Published by IOP Publishing for Sissa Medialab

Received: *February 6, 2024*
Accepted: *November 1, 2024*
Published: *???, 2025*

PET Electronics & Technologies 2023
Duisburg, Germany
12–13 September 2023

Design space exploration for particle detector read-out implementations in Matlab and Simulink on the example of the SHiP SBT

F. Rössing,[a,*] **D. Arutinov,**[a] **A. Brignoli,**[b] **H. Fischer,**[c] **C. Grewing,**[a] **H. Lacker,**[b] **F. Lyons,**[c] **A. Zambanini**[a] **and S. van Waasen**[a,d]

[a] *Central Institute of Engineering, Electronics and Analytics – Electronic Systems (ZEA-2), Forschungszentrum Jülich GmbH, 52428 Jülich, Germany*
[b] *Humboldt-Universität zu Berlin, Institut für Physik, 12489 Berlin, Germany*
[c] *Albert-Ludwigs-Universität Freiburg, Physikalisches Institut, 79104 Freiburg, Germany*
[d] *Faculty of Engineering, Communication Systems, University of Duisburg-Essen, 47057 Duisburg, Germany*

E-mail: f.roessing@fz-juelich.de

Abstract: On a very fundamental level, particle detectors share similar requirements for their read-out chain. This is reflected in the way that typical read-out solutions are developed, where a previous design is modified to fit changes in requirements. One of the two common approaches is the current-based read-out, where the waveform of the sensor output is sampled in order to later extract information in the central processing unit. This approach is used in many detector applications using scintillation based detectors, including PET.

With this contribution, we will introduce how we use Matlab in order to simulate the read-out electronics of particle detectors. We developed this simulation approach as a base for our ongoing development of software-defined read-out Application Specific Integrated Circuits (ASICs) that cover the requirements of a variety of particle detector types. Simulink was chosen as a base for our developments as it allows simulation of mixed-signal systems and comes with built-in toolkits to aid in developments of such systems.

With our approach, we want to take a new look at how we approach designing such a read-out, with a focus on digital signal processing closer to the sensor in order or reduce transmission bandwidth,

[*] Corresponding author.

https://doi.org/xxxxxxx

making use of known signal characteristics and modern methods of communications engineering. We are taking into account the time profile of an event, the bandwidth-limiting properties of the sensor and attached electronics, digitization stages and finally the parameterization of approaches for digital processing of the signal.

We will show how we are applying the design approach to the development of a read-out for the proposed SHiP SBT detector, which is a scintillation based detector relying on SiPMs sensors, using this as an example for our modelling approach and show preliminary results.

Given the similarity in principle, the modelling approach could easily be modified for PET systems, allowing studies of the read-out chain to possibly gain in overall performance.

Keywords: Data acquisition concepts; Digital signal processing (DSP); Simulation methods and programs; Front-end electronics for detector readout

ArXiv ePrint: 2402.03122

Contents

1 Motivation

The fundamental characteristics of particle detectors are largely similar. They generate a current pulse, which is the result of the detector‘s response to the incident energy, and is influenced by the electrical characteristics of the sensor and read-out electronics. To achieve optimal performance, it is essential to have a comprehensive understanding of the entire system. One effective approach to gain insight into complex systems is to construct a model and test it against real-world observations. This approach is frequently employed in particle physics, utilizing the computational power of modern computers to simulate the behavior of the entire detector and read-out system. In a detector chain, each domain has its own specialized software for such modeling. The characteristics of an event in the individual channel are typically modeled with tools such as Geant4 [1], which simulate the particles' behavior in the active material of the detector. The electrical characteristics of the sensor and peripheral electronics and their signal shaping effects are modeled using a derivative of SPICE, an established system for circuit simulations. Finally, the digital post-processing of such signals can be modeled with the tools we already use for digital design, such as Vivado and Cadence. These tools are highly complex and are used only used by the experts who work on the particular subsystem. In order to derive requirements for our ASIC implementations, it is necessary to be able to study the signals and signal processing of a system using a common tool chain. We have chosen Matlab/Simulink to implement a custom approach for studying the mixed-signal processing in particle detectors. In the following, we will provide a brief overview of our approach to modelling and how we are applying it to our developments for a real detector prototype.

