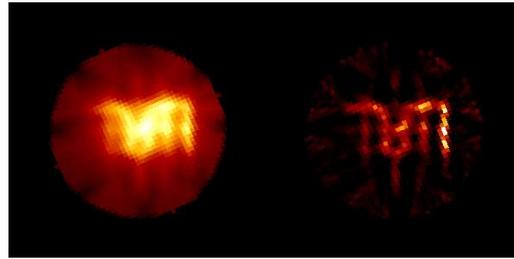


Figure 6.17: Reconstructed images of the TUM source, using simple backprojection (*left*) and filtered backprojection (*right*). [Sch09]

provement. Since the simple spline interpolation is also interpolating the noise, the time resolution is degrading for this case.

An example of the image reconstruction with the updated MADPET-I system is shown in Figure 6.16 and 6.17. Therefore, a source for the lab course with the TUM logo as active area was filled with 0.7 MBq of ^{18}F -FDG and data were acquired. Since the six detector modules do not cover all necessary projections to allow a successful reconstruction, the source was successively rotated in steps of 45° . For each of the eight projections, data were acquired for 60 sec. The lab course then requires the students to do all calibration and analysis steps as described in the previous chapters. After the energy and time calibration, the acquired data are processed by an image reconstruction program, which implements simple backprojection and filtered backprojection as the two simplest reconstruction methods (cf. Section 2.3). The final reconstructed images are shown in Figure 6.17.

6.4 Silicon Photomultiplier Measurement Setup

The recent improvements in the field of photon detection devices, which led to the detector concept of *silicon photomultipliers* (SiPM)³, were already discussed in Section 2.2.2. Since SiPM detectors combine the advantages of photomultiplier tubes and semiconductor based APDs—namely the good timing properties and the insensitivity to magnetic fields—they gain more and more importance in positron emission tomography.

6.4.1 Hardware Overview

For first evaluation measurements, sample devices of the *Multi Pixel Photon Counters* (MPPC) from Hamamatsu Photonics were available [HP09a, HP09b]. The active area

³Silicon photomultiplier (SiPM) is used here as a reference to all detectors which use the concept of light detection by an array of Geiger-mode APD cells or a similar operation principle. Usually, the devices on the market may use different names since the manufacturers are using their own registered names/trademarks for their respective product.

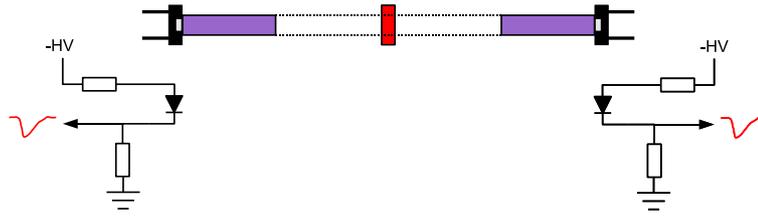


Figure 6.18: Measurement setup for the SiPM/MPPC evaluation. Two SiPM detectors are directly coupled to LYSO scintillator crystals (purple), with the radioactive source (red) placed between both detectors.

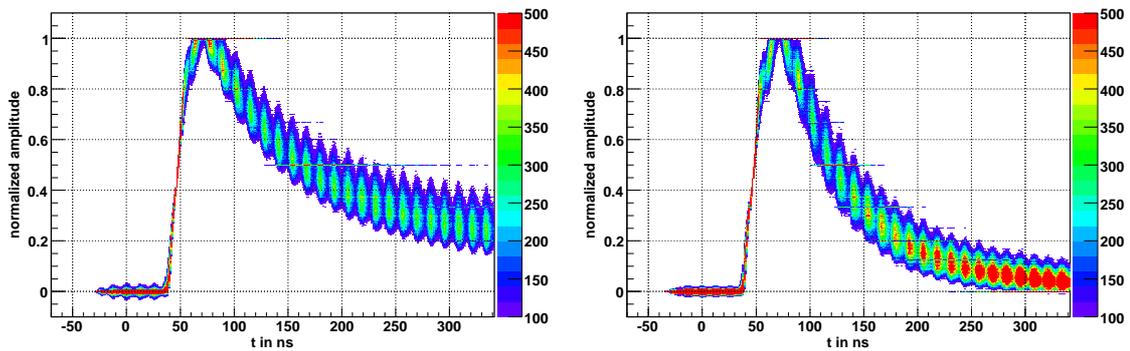


Figure 6.19: Normalized and time aligned signal pulses from 400 cell (left) and 1600 cell (right) MPPCs, glued to LYSO scintillator crystals and irradiated with a ^{22}Na source. The SADC sampling rate is $f_s = 77.76$ MHz.

of these detectors was $1 \times 1 \text{ mm}^2$ with either 400 cells ($50 \times 50 \mu\text{m}^2$, S10362-11-050C) or 1600 cells ($25 \times 25 \mu\text{m}^2$, S10362-11-025C). Each MPPC was independently packaged in a ceramic carrier with an epoxy cover on the active area. This allowed to glue the $1.4 \times 1.4 \times 12 \text{ mm}^3$ LYSO scintillator crystals directly onto the epoxy cover. Figure 6.18 shows the measurement setup for the MPPC evaluation.

Due to the high intrinsic gain of the MPPC devices, in the order of 10^5 – 10^6 [HP09b], the MPPC signal can be accessed by the simple circuit from Figure 6.18, without an additional preamplifier. The achievable signal amplitudes are in the range of 50–100 mV, depending on the incident light, the number of cells in the MPPC device and the resistor values. As an increased number of cells reduces the size of each individual cell (for a fixed total active area), the resulting signal pulses for high cell count devices are usually smaller due to the reduced cell gain. However, the linearity between the incoming photon flux and the electrical output signal can be improved, since more detector cells are available and contribute to the analog signal output.

Figure 6.19 shows the normalized pulse shapes from a 400 cell and 1600 cell MPPC/LYSO module, irradiated with a ^{22}Na source and sampled with a 77.76 MHz SADC module. The good timing properties of SiPM detectors are based on the relatively large current flow caused by the Geiger-mode discharge of the cells, which results in a fast rising signal

edge of only few nanoseconds duration. For a SiPM/LYSO-scintillator assembly, the observable signal rise time is in the range of about $t_r \approx 10\text{--}30$ ns, due to the scintillator light decay.

Since the measured rise time of the raw SiPM signal is $t_r < 20$ ns, only one to two ADC sample points are available on the rising signal edge for a sampling rate of $f_s = 77.76$ MHz. Thus, the CFD time extraction algorithm is biased, as the linear interpolation has mostly only one real sample point on the rising signal edge. This leads to the characteristic structure in the signal plots of Figure 6.19, as certain t_f values are artificially enhanced (cf. Section 4.4.2). Further on, the signal amplitude is higher for the 400 cell detector, which leads to a longer decay tail in the normalized amplitude plot. To avoid additional amplification stages, only the 400 cell devices were used for further measurements, since they provide already a sufficiently high signal amplitude for the SADC readout.

The typical achievable energy resolution is in the order of $(\Delta E/E)_{\text{FWHM}} \approx 12\%$ for 400 cells and $(\Delta E/E)_{\text{FWHM}} \approx 21\%$ for 1600 cells. However, the 400 cell SiPM data are influenced by nonlinearity effects, due to the very limited number of detector cells compared to the incoming photon flux, which leads to an artificial reduction of the measured energy resolution. Additionally, the achievable values are closely related to the optical coupling between the scintillator crystal and the detector. A detailed evaluation of the amplitude properties and temperature dependence of SiPMs with different cell numbers can be found in [Für09], who utilized also the SADC based data acquisition for some measurements.

6.4.2 Measurement Results

6.4.2.1 Amplitude Resolution and Time Resolution

To evaluate the applicability of the SiPM detectors for a combined PET/MR setup, data were acquired with two measurement setups. First, the setup from Figure 6.18 was operated in a normal lab environment with an ^{18}F -FDG source and data were acquired with a sampling ADC module ($f_s = 77.76$ MHz). In a second step, the same unshielded setup was placed inside a clinical 1.5 T MR scanner (Philips Achieva) and data were acquired while the MR system was running a typical brain scan protocol. Since the MR setup required to place the SADC modules outside of the shielded scanner room, the SiPM signals were routed via 10 m of 50Ω coaxial cables to the remote SADC. Table 6.5 lists the energy resolution for both setups, obtained with the different amplitude extraction and interpolation algorithms.