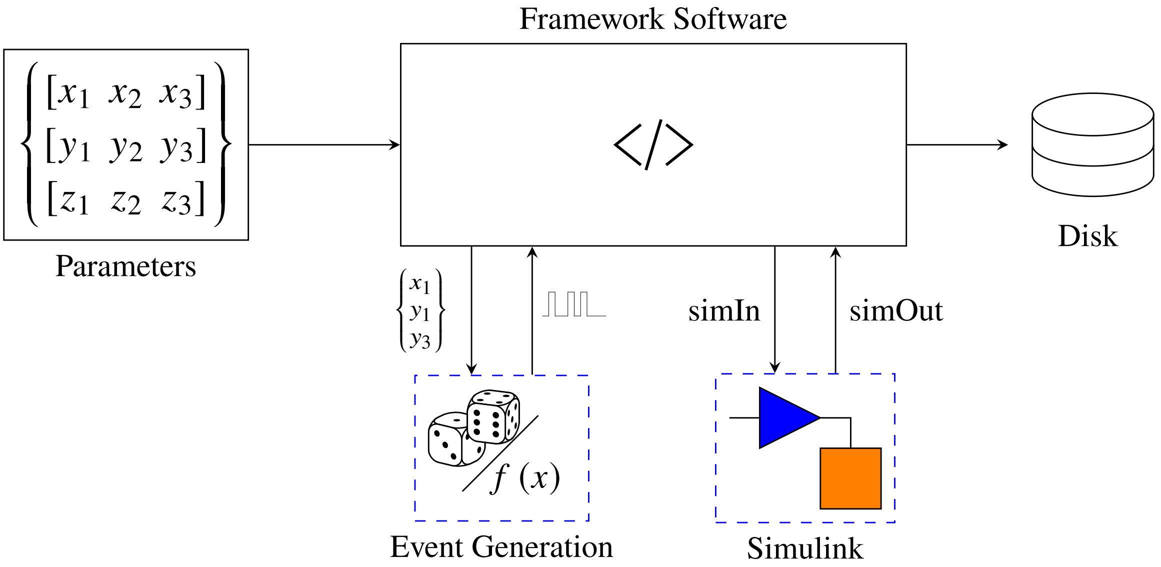

Framework Software
$\left\{\begin{matrix}[x_1\ x_2\ x_3]\\ [y_1\ y_2\ y_3]\\ [z_1\ z_2\ z_3]\end{matrix}\right\}$
Parameters
</>
Disk
$\left\{\begin{matrix}x_1\\ y_1\\ y_3\end{matrix}\right\}$
simIn
simOut
$f\ (x)$
Event Generation
Simulink

Figure 1. Overview of the central architecture our software. It takes care of distribution of parameters, interfacing and parallelization of simulations.

2 Simulation of particle detector read-out in Matlab and Simulink

2.1 Overview

Matlab and Simulink are standard tools in engineering for the simulation of complex control systems and mixed-signal systems. As such, they are a reliable tool for our developments. The basic concept and central operating principles are outlined in section 2.2. In section 2.3 we will briefly describe how we separated the modeling domains and evaluate their implementations individually. The performance of the simulations will be then discussed in 2.5. In section 3.3 the Simulink model for the Search for Hidden Particles (SHiP) Surrounding Background Tagger (SBT) Detector will be discussed as a case study for both the modelling approach and performance, with the aim of demonstrating the results achieved. These results will be presented in section 3.4.

2.2 Architecture

In order to explore parameter spaces for a detector read-out, the ability to perform sweeps of parameters in an efficient manner is essential. This is facilitated by a set of classes and functions that we have implemented in Matlab. The core functionality of repeated parameterized simulations is illustrated in figure 1. In a conceptual sense, the software employs a set of parameters to generate a set of events and model configurations. The specific models utilized in event generation and in Simulink must be implemented according to the users requirements, as indicated by the dashed outline of the components in figure 1.

2.3 Modelling domains

Our approach necessitates the investment of modelling effort across multiple fronts. Firstly, the input signal must be modelled in the active part of the detector, which in this context is referred to as event generation. Secondly, the response of the active detector segment to incoming signal quanta must be modelled. This is referred to as sensor modelling. The output of this sensor model is the detector signal current, which is then processed by analog electronics, digitized and further processed by digital signal processing means. These will be referred to as analog and digital processing models.

Event and sensor modelling. The input to the Simulink model reflects the time characteristics of the detector event. For the simulations performed in the context of this publication we have prepared results from Geant4 simulations. Alternatively custom Monte-Carlo or parameteric approaches can be implemented using the interfaces required by the software.