As the lab setup could not be operated in a completely dark environment, the remaining background light causes the degraded energy resolution due to additional background noise. In contrast, the MR measurements could be carried out in a completely darkened room, which is also reflected in the increased SNR value. As a result of the short signal

Algorithm	Mean energy resolution (FWHM)	
	Lab setup	MR setup
	(SNR \approx 226)	(SNR \approx 337)
peak-hold	17.1%	14.7%
parabola	18.0%	14.7%
spline $M = 8$	17.2%	14.6%
spline $M = 16$	17.2%	14.6%
smooth spline $M = 1, \lambda = 3$	14.6%	13.8%
smooth spline $M = 8, \lambda = 3$	14.5%	13.8%

Table 6.5: Hamamatsu 400 cell MPPC mean energy resolution for different amplitude extraction/interpolation algorithms in the lab setup (6.49×10^6 events) and MR scanner setup (4.79×10^6 events).

	Algorithm	Minimum time resolution			F_{CFD}
		FWHM	σ	$\sigma/\sqrt{2}$	
Lab setup	raw data	2.4 ns	1.0 ns	0.72 ns	$\sim 12\%$
	spline $M = 8$	1.9 ns	0.81 ns	0.57 ns	($\sim 2\%$)
	spline $M = 16$	1.9 ns	0.81 ns	0.57 ns	($\sim 3\%$)
	smooth spline $M = 1, \lambda = 3$	2.9 ns	1.2 ns	0.87 ns	$\sim 10\%$
	smooth spline $M = 8, \lambda = 3$	2.9 ns	1.2 ns	0.87 ns	$\sim 10\%$
MR setup	raw data	2.4 ns	1.0 ns	0.72 ns	$\sim 16\%$
	spline $M = 8$	1.7 ns	0.72 ns	0.51 ns	($\sim 3\%$)
	spline $M = 16$	1.7 ns	0.72 ns	0.51 ns	($\sim 3\%$)
	smooth spline $M = 1, \lambda = 3$	3.0 ns	1.3 ns	0.90 ns	$\sim 6\%$
	smooth spline $M = 8, \lambda = 3$	3.0 ns	1.3 ns	0.90 ns	$\sim 8\%$

Table 6.6: Hamamatsu 400 cell MPPC minimum time resolution for different interpolation algorithms in the lab setup (6.49×10^6 events) and MR scanner setup (4.79×10^6 events). The F_{CFD} column indicates the constant fraction threshold setting for the best time resolution.

pulse, only the smoothing of the signal waveform allows an improvement of the energy resolution, as it provides some averaging of the samples around the peak value.

Since both detector devices are placed at opposite sides of the radioactive source, two coincident 511 keV photons from a positron electron annihilation can be measured. The calculation of the time difference for all registered events in the opposing detectors provides then a figure for the time resolution of the detector setup. Again, the assumption of a uniform distribution of the detector signal time shift ϕ to the ADC clock—resulting in a uniform fine time t_f distribution of the CFD output values—is critical to obtain useful results. As already mentioned, certain t_f values are artificially enhanced for the SiPM measurements due to the fast rising signal edge. Thus, the t_f values have to be corrected by the fine time rebinning method from Section 6.2.2 before the time difference calculation. Table 6.6 shows the achieved time resolution for both measurement setups after t_f rebinning.

Compared to the APD based detectors of the MADPET systems, the achieved time resolution of the SiPM based detectors is about twice better. However, the SiPM setup is only evaluating two coincident SiPM channels, whereas the MADPET measurements include multiple front-end channels (48 + 48 channels for MADPET-I and 64 + 64 channels for MADPET-II). Thus, the SiPM data includes no ambiguities from channel delay mismatches as only two channels are evaluated. Nevertheless, the SiPM data can be acquired without any preamplifier close to the detector as the SiPM signal has already sufficiently high amplitude. This would allow a much simpler front-end setup as currently used in the MADPET systems.

The different measurement setups in the lab and the MR scanner system give almost similar results for the time resolution. The slight increase in the raw data threshold value for the MR setup can be explained by the longer readout cables (10 m), which slow down the rising signal edge due to the damping of high frequency components. This effect is even more distinct in the smoothing spline preprocessing, which leads to a worse time resolution due to the degradation of the rising signal edge. The strong magnetic field of 1.5 T and the simultaneous operation of the MR scanner show however no impact on the time resolution performance.

The strong influence of the fast rising signal edge on the time resolution is indicated by the low constant fraction threshold settings F_{CFD} , which are around 10%. This is in contrast to the 30–40% for the slow APD signals in the MADPET measurements. The very low threshold (2–4%) and time resolution values for the spline interpolation are however caused by rounding artifacts from the division operation in the CFD algorithm. This results in a regular peak structure in the fine time values as already shown in Figure 6.5, which cannot be fully corrected by the fine time rebinning procedure. Therefore, a similar structure remains also in the rebinned t_f data, which leads to a more narrow time difference histogram than expected for uniformly distributed values. Thus, the obtained time resolution values and the t_f distribution have to be checked carefully, once the “optimal” threshold setting is reaching extreme values ($F_{CFD} \ll 10\%$).

6.4.2.2 MR Interference

Another critical point of the measurement setup is the influence of the running MR system on the PET data acquisition. From the obtained resolution values for amplitude and time in the previous Section 6.4.2.1, it can already be concluded that the PET system can be operated together with the MR system. This is in particular remarkable since the detector setup was placed completely unshielded into the MR scanner system and was thus fully exposed to the harsh electromagnetic environment during the running MR scans.

The main frequency component in the MR system is determined by the magnetic field strength B_0 of the scanner system and the gyromagnetic ratio γ of protons (${}^1_1\text{H}^+$) which are the main imaging target for clinical MRI. The MR frequency can be calculated by

$$(6.3) \quad \begin{aligned} f_{\text{MR}} &= \frac{\gamma}{2\pi} \cdot B_0, \\ &= 42.58 \frac{\text{MHz}}{\text{T}} \cdot B_0. \end{aligned}$$

For the Philips Achieva MR system, the magnetic field is $B_0 = 1.5 \text{ T}$, which results in an MR frequency of $f_{\text{MR}} = 63.86 \text{ MHz}$. This frequency is transmitted by the MR system to flip the precessing hydrogen atoms in the imaging object out of their alignment with the main magnetic field B_0 . In a second step, the relaxation time of the individual hydrogen spins to the B_0 field orientation can be measured and is used for the MR image reconstruction. To be able to address the complete imaging volume of the MR system, the magnetic field in the volume is slightly varied, which has however only a minor influence on f_{MR} in the range of few 100 kHz. Thus, the main electromagnetic noise component is around 63.86 MHz for the 1.5 T MR system.

Due to the high transmitted radio frequency (RF) power, the noise is picked up by both SiPM detector assemblies and is routed via the signal cables to the SADC system. As the SADC is sampling with $f_s = 77.76 \text{ MHz}$, the MR frequency violates the sampling theorem (4.7) which results in undersampling of the 63.86 MHz noise. The resulting frequency in the ADC data is therefore $f_{\text{MR}}^{(\text{ADC})} = f_s - f_{\text{MR}} \approx 13.9 \text{ MHz}$. However, the regular RF oscillation is ideally suppressed by the implemented trigger algorithm (cf. Section 4.2). This is illustrated in Figure 6.20, which shows the average time between two consecutive acquired events. During the MR activity, this time value is increased, which corresponds to a reduction of the trigger rate. Since the received RF signal has a higher amplitude compared to the SiPM signal, no detector signals can be acquired during an RF transmission. Once the RF signal disappears, the acquisition can immediately continue, which results only in a small reduction of the overall PET event statistics.

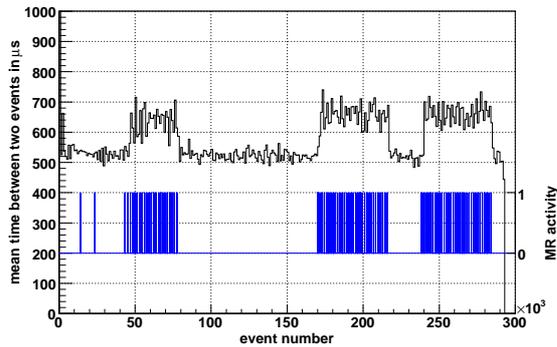


Figure 6.20: Mean time between two consecutive events acquired (*black*) and concurrent MR RF activity (*blue*). The MR activity curve can be extracted from the SADC data and indicates each engaging of a RF transmission.

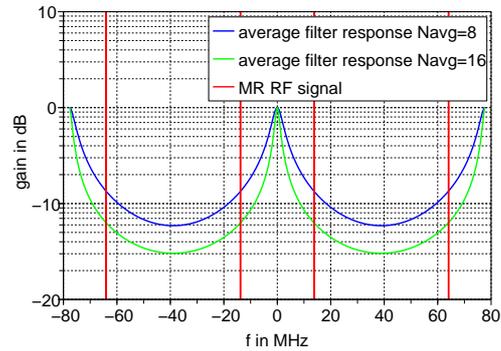


Figure 6.21: Attenuation of the average filter in the trigger decision for different averaging length settings (*blue, green*) and position of the real and aliased MR frequency (*red*).

Figure 6.21 shows the behavior of the average filter in the trigger stage of the SADC firmware. Depending on the number of averaged samples N_{avg} , the accepted input frequency range can be modified to match the spectrum of the expected detector signals (cf. Figure 4.6). This allows a sufficient damping of the RF frequency peaks in the spectrum to a level which does not exceed the trigger threshold setting.

In addition, the trigger scheme allows to extract information about the running MR sequences. The threshold decision usually accepts also events with emerging RF pulses as they cause a step in the signal baseline level of the ADC input. This information can then be used to correlate the continuously running PET measurement with the MR scan protocol. Since current MR systems allow already real-time imaging of the heartbeat [F⁺06], this feature can be useful for e.g. motion correction of the acquired PET data. Especially for “PET insert” detectors which are (temporarily) mounted in an existing MR scanner, the integral MR information in the PET data is beneficial, as no additional connections with the MR system are required.

The described method is used to extract the MR activity information shown in Figure 6.20 from the acquired data. Due to the characteristic MR frequency ($f_{\text{MR}}^{\text{(ADC)}} \approx 13.9$ MHz), the RF events can be discriminated from the normal detector pulse data and provide thus a time-tag for the beginning of an MR sequence, when the RF signal is emitted.

Chapter 7

Summary and Outlook

The simulations and measurements in the previous chapters show, that a sampling ADC approach can be successfully used for data acquisition in positron emission tomography. The early migration of the data processing tasks from the analog world to the digital domain provides more flexibility compared to e.g. an analog ASIC based solution. The use of field programmable logic devices allows a compact integration and maintains flexibility as the implemented firmware remains still reconfigurable after the installation. This readout concept is equally useful for detector designs in positron emission tomography as well as in high energy physics, which opens up a large potential for synergies between both fields.

The major part of this thesis addresses the requirements of a readout system for the MADPET-II small animal PET system. The current MADPET-II front-end specifications are therefore used as a starting point for the simulations in Chapter 4, to evaluate different algorithms for the feature extraction. Due to the rather slow signal pulse of the current MADPET-II preamplifiers, the determination of the signal amplitude from the sampled digital data is uncritical and the energy resolution is currently limited by the performance of the scintillator/APD detector front-end. A more challenging task is the determination of a precise timestamp for each detector signal, as it directly influences the quality of the PET image reconstruction. The digital implementation of a constant fraction discriminator with linear interpolation between sampling points provides a robust method to enhance the timing precision within the fixed ADC sampling interval. Slightly better results may be obtained with the more complex least squares error minimization methods, however this approach requires precise knowledge about the expected signal pulse shape and the achievable time resolution degrades significantly with mismatching pulse parameter assumptions. With the current MADPET-II frontend setup, this would require a careful parameter calibration/estimation for each detector channel, which remains still sensitive to e.g. temperature changes of the detector APDs.

The simulation results are confirmed by the measurements presented in Chapter 6. For a 128 channel prototype setup of the MADPET-II scanner, the achieved energy resolution of 14.8% FWHM is comparable to measurements obtained with the existing analog

ASIC based readout system for MADPET-II [Spa08, Tap08]. The time resolution of 4.8 ns FWHM is significantly improved compared to the ASIC based system, which gives about 11.7 ns FWHM for a similar measurement with 256 front-end channels [Spa08]. A further improvement in the time resolution may be achieved by optimizing the pulse parameters of the preamplifier signals, which can be achieved for example by a different preamplifier circuit.

Slightly better values than for MADPET-II are obtained with the updated MADPET-I lab course setup (9.5% FWHM energy resolution and 4.1 ns FWHM time resolution), which uses larger scintillator crystals and APDs, as well as a different type of preamplifier circuit [Sch09]. The USB based sampling ADC data acquisition system allows to build a compact PET scanner setup and provides an up-to-date readout system for the training of students. The matrix-based time calibration method introduced in Chapter 3, allows thereby a fast preparation of the acquired data for the PET image reconstruction.

First measurements with the emerging silicon photomultiplier (SiPM) sensor devices show also the applicability of a sampling ADC based data acquisition for this type of detectors. However, the full SiPM potential—with a time resolution in the range of some 100 ps [B⁺06, S⁺09]—could not be reached with the initial setup which is directly sampling the unamplified SiPM signal. With a sampling rate of 77.76 MHz, a time resolution of 2.4 ns FWHM can be reached using two coincident detectors with LYSO scintillator crystals and SiPM devices with 400 cells. Nevertheless, such a setup can be superior to an APD based design, as no additional preamplifier circuit is required for the SiPM signal. This provides a saving in space and power budget, which is beneficial for space constrained detector assemblies as required e.g. for combined multimodal PET/MRI systems. The robust trigger logic in the SADC firmware enables a continuous data acquisition in the noisy MRI environment, without external gating signals from the MRI system. In addition, it provides a precise tracking in time of the performed MRI protocol, which enables the subsequent correlation of both PET and MRI data, e.g. for motion correction.

Since the current ADC sampling rate is quite low compared to the fast SiPM signal rise time, the minimum achievable time resolution is limited. Besides the choice of a faster ADC type, the combination of the existing ADCs with a fast analog switched capacitor buffer ASIC—like the DRS4 chip [Rit08, DRS]—can increase the effective sampling rate up to several GHz and should thus allow to reach the sub-nanosecond time resolution region. Further advances in FPGA architecture nowadays also open the way for fast multichannel TDC implementations directly in the FPGA fabric [WZ08]. This concept may as well provide a cost efficient and high precision readout concept for an increased number of PET scanner channels.

The new design of the compact MSADC module addresses the limitations of the VME based SADC acquisition modules, which were used for the first measurements. The reduced size opens new applications with direct mounting options close to the detector frontend. Additionally, the module is targeted to the new ATCA crate standard, which is replacing the VME based systems in upcoming experiments in high energy physics. Thus, the MSADC can be the major building block in an updated sampling

ADC based acquisition system for the complete MADPET-II scanner or experiments like COMPASS [M⁺07] and PANDA [PAN05]. Until now, more than 190 MSADC modules are already successfully used in the upgraded readout electronics for the COMPASS ECAL2 electromagnetic calorimeter.

Appendix A

List of Abbreviations

AC	Alternating Current
ADD	Addition
ADC	Analog-to-Digital Converter
ALICE	A Large Ion Collider Experiment
ALTRO	Alice TPC Read Out
AMC	Advanced Mezzanine Card
APD	Avalanche Photodiode
ASIC	Application Specific Integrated Circuit
ATCA	Advanced Telecom Computing Architecture
BGO	Bismuth Germanate
CCI	Clock and Control Interface
CAMAC	Computer Automated Measurement and Control
CERN	Conseil Européen pour la Recherche Nucléaire
CFD	Constant Fraction Discriminator
CLB	Configurable Logic Block
CMP	Compare
COMPASS	Common Muon Proton Apparatus for Structure and Spectroscopy
CORDIC	Coordinate Rotation Digital Computer
CPLD	Complex Programmable Logic Device
CRLB	Cramér Rao Lower Bound
CT	X-ray Computed Tomography
DAC	Digital-to-Analog Converter
DAQ	Data Acquisition
DC	Direct Current
DDR	Double Data Rate
DESY	Deutsches Elektronen-Synchrotron
DIMM	Dual In-Line Memory Module