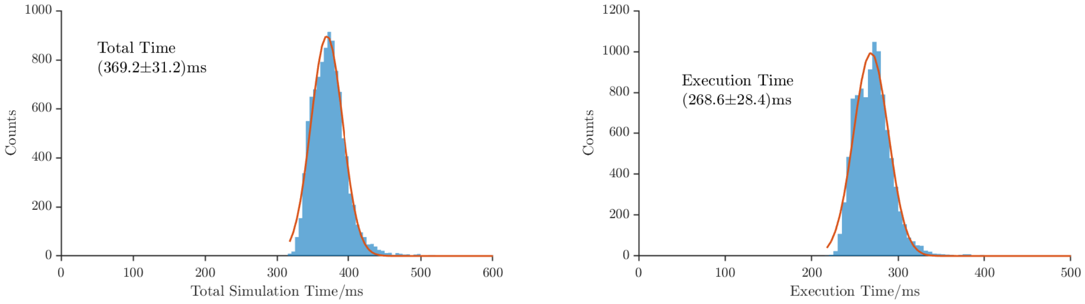
Simulink transient modelling. Simulink is a software tool that enables the transient simulation of complex models [2]. It facilitates the co-simulation of continuous and discrete time systems, rendering it an optimal choice for simulating both the analog and the digital domains of the signal processing chain for a particle detector. Our models typically comprise a continuous-time component, which models the behavior of analog electronics, and a discrete-time system, which is an implementation of the digital domain logic in Matlab. By deploying our digital logic onto an FPGA using the HDL Coder toolbox, we can test it in a digital hardware environment [3]. The analog domain of our model encompasses both the sensor response to incoming particles and the analog signal processing, including preamplifiers and additional filtering that might be performed for signal shaping. In practice, the user may choose the model complexity and accuracy, as the developed software does not rely on a specific model structure. As will be discussed later, we primarily utilized the Laplace domain representation (system transfer functions) for expressing and solving the equations of the electrical systems. This approach was selected as it is a straightforward method for modeling continuous-time signals and is well-supported in Simulink. Other potential alternatives include the creation of custom models or the utilization of the Simscape Electrical toolbox. In the digital domain, Simulink offers z-transform transfer functions and a plethora of other discrete-time logic blocks via integrated libraries for modeling. This enables the simulation of digital filters, the performance of mathematical operations on the signal, and the application of other kinds of transformations. Moreover, core functionalities of digital systems can be implemented in Simulink, including memory, state machines, and communication interfaces. Simulink supports multi-rate digital systems, allowing the user to take advantage of faster simulations by reducing the sample rate of subsystems through parallelization of data.

2.4 Simulation control

Simulink models can be controlled by Matlab on a run-to-run basis, using objects from the `Simulink.SimulationInput` class. These objects allow the parameters of a model to be set and inputs to be provided. The framework developed for this purpose provides functions to automatically create simulation objects and arrange them in a way that allows fast simulations of large datasets. It also utilizes Simulink's built-in precompilation of models and parallelization.

2.5 Simulation performance

For the purposes of evaluation, we ran 10 000 simulations on an Intel NUC9i7QNX with a 6 Core i7-9750H and 64 GB RAM. In the context of this evaluation, the model developed for simulation in the SHiP SBT application context was utilized. The model contains sub-models for the complete analog chain, as well as model of an on-FPGA SRAM for data buffering, including read and write logic and multiple feature extraction algorithms. Each simulation covers 1 µs of simulated detector



(a) Total time per simulation, including simulation setup and teardown.

(b) Time taken per simulation, without any pre- or post-processing functions.

Figure 2. Time performance of simulations using our software.

signal and contains a single detector event. The time per simulation is shown in figure 2. The utilization of all cores for parallel simulation results in a reduction of overall run time. As illustrated by the displayed data, the average simulation takes about 380 ms, including approximately 100 ms for model setup and tear down functions. The results indicate, that only approximately 75 % of the total time is utilized for the actual simulation. Consequently, longer multi-event or more complex simulations may be a more attractive option for improving the runtime utilization. Unfortunately Matlab does not offer other metrics for performance. Using more crude methods of manually checking hardware ressource usage showed a peak CPU utilization of >93 %, and a peak memory usage of about 18 GB. Hinting, that Simulink comes with a good ressource utilization in terms of CPU time. This points toward further gains in simulation speed per event can only be improved by using more or more performant CPUs for the simulations.

3 Application to the SHiP SBT detector

In order to demonstrate the accuracy and usefulness of the aforementioned software, we have chosen to apply the approach to an ongoing detector development. The Search for Hidden Particles experiment is a particle physics experiment proposed for the ECN3 beam dump facility of the SPS at CERN [4, 5]. The Surrounding Background Tagger sub-detector serves as a background detector for the large vacuum decay vessel, which constitutes the main part of the detector [6].

3.1 SHiP SBT detector introduction

The detector features a long vacuum decay volume, which allows for possible hidden particles to decay with a longer baseline. To track possible intrusions into the decay volume, which could influence the measurements of the downstream detectors, the decay vessel is surrounded by the SHiP SBT. This detector consists of large steel tanks filled with a liquid scintillator. The individual steel tanks are equipped with Wavelength-shifting Optical Modules (WOMs), absorb the scintillation light and emit it into the lightguide with a shifted wavelength. The light guide is mounted to a Printed Circuit Board (PCB) that hosts a ring of Silicon Photo Multipliers (SiPMs), which convert the collected light into electrical signals. The ring is made up from 40 individual Hamamtsu s14160-3050HS [7] silicon photomultipliers. For readout purposes, five SiPMs are connected in parallel to form a single read-out channel [5, 6]. The primary objective of the SBT detector development is demonstration of the ability to detect interactions. Secondary objectives include quantifying the deposited energy and a position resolution as high as possible. By combining the information from neighboring scintillator tanks, the reconstruction of a particles trajectory can be improved. The characteristics of photons measured by the SiPMs are largely dependent on the liquid scintillator, the geometry, and reflectivity of the inner tank walls and photo conversion and collection efficiencies of the WOM and SiPM.

3.2 SHiP SBT event model

Given the dependencies on tank geometry and scintillator properties, the simulation of the events is a task that is best left to specialized tools such as Geant4. For the purpose of this work, we are using the Geant4 model code that was written for the studies of a one-cell prototype detector [8]. The model accounts for a number of detector characteristics, including scintillation spectra and decay times of the liquid scintillator, wall reflectivity, absorption- and emission spectra of the wavelength shifting layer, the WOM transmission, and the wavelength-dependent sensitivity of the SiPMs.[1] Geant4 simulations were performed, each containing a single event, with muons of different energies, ranging from 100 MeV to 10 GeV penetrating the scintillator tank perpendicular to the large face at different positions. Hit positions and tank geometry are displayed in figure 3. While we certainly not cover the whole range of possible events expected in operation of the SHiP SBT sub-detector, the data obtained this way will be suitable as an initial gauge for the usability and reliability of our software. From the Geant4 simulation data, we can derive some fundamental limits for the performance of the detector. As illustrated in figure 4(a), the arrival time of the first photon varies by approximately 0.7 ns on SiPM group 0. We observe a comparable fluctuation on all groups. The combination of the groups yields a significantly improved timing fluctuation of approximately 0.3 ns. This sets the boundary for the achievable time resolution accordingly. Given the relatively high amount of dark counts present in SiPMs, triggering of the first photon is unlikely, and we will only be able to achieve a worse time resolution. With regard to the photon number resolution, shown in figure 4(b), it exhibits a fluctuation of ±20 % with respect to its mean value, thereby limiting the achievable resolution of the photon number estimator.

3.3 Simulink model of a single detector channel

The Geant4 output can now be used as the basis for the simulations in our framework. In Matlab, we add dark counts to the simulated event photons, with an adjustable dark count rate; the implementation is similar to [9]. In Simulink, we need to emulate the response of a SiPM to incoming photons. For this, we employ a modelling approach similar to [10] and [11], where the modeling of SiPMs in the context of the SHiP SBT has already been discussed. We use the equivalent circuit, as shown in figure 5(a). We add an idealized model of our front-end electronics to this, so we can account for the shaping effects of the coupling circuitry. The amplifier and coupling components are shown in figure 5(b).

In figure 5(a), the distinction between firing and passive cells of a SiPM is made. For a SiPM with quenching impedance R_q and C_q and the cell capacity C_d, for N total cells and N_f fired cells,

[1] We used a tank wall reflectivity of 100 %, which was determined to be unreasonable in test beam measurements, and the actual reflectivity was measured to be closer to 65 %. simulations with the updated Geant4 model and improved electronics will be published at a later stage.