A LIST OF ABBREVIATIONS

DIV	Division
DOI	Depth of Interaction
EMI	Electromagnetic Interference
FDG	Fluorodeoxyglucose
FIFO	First-In First-Out
FIR	Finite Impulse Response
FOV	Field of View
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
FWHM	Full Width at Half Maximum
GSI	Gesellschaft für Schwerionenforschung
HDL	Hardware Description Language
IIR	Infinite Impulse Response
ILC	International Linear Collider
LOR	Line of Response
LSO	Lutetium Oxyorthosilicate
LUT	Lookup Table
LVDS	Low Voltage Differential Signal
LYSO	Lutetium-Yttrium Oxyorthosilicate
MADPET	Munich Avalanche (Photo)Diode Positron Emission Tomograph
MPPC	Multi-Pixel Photon Counter
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
MSADC	Mezzanine Sampling Analog-to-Digital Converter
MUL	Multiplication
ndf	Number of Degrees of Freedom
NIM	Nuclear Instrumentation Module
PANDA	Antiproton Annihilations at Darmstadt
PCB	Printed Circuit Board
PCI	Peripheral Component Interconnect
PCIe	PCI Express
PDF	Probability Density Function
PET	Positron Emission Tomography
PMT	Photomultiplier Tube
PTFE	Polytetrafluorethylene
RAM	Random Access Memory
REG	Register
RF	Radio Frequency
RMS	Root Mean Square

RRS	Recursive Running Sum
RTM	Rear Transition Module
SAD	Sum of Absolute Differences
SADC	Sampling Analog-to-Digital Converter
SDRAM	Synchronous Dynamic Random Access Memory
SiPM	Silicon Photomultiplier
SNR	Signal-to-Noise Ratio
SONET	Synchronous Optical Networking
SPECT	Single Photon Emission Computed Tomography
SVD	Singular Value Decomposition
TCS	Trigger Control System
TDC	Time-to-Digital Converter
TOF	Time of Flight
TPC	Time Projection Chamber
USB	Universal Serial Bus
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VME	Versa Module Eurocard
XFEL	X-ray Free-Electron Laser

Table A.1: List of abbreviations.

Appendix B

Mathematical Notation

\mathbf{A}	matrix
\mathbf{A}^T	transposed matrix
\mathbf{A}^+	pseudoinverse matrix
\mathbf{I}_i	identity matrix ($i \times i$)
$\mathbf{1}_{i,j}$	one matrix ($i \times j$)
\mathbf{x}	vector
\hat{x}	estimated value
\bar{x}	mean value
$\mathcal{F}(\bullet)$	fourier transform
$\mathcal{F}^{-1}(\bullet)$	inverse fourier transform
$\mathcal{L}(\bullet)$	laplace transform
$\mathcal{L}^{-1}(\bullet)$	inverse laplace transform
$\mathcal{Z}(\bullet)$	z-transform
$\mathcal{Z}^{-1}(\bullet)$	inverse z-transform
$E\{\bullet\}$	expectation operator
$f(\bullet)$	continuous function
$f[\bullet]$	discrete function
$\mathcal{N}(\mu, \sigma^2)$	normal distribution with mean μ and variance σ^2
$\mathcal{U}[a, b]$	uniform distribution over the interval $[a, b]$
j	imaginary unit
$\text{Re}(\bullet)$	real part
$\text{Im}(\bullet)$	imaginary part

Table B.1: List of mathematical symbols.

Appendix C

Transformations Overview

C.1 Fourier Transform

The Fourier transform is named after the French mathematician and physicist Jean Baptiste Joseph Fourier (1768–1830). The broad field of Fourier analysis provides mathematical techniques to decompose arbitrary signals into sums of sinusoidal functions. Thus the frequency components of a given time domain signal can be determined. For continuous time domain signals $x(t)$, the frequency domain Fourier transform $X(\omega)$ is given by

$$\begin{aligned} X(\omega) &= \mathcal{F}(x(t)) \\ (C.1) \quad &= \int_{-\infty}^{\infty} x(t) \cdot e^{-j\omega t} dt, \end{aligned}$$

with the angular frequency $\omega = 2\pi f$ and $e^{-j\omega t} = \cos(\omega t) - j \sin(\omega t)$. The inverse Fourier transform back into the time domain can then be stated as

$$\begin{aligned} x(t) &= \mathcal{F}^{-1}(X(\omega)) \\ (C.2) \quad &= \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) \cdot e^{j\omega t} d\omega. \end{aligned}$$

For a discrete signal representation $x[n]$, the corresponding discrete Fourier transform $X[k]$ is given by

$$\begin{aligned}
 (C.3) \quad X[k] &= \mathcal{F}(x[n]) \\
 &= \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi kn/N},
 \end{aligned}$$

and the inverse transform

$$\begin{aligned}
 (C.4) \quad x[n] &= \mathcal{F}^{-1}(X[k]) \\
 &= \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cdot e^{j2\pi kn/N}.
 \end{aligned}$$

C.2 Laplace Transform

The concept of transforming a given signal into its sinusoidal components by the Fourier transform can be extended to the decomposition into sinusoidal as well as exponential components. This results in the Laplace transform, named after the French mathematician Pierre Simon De Laplace (1749–1827). The Laplace transform converts a continuous time domain signal $x(t)$ into the so called s -domain representation $X(s)$

$$\begin{aligned}
 (C.5) \quad X(s) &= \mathcal{L}(x(t)) \\
 &= \int_0^{\infty} x(t) \cdot e^{-st} dt.
 \end{aligned}$$

The variable s describes a complex plane and can be decomposed into the real and complex part $s = \sigma + j\omega$. For the case of $\sigma = 0$, the Laplace transform therefore corresponds to the Fourier transform. The inverse Laplace transform is given by the complex integral

$$\begin{aligned}
 (C.6) \quad x(t) &= \mathcal{L}^{-1}(X(s)) \\
 &= \frac{1}{j2\pi} \int_{c-j\infty}^{c+j\infty} X(s) \cdot e^{st} ds.
 \end{aligned}$$

The constant c thereto has to be chosen accordingly, to have the integration path in the region of convergence of $X(s)$. This condition is fulfilled if $c > \text{Re}(s_{\infty})$ for all singularities s_{∞} of $X(s)$.

C.3 z-Transform

Similar to the Laplace transform, the z-transform decomposes the input signal into sinusoidal and exponential components. Whereas the Laplace transform requires a continuous input signal, the z-transform targets discrete signals. By the substitution $\sigma = \ln(r)$ and changing to discrete signals $x[n]$ the z-transform $X(z)$ can be derived from Equation (C.5)

$$(C.7) \quad \begin{aligned} X(z) &= \mathcal{Z}(x[n]) \\ &= \sum_{-\infty}^{\infty} x[n] \cdot z^{-n}, \end{aligned}$$

with $z = r \cdot e^{j\omega}$. Hence, the complex z-plane is arranged in polar coordinates, whereas the s-plane is arranged in a rectangular coordinate system. Therefore, vertical lines in the s-plane are mapped into concentric circles around the origin in the z-plane, with the s-plane imaginary axis ($\sigma = 0$) corresponding to the z-plane unit circle ($r = 1$). Respectively, the left and right part of the s-plane are mapped to the interior and exterior of the unit circle. It should also be noted that although the input signal $x[n]$ is discrete, the z-transform of the signal $X(z)$ is again a continuous function.

The frequency range in the z-plane runs from 0 to 2π , with $\omega = 0$ corresponding to DC level and $\omega = 2\pi$ corresponding to the sampling frequency f_s of the input signal. The requirement of the sampling theorem (given in equation (4.7)), that a discrete (or sampled) signal may only have frequency components between 0 and $f_s/2$, corresponds to the available frequency range on the z-plane unit circle. Thus, the allowable frequency range of 0 to $\pm f_s/2$ translates to both halves of the unit circle ranging from $\omega = 0$ to $\omega = \pm\pi$.

Additionally, the z-transform allows to extract directly a data flow representation of filter circuits from the z-domain description. Therefore, each z^{-1} expression corresponds to a unit delay of one sampling clock cycle. If (C.7) describes the transfer function of a digital filter, $x[n]$ corresponds to the filter coefficients, which specify the properties and frequency behavior of the filter.