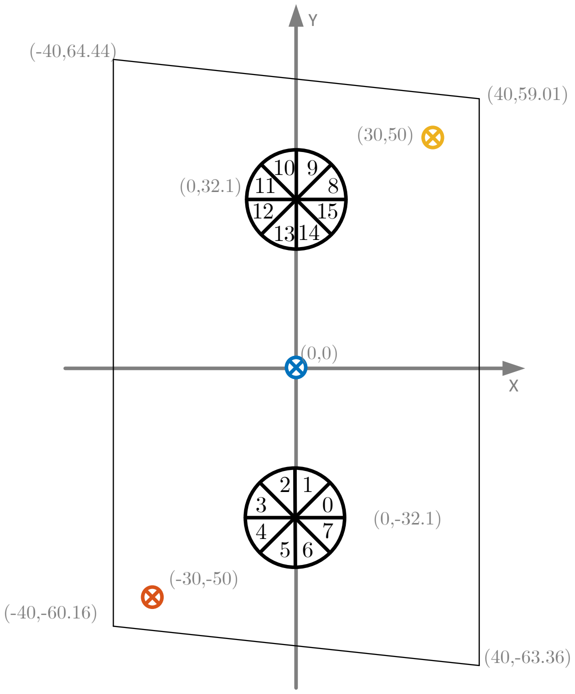
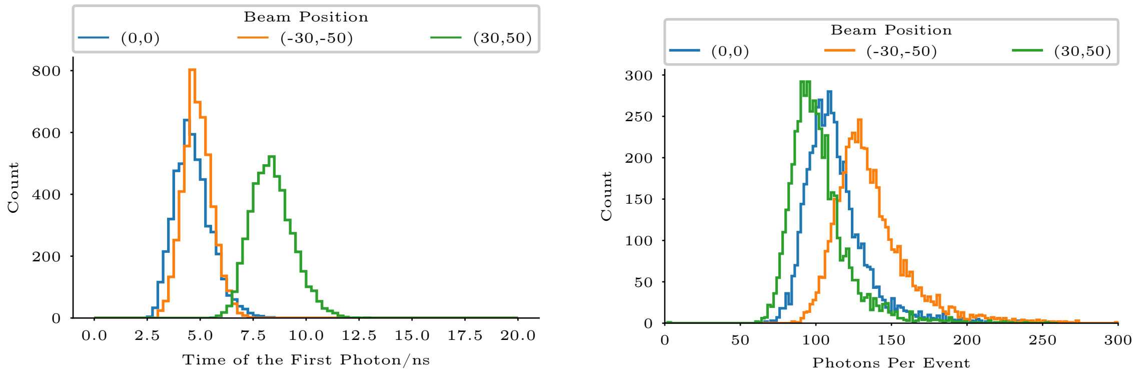


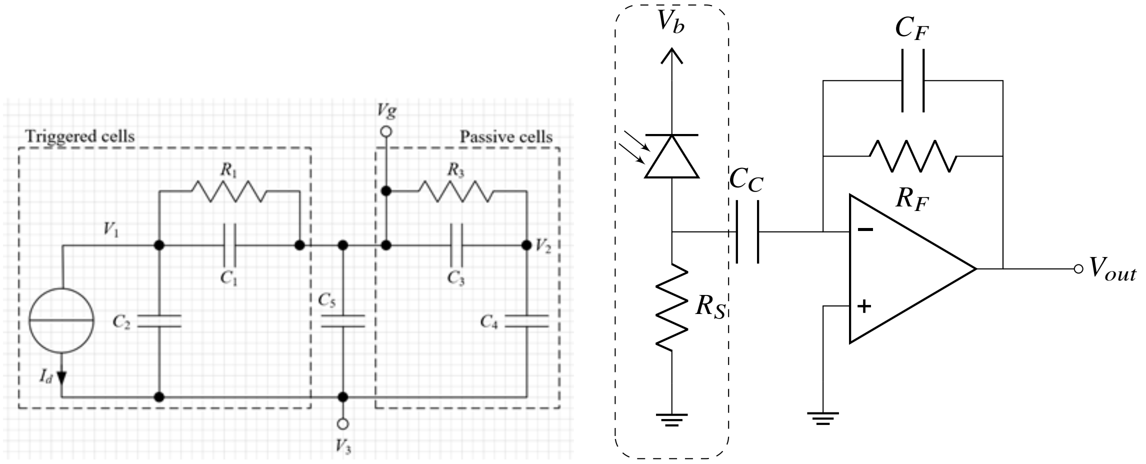
Figure 3. Sketched view of a SHiP SBT detector cell, with the WOM positions and SiPM groups indicated by the cake sliced circles, the numbers in the slices indicate the SiPM groups. The simulated beam positions are indicated by the crossed circles. All coordinates in centimeter, depiction not to scale.



(a) Arrival Time Distribution for the first photon of an event at SiPM Group 0. The individual distributions show a standard deviation on the arrival time of 0.7 ns with only a small deviation between the hit positions. The mean arrival time shows a strong dependence on the hit position, as can be expected.

(b) Photon number distributions for photons collected on SiPM Group 0 per event. The individual distributions show a variation if ~ 20 percent w.r.t. their mean.

Figure 4. Time and photon number resolution for the SHiP SBT one cell prototype, derived from Geant4 simulations.



(a) SiPM equivalent circuit model, modelled after [10]. (b) SiPM connected to a trans-impedance front-end.

Figure 5. Equivalent circuitry as used for modelling purposes. For the SHiP SBT single channel model, we use five SiPM structures, highlighted by the dashed box (b), in parallel.

the fired cells contribute $R_1 = \frac{R_q}{N_f}$, $C_1 = C_q N_f$ and $C_2 = C_d N_f$. The passive cells $R_3 = \frac{R_q}{N - N_f}$, $C_3 = C_q \left(N - N_f\right)$ and $C_4 = C_d \left(N - N_f\right)$. In figure 5(b), we have added the coupling components R_s and C_c that we plan to add in the signal path. We simulate an ideal amplifier, with the frequency limiting pole provided by the feedback structure. The following equations were used to derive our models:

$$I_d + sC_2\left(V_1 - V_3\right) + \frac{V_1}{R_1} + sC_1 V_1 = 0 \tag{3.1}$$

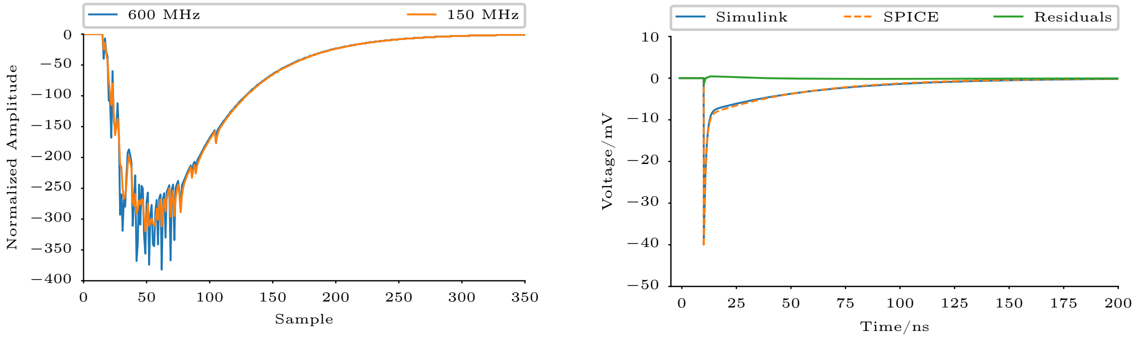
$$sC_4\left(V_2 - V_3\right) + \frac{V_2}{R_3} + sC_3 V_2 = 0 \tag{3.2}$$

$$I_d + sC_2\left(V_1 - V_3\right) + sC_4\left(V_2 - V_3\right) - C_5 V_3 = s + \frac{V_3}{R_6} + sC_c V_3 \tag{3.3}$$

$$Z_{in} = \frac{1}{sC_c}; \quad Z_F = \frac{1}{\frac{1}{R_F} + sC_F}; \quad \frac{Z_F V_{in}}{Z_{in} + Z_F} + \frac{Z_{in} V_{out}}{Z_{in} + Z_F} = 0; \quad V_{in} = V_3 \tag{3.4}$$

We can plug the solutions for V_3 and V_{out} into a Simulink transfer function block and thus simulate the analog electronics. We performed a proxy verification by comparing the simulation results from Simulink to the SPICE model from [10] in figure 6(b).

Simulink can be employed to model a multitude of properties associated with an ADC, encompassing both basic parameters such as ADC bit resolution and intrinsic nonlinearities. The ADC output can then be utilized to assess the efficacy of various Digital Signal Processing (DSP) methods in terms of their performance. For the purposes of this paper, we have tested a few basic approaches. The most common approach for timing and triggering is to apply a threshold comparison to the signal and record the time of threshold crossing as the signal timestamp. It is typically implemented using a fast comparator and a TDC. In this context, we simulate a digital threshold as a baseline algorithm for triggering and timing, as we plan to implement the read-out using only an ADC. Parallel to that, we looked at a simple integration of the signal current as an estimator for the total charge created in a detector, and thus for the total deposited energy. Or in the case of the SHiP SBT the total number of photons collected. As a baseline, we will integrate the signal in the digital domain,



(a) Example waveform from simulations. Higher bandwidths lead to the fast SiPM components becoming prominent and contributing large spikes to the signal, that could potentially reduce resolution. The slower base pulse, is the light curve of the liquid scintillator.