List of Figures

2.1	Schematic view of positron decay, positron range in matter, annihilation and tomographic photon detection.	6
2.2	Detection of annihilation photons under different conditions.	9
2.3	Schematic view of an avalanche photodiode.	11
2.4	Schematic view of a silicon photomultiplier detector and illustrated realization on silicon.	12
2.5	Schematic view of the block detector concept.	14
2.6	Improvement of spatial resolution by using depth of interaction (DOI) information.	15
2.7	2D/3D data acquisition concepts.	16
2.8	Coincident event types and corresponding line of response (LOR).	17
3.1	MADPET-II dual layer detector module with integrated preamplifiers and close-up of the scintillator crystals of the first layer.	23
3.2	MADPET-II detector front-end.	23
3.3	Hamamatsu Photonics S8550 8×4 APD array and APD array with mounted scintillator crystals for the first detector layer.	24
3.4	APD high voltage supply circuit and preamplifier connection.	24
3.5	Energy histogram for a single MADPET-II detector channel.	27
3.6	Example of $2 + 2$ detectors in coincidence, resulting in 4 possible line-of-response (LOR) combinations.	29
3.7	Illustration of the time-misalignment influence on the time resolution.	29
3.8	Measurement setup for the time calibration with a centered timing probe.	31
3.9	Ideal measurement setup for the iterative time calibration.	31
3.10	Comparison of the LOR time difference histograms for the segmented method and the fan-sum method.	42
3.11	Time difference histograms of the full MADPET-II system before and after time calibration with 10 iterations of the fan-sum method.	43
3.12	Improvement of different time-calibration methods after multiple iterations of the calibration algorithm.	43

LIST OF FIGURES

4.1	Schematic view of the time domain detector pulse shape model $h(t)$	46
4.2	Normalized pulse shapes of $h_{\text{LSO}}(t)$ and $h_{\text{APD}}(t)$	49
4.3	Convolved pulse shape $h(t) = h_{\text{LSO}}(t) * h_{\text{APD}}(t)$	49
4.4	Magnified rising edge of $h(t)$ for 10%–90% rise/fall parameterization and τ parameterization.	49
4.5	Normalized and time aligned detector pulses, acquired with a sampling ADC.	49
4.6	Normalized frequency response $H(\omega) = \mathcal{F}(h(t)) = H(s) _{s=j\omega}$	51
4.7	Normalized frequency response $H[\omega]$ of $h(t)$ when sampled at 80 MHz compared to the spectrum $H(\omega)$ of the continuous signal.	51
4.8	Implementation alternatives for the average filter.	54
4.9	Frequency response of the average filter for different window sizes N_{avg}	55
4.10	Optimized trigger average filters with eliminated $z^{-N_{\text{avg}}}$ delay chain and fully pipelined implementation.	56
4.11	Frequency response of the optimized trigger average filters compared to the accurate RRS average filter.	57
4.12	IIR filter structure to generate the sampled pulse shape of $H(z)$	60
4.13	Peak-hold circuit implementation.	61
4.14	Underestimation of the maximum value when derived from sampled data.	62
4.15	Relative maximum error versus sampling frequency for the peak-hold method.	62
4.16	Parabola interpolation $p(t)$ of the continuous pulse $h(t)$, based on sampled data $h[n]$	63
4.17	Relative maximum error versus sampling frequency for the parabola method and peak-hold.	63
4.18	Data flow for the parabola based maximum interpolation.	64
4.19	Upsampling data flow.	65
4.20	Spectrum of the original signal $X[f]$ and the expanded intermediate signal $X_2[f]$	67
4.21	Frequency characteristics of different upsampling filters, compared to the spectrum of $x_2[n']$	67
4.22	Upsampling of the signal $h[n]$ by $M = 16$, with varying interpolation filter length L	67
4.23	Relative maximum error versus sampling frequency for the upsampling method, the parabola method and peak-hold.	67
4.24	Data flow for the cubic spline based interpolation.	72
4.25	Cubic spline interpolation of $h[n]$	73
4.26	Relative maximum error versus sampling frequency for the cubic spline interpolation method.	73

4.27	Comparison of the relative maximum error and the standard deviation of the obtained maximum value for noisy data samples at different SNR values for peak-hold, parabola interpolation and upsampling.	77
4.28	Comparison of the relative maximum error and the standard deviation of the obtained maximum value for noisy data samples at different SNR values for upsampling and cubic spline interpolation.	78
4.29	Smoothing cubic spline interpolation λ parameter scan and comparison of the relative maximum error and the standard deviation of the obtained maximum value for noisy data samples at different SNR values.	79
4.30	Leading edge amplitude walk.	84
4.31	Constant fraction timing.	84
4.32	Principle of the digital constant fraction discrimination algorithm with linear interpolation.	85
4.33	Schematic view of the CFD algorithm data flow.	86
4.34	Simulation of the digital CFD algorithm with linear interpolation at varying sampling rates.	88
4.35	Effect of the interpolation with baseline values due to undersampling.	88
4.36	Rising edge linear interpolation mismatch due to undersampling.	88
4.37	Simulation of the digital CFD algorithm with linear interpolation for varying fraction settings.	89
4.38	Calculation of the Fisher information for individual samples of $h[n; \phi]$ and mapping to the optimal CFD fraction.	91
4.39	Calculation of the Cramér-Rao lower bound for the CFD algorithm at different sampling rates.	92
4.40	Comparison of the CFD algorithm performance, simulated on sampled data and spline interpolated data.	93
4.41	Smoothing parameter (λ) scans for the CFD algorithm with smoothing spline interpolated data at different SNR values.	94
4.42	Comparison of the the CFD algorithm performance, simulated on smoothing cubic spline interpolated data.	94
4.43	Schematic view of the least squares grid search method.	97
4.44	Simulation of the grid search method at varying sampling rates.	98
4.45	Simulation of the sensitivity to model parameter variations for the grid search method.	98
4.46	Parameter scan simulations for the number of samples N_s and the number of pulse model realizations N_ϕ	99
4.47	Simulation of the analytic least squares method at varying sampling rates.	102
4.48	Simulation of the sensitivity to model parameter variations for the analytic least squares method.	102
4.49	Schematic view of the analytic least squares method.	103

LIST OF FIGURES

4.50	Calculation of the sampled Fisher information $f[n]$ to determine the optimal CFD threshold setting.	106
4.51	Comparison of the CFD algorithm performance for the optimized pulse shape.	106
4.52	Comparison of the relative maximum error and the standard deviation of the obtained maximum value.	107
4.53	Comparison of the standard deviation for the extracted pulse maximum with parabola interpolation and smoothing cubic spline interpolation. . .	108
5.1	Sampling ADC VME module.	110
5.2	Overview of the sampling ADC data flow.	111
5.3	Multiplexer VME module.	112
5.4	Example configuration of an SADC readout network.	113
5.5	PCI bus interface and buffer module with unmounted S-Link card.	115
5.6	Simplified view of an FPGA fabric.	116
5.7	Overview of the firmware data flow in the processing FPGA on the SADC.	119
5.8	Overview of the data flow in the trigger block.	120
5.9	Overview of the trigger logic.	120
5.10	Overview of the average window and buffer relations, defining the starting point of the event data.	121
5.11	Overview of the data flow in the event buffer block.	122
5.12	Overview of the data flow in the channel multiplexer with time-order scheduling.	123
5.13	Overview of the data flow in the processing blocks.	124
5.14	Processing chain for the MADPET-II application.	124
5.15	Mezzanine sampling ADC module.	129
5.16	Overview of the data flow on the mezzanine sampling ADC module.	131
5.17	USB interface for the MSADC module.	132
6.1	Measurement setup with 2×2 opposing MADPET-II front layer detector modules.	134
6.2	Energy resolution of the 128 MADPET-II channels acquired.	135
6.3	Position of the 511 keV photopeak in ADC values for each channel.	135
6.4	Typical distribution of the coarse-time t_c values.	137
6.5	Typical distribution of the fine-time t_f values.	137
6.6	Distribution of the fine-time t_f values for different constant-fraction thresholds.	138
6.7	Rebinned distribution of the fine-time t_f values for different constant-fraction thresholds.	138

6.8	Time resolution for different constant fraction settings, before and after time calibration.	139
6.9	Fit quality for the determination of the time resolution from the global time difference histogram.	139
6.10	Signal to noise ratio for each channel of the MADPET-II setup.	140
6.11	Detector setup with the 3×3 opposing MADPET-I detector modules. . . .	141
6.12	MADPET-I detector front-end with radioactive source placed on a rotating table in the field of view.	142
6.13	Electronic system for the MADPET-I lab course.	142
6.14	Normalized and time aligned MADPET-I detector pulses, acquired with the SADC system.	143
6.15	Signal to noise ratio for the MADPET-I detector front-end.	143
6.16	Sample source for the lab course with the TUM logo as active region. . . .	144
6.17	Reconstructed images of the TUM source, using simple backprojection and filtered backprojection.	144
6.18	Measurement setup for the SiPM/MPPC evaluation.	145
6.19	Normalized and time aligned signal pulses from 400 cell and 1600 cell MPPCs.	145
6.20	Mean time between consecutive events acquired and concurrent MR RF activity.	150
6.21	Attenuation of the average filter in the trigger decision and position of the real and aliased MR frequency.	150

List of Tables

2.1	Tracers for PET and their applications.	7
2.2	Common scintillator materials.	10
3.1	Component count for the existing MADPET-II data acquisition system. . .	26
3.2	Simulation of the time calibration methods for different LOR geometries. .	40
3.3	Properties of the time calibration: Computational complexity, memory re- quirements and computing time.	41
3.4	Number of individual LORs for the three time calibration methods.	41
4.1	Standard deviation of the maximum estimation for selected SNR values. .	81
4.2	Hardware requirements for the maximum extraction algorithms.	82
5.1	Format of the SADC output packet for 10 bit ADCs.	125
5.2	Data word format for 12 bit ADCs.	125
5.3	Required clock cycles for each stage of the SADC firmware processing pipeline and resulting time delays.	125
6.1	Mean energy resolution of the MADPET-II setup for different amplitude extraction/interpolation algorithms.	135
6.2	Minimum time resolution of the MADPET-II setup for the CFD based time extraction with different interpolation options.	140
6.3	MADPET-I mean energy resolution for different amplitude extrac- tion/interpolation algorithms.	143
6.4	MADPET-I minimum time resolution for the CFD based time extraction with different interpolation options.	143
6.5	Hamamatsu 400 cell MPPC mean energy resolution for different amplitude extraction/interpolation algorithms.	147
6.6	Hamamatsu 400 cell MPPC minimum time resolution for different inter- polation algorithms.	147
A.1	List of abbreviations.	157
		171

LIST OF TABLES

B.1 List of mathematical symbols. 159

Bibliography

- [ALT03] R. Esteve Bosch, A. Jiménez de Parga, B. Mota, and L. Musa. *The ALTRO Chip: A 16-Channel A/D Converter and Digital Processor for Gas Detectors*. IEEE Trans. Nucl. Sci. **50** 6, 2460–2469 (2003).
- [AS308] Altera Corp. *Stratix III Device Handbook*, 2008. v1.8.
- [AS408] Altera Corp. *Stratix IV Device Handbook*, 2008. v2.1.
- [Ash96] P.C. Ashenden. *The Designer's Guide to VHDL*. Morgan Kaufmann Publishers, Inc., 8th edition, 1996.
- [ATI01] ATIS Committee T1A1. *ATIS Telecom Glossary 2000*. American National Standards Institute, Inc. (T1.523-2001).
- [B⁺95] D. Bidello et al. *Silicon avalanche detectors with negative feedback as detectors for high energy physics*. Nucl. Instr. and Meth. **A 367**, 212–214 (1995).
- [B⁺03] P. Buzhan et al. *Silicon photomultiplier and its possible applications*. Nucl. Instr. and Meth. **A 504**, 48–52 (2003).
- [B⁺05] J.E. Breeding et al. *Baseline Correction in PET Utilizing Continuous Sampling ADCs to Compensate for DC and Count Rate Errors*. United States Patent **US 6903344 B2** (2005).
- [B⁺06] P. Buzhan et al. *Timing by silicon photomultiplier: A possible application for TOF measurements*. Nucl. Instr. and Meth. **A 567**, 353–355 (2006).
- [BR97] R. Brun and F. Rademakers. *ROOT—An object oriented data analysis framework*. Nucl. Instr. and Meth. **A 389**, 81–86 (1997).
- [CGJ⁺02] M. Caffrey, P. Graham, E. Johnson, M. Wirthlin, and C. Carmichael. *Single-Event Upsets in SRAM FPGAs*. Proceedings of Military and Aerospace Applications of Programmable Logic Devices, 1–6 (2002).
- [Che06] S.R. Cherry. *Multimodality In Vivo Imaging Systems: Twice the Power or Double the Trouble?* Annu. Rev. Biomed. Eng. **8**, 35–62 (2006).
- [CSP03] S.R. Cherry, J.A. Sorenson, and M.E. Phelps. *Physics in Nuclear Medicine*. Saunders, 3rd edition, 2003.

BIBLIOGRAPHY

- [CY05] Cypress Semiconductor Corp. *HOTLink Transmitter/Receiver Datasheet*, 2005.
- [Dav04] A.M. Davies. *Analyse linearer Schaltungen*. Hüthig, 1st edition, 2004.
- [dB78] C. de Bor. *A practical guide to splines*. Springer-Verlag, 1st edition, 1978.
- [DRS] *DRS Chip Home Page*, <http://drs.web.psi.ch>.
- [F⁺06] J.P. Finn et al. *Cardiac MR Imaging: State of the Technology*. *Radiology* **241** 2, 338–354 (2006).
- [F⁺08] R. Fontaine et al. *Timing Improvement by Low-Pass Filtering and Linear Interpolation for the LabPET Scanner*. *IEEE Trans. Nucl. Sci.* **55** 1, 34–39 (2008).
- [F⁺09] R. Fontaine et al. *The Hardware and Signal Processing Architecture of LabPETTM, a Small Animal APD-Based Digital PET Scanner*. *IEEE Trans. Nucl. Sci.* **56** 1, 3–9 (2009).
- [Für09] S. Fürst. *Development and Evaluation of novel Detectors for Combined PET/MR Imaging, Based on SiPMs and Fast Scintillation Crystals*. Diploma thesis, Technische Universität München, 2009.
- [G⁺04] V. Golovin et al. *Novel type of avalanche photodetector with Geiger mode operation*. *Nucl. Instr. and Meth. A* **518**, 560–564 (2004).
- [G⁺06a] S. Gomi et al. *Development of Multi-Pixel Photon Counters*. *IEEE NSS Conf. Rec.* **2**, 1105–1111 (2006).
- [G⁺06b] P. Guerra et al. *New Embedded Digital Front-End for High Resolution PET Scanner*. *IEEE Trans. Nucl. Sci.* **53** 3, 770–775 (2006).
- [GK65] G. Golub and W. Kahan. *Calculating the Singular Values and Pseudo-Inverse of a Matrix*. *J. SIAM Numer. Anal. Ser. B* **2** 2, 205–224 (1965).
- [Gom98] C. Gomez. *Engineering and Scientific Computing with Scilab*. Birkhäuser, 1998.
- [GR70] G.H. Golub and C. Reinsch. *Singular Value Decomposition and Least Squares Solutions*. *Numer. Math.* **14** 5, 403–420 (1970).
- [Gru01] B. Grube. *The Trigger Control System and the Common GEM and Silicon Read-out for the COMPASS Experiment*. Diploma thesis, Technische Universität München, 2001.
- [Gru06] B. Grube. *A Trigger Control System for COMPASS and a Measurement of the Transverse Polarization of Λ and Ξ Hyperons from Quasi-Real Photo-Production*. Ph.D. thesis, Technische Universität München, 2006.

- [Hof75] R. Hofstadter. *Twenty Five Years of Scintillation Counting*. IEEE Trans. Nucl. Sci. **22** 1, 13–25 (1975).
- [HP04] Solid State Division Hamamatsu Photonics, K.K. *Characteristics and use of Si APD (Avalanche Photodiode)*, 2004. Technical Information SD-28.
- [HP06] Solid State Division Hamamatsu Photonics, K.K. *Si APD array S8550*, 2006. Datasheet.
- [HP09a] Solid State Division Hamamatsu Photonics, K.K. *Multi-Pixel Photon Counter*, 2009. Technical Information.
- [HP09b] Solid State Division Hamamatsu Photonics, K.K. *Multi-Pixel Photon Counter*, 2009. S10362-11 series.
- [ILC07] ILC Collaboration. *International Linear Collider Reference Design Report*. (August, 2007). <http://www.linearcollider.org>.
- [IV508] Tokyo Electron Device Ltd. *Virtex-5 LXT/SXT PCI Express Evaluation Platform Board*, 2008. TB-5V-LX50/110T/SX50T/FX70T-PCIEXP, User's Manual, Rev. 2.01.
- [JR75] F. James and M. Roos. *Minuit—A System for Function Minimization and Analysis of the Parameter Errors and Correlations*. Comput. Phys. Comm. **10**, 343–367 (1975).
- [JS93] G. Jacovitti and G. Scarano. *Discrete Time Techniques for Time Delay Estimation*. IEEE Trans. Sig. Proc. **41** 2, 525–533 (1993).
- [K⁺07] M. Korzhik et al. *Development of scintillation materials for PET scanners*. Nucl. Instr. and Meth. A **571**, 122–125 (2007).
- [Kay93] S.M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 10th edition, 1993.
- [Kno99] G.F. Knoll. *Radiation detection and measurement*. John Wiley & Sons, 3rd edition, 1999.
- [L⁺05] J.-D. Leroux et al. *Time Determination of BGO-APD Detectors by Digital Signal Processing for Positron Emission Tomography*. IEEE NSS Conf. Rec. **3**, 1723–1727 (2005).
- [L⁺09] J.-D. Leroux et al. *Time Discrimination Techniques Using Artificial Neural Networks for Positron Emission Tomography*. IEEE Trans. Nucl. Sci. **56** 3, 588–595 (2009).
- [LEC09] Lattice Semiconductor Corp. *LatticeECP2/M Family Data Sheet*, 2009. v03.4.