(b) Comparison of SiPM signals captured with the SPICE model from [10] and our implementation in Simulink. Residuals are derived from a linear interpolation. The observed residuals are an order of magnitude smaller than the simulated signal.

Figure 6. Waveforms as obtained from our simulation models. On the left using simulated photon arrivals from a cell of the SHiP SBT, on the right from a single photo electron.

by summing up all samples above the threshold, which should give a good estimator for the number of photons. We also looked at sampling the peak of the signal and we recorded the signals Time over Threshold (ToT), which can both serve as a photon number estimator, and also work as a good filter parameter for filtering noise triggers.

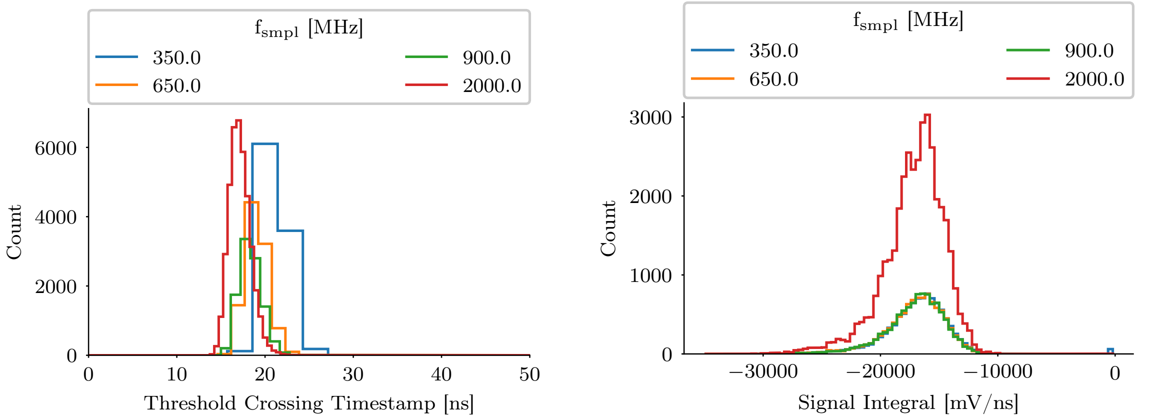
3.4 Simulation results

In order to narrow down the parameter space for a possible ASIC implementation, we looked at multiple parameter influences. For each simulation run, we simulate a single model parameterization with 10000 different events, taken from our Geant4 datasets. We have kept a trans-impedance of 500 Ω and an ADC dynamic range of −0.3 V to 0.3 V,[2] and fixed the SiPM parameters according to [10]. As a first exploration of the parameter space a variety of parameter sweeps were performed, by using a default model configuration and sweeping a single model parameter against that parameterization. From these we obtain histograms of our observables for different parameter values. Examples of those histogramms are shown in figure 7. For every parameterization an appropriate distribution is fit to the histogram, using bootstraping [12]. In figures 8 and 10 we report the fitted standard deviation of the fit, with the parameter uncertainty derived from the bootstrapping results.

The figures 8(a) and 8(c) show how the distribution width of the first timestamp is affected by the sample rate and the front-end bandwidth. The influence of the ADC sample rate seems to be small, compared to effort. From a reconstructed timing resolution of 1.4 ns to ~1.2 ns would require a fourfold increase in sample rate, leading to a highly increased complexity in implementation.

Contrary to usual cases, we observe a degradation in time resolution with increasing bandwidth, as the influence of the high frequency components of the SiPM signals become more dominant than the slower underlying scintillator light curve. Especially for low bandwidths we observe a high,

[2] We chose these parameters because they matched the ADC we plan to use for verification.



(a) Reported time stamp from the simulink model. The binwidth is invers proportional to the sample rate.