BIBLIOGRAPHY

- [LGMY01] M.W. Lenox, T. Gremillion, S. Miller, and J.W. Young. *Coincidence Time Alignment for Planar Pixellated Positron Emission Tomography Detector Arrays*. IEEE NSS Conf. Rec. **4**, 1952–1954 (2001).
- [LSC08] Lattice Semiconductor Corp. *LatticeSC/M Family Data Sheet*, 2008. v02.2.
- [Lud95] T. Ludziejewski et al. *Advantages and Limitations of LSO Scintillator in Nuclear Physics Experiments*. IEEE Trans. Nucl. Sci. **42 4**, 328–336 (1995).
- [Lyo04] R.G. Lyons. *Understanding Digital Signal Processing*. Prentice Hall, 2nd edition, 2004.
- [M⁺03] J.C. Moyers et al. *Continuous Sampling and Digital Integration for PET Scintillation*. United States Patent **US 6664543 B2** (2003).
- [M⁺07] G.K. Mallot et al. *The COMPASS experiment at CERN*. Nucl. Instr. and Meth. **A 577**, 455–518 (2007).
- [M⁺09] A.B. Mann et al. *The Universal Sampling ADC Readout System of the COMPASS Experiment*. IEEE NSS Conf. Rec., 2225–2228 (2009).
- [Man04] A.B. Mann. *Data Acquisition and Fast Signal Processing for a new Positron Emission Tomograph*. Diploma thesis, Technische Universität München, 2004.
- [MB04] U. Meyer-Baese. *Digital Signal Processing with Field Programmable Gate Arrays*. Springer, 2nd edition, 2004.
- [McE07] D.P. McElroy, T.J. Thompson, V.Ch. Spanoudaki, and S.I. Ziegler. *Use of a Central Positron Emitting Reference Source to Improve the Timing Alignment of a Singles List-Mode Small Animal PET Scanner*. IEEE Trans. Nucl. Sci. **54 1**, 50–54 (2007).
- [Mei05] A. Meister. *Numerik linearer Gleichungssysteme*. Vieweg, 2nd edition, 2005.
- [MGK⁺06] A. Mann, B. Grube, I. Konorov, S. Paul, L. Schmitt, D.P. McElroy, and S.I. Ziegler. *A Sampling ADC Data Acquisition System for Positron Emission Tomography*. IEEE Trans. Nucl. Sci. **53 1**, 297–303 (2006).
- [MIN] *MINUIT Documentation, CERN Program Library*, <http://wwwasdoc.web.cern.ch/wwwasdoc/minuit/minmain.html>.
- [MKP07a] A.B. Mann, I. Konorov, and S. Paul. *A Versatile Sampling ADC System for On-Detector Applications and the AdvancedTCA Crate Standard*. 15th IEEE-NPSS Real-Time Conference, 1–5 (2007).
- [MKP⁺07b] A.B. Mann, I. Konorov, S. Paul, V.C. Spanoudaki, and S.I. Ziegler. *A Flexible AdvancedTCA Based Sampling ADC System for Multimodality Positron Emission Tomography*. IEEE NSS Conf. Rec. **3**, 1729–1732 (2007).

- [Mod91] R. Moddemeijer. *On the Determination of the Position of Extrema of Sampled Correlators*. IEEE Trans. Sig. Proc. **39** **1**, 216–219 (1991).
- [Moo20] E.H. Moore. *On the reciprocal of the general algebraic matrix*. Bull. American Math. Society **26** **9**, 394–395 (1920).
- [MPT⁺09] A.B. Mann, S. Paul, A. Tapfer, V.C. Spanoudaki, and S.I. Ziegler. *A Computing Efficient PET Time Calibration Method Based on Pseudoinverse Matrices*. IEEE NSS Conf. Rec., 3889–3892 (2009).
- [MS92] C.L. Melcher and J.S. Schweitzer. *Cerium-doped Lutetium Oxyorthosilicate: A Fast, Efficient New Scintillator*. IEEE Trans. Nucl. Sci. **39** **4**, 502–505 (1992).
- [OS99] A.V. Oppenheim and R.W. Schafer. *Zeitdiskrete Signalverarbeitung*. R. Oldenbourg, 3rd edition, 1999.
- [P⁺08] S.-J. Park et al. *Digital Coincidence Processing for the RatCAP Conscious Rat Brain PET Scanner*. IEEE Trans. Nucl. Sci. **55** **1**, 510–515 (2008).
- [PAN05] PANDA Collaboration. *Technical Progress Report for PANDA*. (February, 2005). <http://www-panda.gsi.de>.
- [Pen55] R. Penrose. *A Generalized Inverse for Matrices*. Proc. Cambridge Phil. Society **51**, 406–413 (1955).
- [Pet07] G. Petzold. *Measurement of the electron-antineutrino angular correlation coefficient a in neutron beta decay with the spectrometer aSPECT*. Ph.D. thesis, Technische Universität München, 2007.
- [Phe06] M.E. Phelps. *PET — Physics, Instrumentation, and Scanners*. Springer, 1st edition, 2006.
- [Pic01a] B.J. Pichler et al. *A 4×8 APD Array, Consisting of Two Monolithic Silicon Wafers, Coupled to a 32-Channel LSO Matrix for High-Resolution PET*. IEEE Trans. Nucl. Sci. **48** **4**, 1391–1396 (2001).
- [Pic01b] B.J. Pichler et al. *Integrated Low-Noise Low-Power Fast Charge-Sensitive Preamplifier for Avalanche Photodiodes in JFET-CMOS Technology*. IEEE Trans. Nucl. Sci. **48** **6**, 2370–2374 (2001).
- [PIC06a] PCI Industrial Computer Manufacturers Group. *AdvancedTCA Base Specification, PICMG 3.0 Revision 2.0*, May 26, 2006.
- [PIC06b] PCI Industrial Computer Manufacturers Group. *AdvancedMC Short Form Specification, PICMG AMC.0 Revision 2.0*, December 28, 2006.
- [PIC06c] PCI Industrial Computer Manufacturers Group. *MicroTCA Short Form Specification, PICMG MTCA.0*, September 21, 2006.

BIBLIOGRAPHY

- [Pic08] R. Picker. *PENeLOPE and AbEx - On the Way Towards a New Precise Neutron Lifetime Measurement*. Ph.D. thesis, Technische Universität München, 2008.
- [Raf01] M. Rafecas, G. Böning, B.J. Pichler, E. Lorenz, M. Schwaiger, and S.I. Ziegler. *A Monte Carlo Study of High-Resolution PET With Granulated Dual-Layer Detectors*. IEEE Trans. Nucl. Sci. **48** 4, 1490–1495 (2001).
- [Raf04] M. Rafecas et al. *Use of a Monte Carlo-Based Probability Matrix for 3-D Iterative Reconstruction of MADPET-II Data*. IEEE Trans. Nucl. Sci. **51** 5, 2597–2605 (2004).
- [Rit08] S. Ritt. *Design and performance of the 6 GHz waveform digitizing chip DRS4*. IEEE NSS Conf. Rec., 1512–1515 (2008).
- [ROO] *The ROOT System Home Page*, <http://root.cern.ch>.
- [RW97] L. Rade and B. Westergren. *Springers Mathematische Formeln*. Springer, 2nd edition, 1997.
- [S⁺02] M. Streun et al. *Coincidence detection by digital processing of free-running sampled pulses*. Nucl. Instr. and Meth. **A** **487**, 530–534 (2002).
- [S⁺04] L. Schmitt et al. *The DAQ of the COMPASS Experiment*. IEEE Trans. Nucl. Sci. **51** 3, 439–444 (2004).
- [S⁺06] M. Streun et al. *The Data Acquisition System of ClearPET Neuro - a Small Animal PET Scanner*. IEEE Trans. Nucl. Sci. **53** 3, 700–703 (2006).
- [S⁺09] D.R. Schaart et al. *A novel, SiPM-array-based, monolithic scintillator detector for PET*. Phys. Med. Biol. **54**, 3501–3512 (2009).
- [Sch08] H.P. Schlemmer. *Simultaneous MR/PET for brain imaging: Feasibility Study*. Radiology **248** 3, 1028–1035 (2008).
- [Sch09] F. Schneider. *Setup and Commissioning of a Positron Emission Tomograph for a Student's Advanced Laboratory Course*. Master thesis, Technische Universität München, 2009.
- [Smi03] S.W. Smith. *Digital Signal Processing*. Newnes, 2003.
- [SMS07] Siemens Medical Solutions USA Inc. *Biograph TruePoint PET/CT System specifications*. (2007).
- [Spa05a] V.Ch. Spanoudaki, D.P. McElroy, K. Zell, and S.I. Ziegler. *A Software Based Iterative Method of Improving the System Wide Timing Resolution of the MADPET-II Small Animal PET Tomograph*. Biomedizinische Technik **50**, 897–898 (2005).

- [Spa05b] V.Ch. Spanoudaki, D.P. McElroy, K. Zell, and S.I. Ziegler. *Effect of Temperature on the Stability and Performance of an LSO-APD PET Scanner*. IEEE Nuclear Science Symposium **5**, 3014–3017 (2005).
- [Spa06] V.Ch. Spanoudaki, D.P. McElroy, and S.I. Ziegler. *An analog signal processing ASIC for a small animal LSO-APD PET tomograph*. Nucl. Instr. and Meth. **A 564**, 451–462 (2006).
- [Spa08] V.C. Spanoudaki. *Development and Performance Studies of a Small Animal Positron Emission Tomograph with Individual Crystal Readout and Depth of Interaction Information and Studies of Novel Detector Technologies in Medical Imaging*. Ph.D. thesis, Technische Universität München, 2008.
- [Tap08] A. Tapfer. *Untersuchungen der Energie- und Zeitauflösung des APD-basierten Kleintier Positronen-Emissions-Tomographen MADPET-II*. Diploma thesis, Universität Karlsruhe, 2008.
- [TI09] Texas Instruments Inc. *ADS5270 Datasheet*, 2009.
- [Tur60] G.L. Turin. *An Introduction to Matched Filters*. IRE Transactions on information Theory **6 3**, 311–329 (1960).
- [UAE91a] M. Unser, A. Aldroubi, and M. Eden. *Fast B-Spline Transforms for Continuous Image Representation and Interpolation*. IEEE Trans. Patt. Anal. Machine Intell. **13 3**, 277–285 (1991).
- [UAE91b] M. Unser, A. Aldroubi, and M. Eden. *Recursive Regularization Filters: Design, Properties, and Applications*. IEEE Trans. Patt. Anal. Machine Intell. **13 3**, 272–277 (1991).
- [UAE93a] M. Unser, A. Aldroubi, and M. Eden. *B-Spline Signal Processing: Part I—Theory*. IEEE Trans. Signal Processing **41 2**, 821–833 (1993).
- [UAE93b] M. Unser, A. Aldroubi, and M. Eden. *B-Spline Signal Processing: Part II—Efficient Design and Applications*. IEEE Trans. Signal Processing **41 2**, 834–848 (1993).
- [Uns99] M. Unser. *Splines—A Perfect Fit for Signal and Image Processing*. IEEE Signal Processing Magazine **16 6**, 22–38 (1999).
- [vdBMBR97] H.C. van der Bij, R.A. McLaren, O. Boyle, and G. Rubin. *S-LINK, a data link interface specification for the LHC era*. IEEE Trans. Nucl. Sci. **44 3**, 398–402 (1997).
- [vSS09] G.K. von Schulthess and H.-P.W. Schlemmer. *A look ahead: PET/MR versus PET/CT*. Eur. J. Nucl. Med. **36 Supplement 1**, S3–S9 (2009).
- [WA04] M.N. Wernick and J.N. Aarsvold. *Emission Tomography*. Elsevier Academic Press, 1st edition, 2004.

BIBLIOGRAPHY

- [WZ08] J. Wu and S. Zonghan. *The 10-ps Wave Union TDC: Improving FPGA TDC Resolution beyond Its Cell Delay*. IEEE NSS Conf. Rec., 3440–3446 (2008).
- [XFE07] XFEL Project Group. *The European X-Ray Free-Electron Laser Technical Design Report*. (July, 2007). <http://xfel.desy.de/>.
- [Xil04] Xilinx Inc. *Xilinx LogiCore, CORDIC v3.0 product specification*, 2004. Datasheet DS249.
- [Xil08] Xilinx Inc. *Xilinx-4 FPGA Configuration User Guide*, 2008. User Guide UG071, v1.10.
- [XV207] Xilinx Inc. *Virtex-II Platform FPGA User Guide*, 2007. User Guide UG002, v2.2.
- [XV408] Xilinx Inc. *Virtex-4 FPGA User Guide*, 2008. User Guide UG070, v2.5.
- [XV509] Xilinx Inc. *Virtex-5 FPGA User Guide*, 2009. User Guide UG190, v5.0.
- [Y+06] K. Yamamoto et al. *Development of Multi-Pixel Photon Counter (MPPC)*. IEEE NSS Conf. Rec. **2**, 1094–1097 (2006).
- [Y+07] K. Yamamoto et al. *Development of Multi-Pixel Photon Counter (MPPC)*. IEEE NSS Conf. Rec. **2**, 1511–1515 (2007).
- [Zie01] S. Ziegler et al. *A prototype high resolution animal positron emission tomograph with avalanche photodiode arrays and LSO crystals*. Eur. J. Nucl. Med. **28 2**, 136–143 (2001).

Acknowledgments

Foremost, I would like to thank Prof. Stephan Paul for the opportunity to work on this interesting topic (and many other ones) at his chair and for his continuous support of the interdisciplinary collaboration with the medical and electric engineering departments. This work, and the involvement in the high energy physics business at CERN and GSI—with its challenging electronics requirements—really turned out to be an engineer’s paradise.

I am also very grateful to Prof. Klaus Diepold for his enduring support and for overseeing this thesis in the electric engineering and information technology department.

Many thanks also go to Prof. Sibylle Ziegler for accepting me in the MADPET group at the university hospital rechts der Isar, and for the fruitful discussions regarding current and future challenges in PET.

I am very indebted to Igor Konorov for many productive discussions regarding all kinds of electronic challenges, current and future projects, and for his constant support during many setup and debugging sessions. Many thanks also go to Heinz Angerer, for always having the right tool or a piece of PCB at hand.

It was a great pleasure to work with the MADPET group and the colleagues at rechts der Isar. Especially, I would like to thank Virginia Spanoudaki, who completed and characterized the MADPET-II hardware setup, and Irene Torres-Espallardo. Together with them, I learned a lot about PET and life during the last years. I appreciate also the work and discussions with the other (current and former) team members: Ralph Bundschuh, Gaspar Delso, David McElroy, Sebastian Fürst, Melanie Hohberg, Jozef Pulko, Jasmine Schirmer, Arne Tapfer and Astrid Velroyen.

I also enjoyed very much the working environment at E18. Many thanks go to my long term room mates Anna-Maria Dinkelbach, Steffi Grabmüller and Boris Grube. Boris provided me as well with a template style for this thesis and many VHDL building blocks for the SADC firmware. I am also thankful to the other members of the COMPASS group for support and helping hands, both at CERN and in Garching.

Florian Schneider worked for his Master thesis on the challenging task of upgrading the MADPET-I scanner with the SADC system. He asked the right questions to bring up the remaining firmware bugs and finally obtained the first reconstructed images. Special thanks also go to Karin Frank, for support in any administrative question, and all other members of E18 for keeping the rich environment with “open doors” running.

ACKNOWLEDGMENTS

For their valuable comments and corrections to this thesis I am also very thankful to Bea Franke, Christian Hesse and Thomas Rosner. I very much appreciate the time they (and many other colleagues) spent reading through all these pages.

Abschließender Dank gebührt meinen Eltern. Sie haben mich während meines Studiums und dieser Arbeit jederzeit unterstützt und alle meine Entscheidungen mitgetragen. Ohne die immer währende Förderung meines frühen Interesses an allen technischen Dingen wäre diese Arbeit sicher so nicht möglich gewesen.