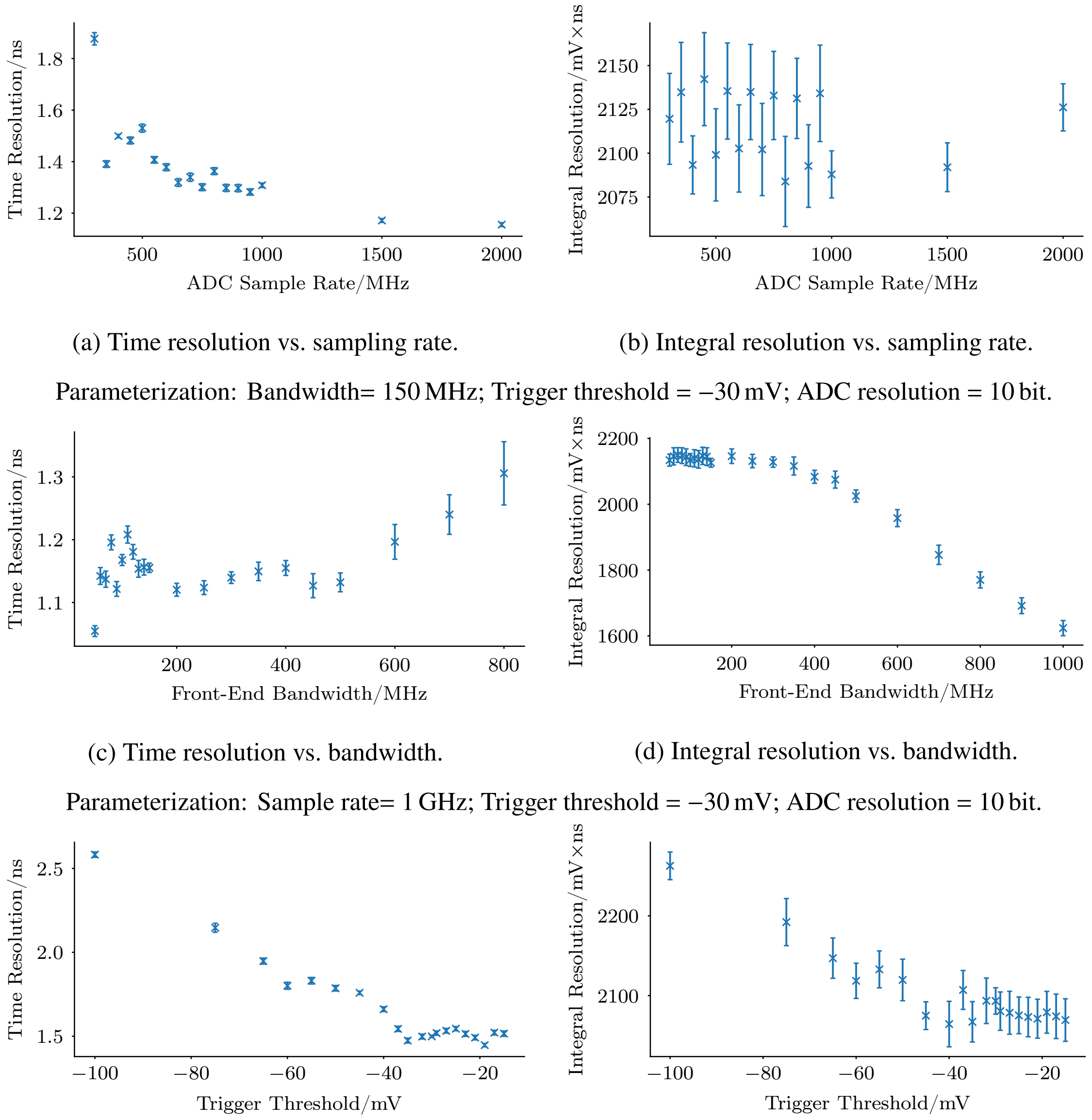
(b) Signal integral as reported by the simulink model.

Figure 7. Reported event features from the Simulink model for different ADC sample rates. Some datasets contain more data, as they have been simulated more than others.

unecpected fluctuation in between bandwidths. We attribute this behavior to a not ideal fit model, for the distribution of timestamps. A gaussian distribution was used to fit the histogram, while espacially for high bandwidth we observed that the histogram showed two prominent extrema in close proximity, indicating that there are two different effects that can contribute to the signal rise time. This indicates that also for low bandwidths a non-symetric distribution has to be chosen. This systematic effect will be investigated further, later on. Parallel to that, figures 8(b) and 8(d), show the behavior of the integral resolution. The integral resolution overall seems to be less sensitive to changes in the sample rate and shows an improvement towards higher bandwidths. We have observed a negligible influence of the ADC bit resolution on both timestamp and integral resolution.

Figures 8(e) and 8(f) show the dependence on the trigger threshold. A clear dependence on the trigger threshold is evident for both observables. In the timestamp domain, we see a jump below 25 mV, which stems from the trigger now being sensitive to individual photons, with the single photon signal peaking at about 20 mV, as shown in figure 6(b). We also observe two steps at approximately −30 mV and −60 mV that we can not fully explain yet. From the plots, we can conclude, that the trigger threshold should be chosen as low as possible. However, choosing a threshold too close to the single photon peak would increase the amount of triggers, hence a compromise has to be made. More investigations, including the effect of a high dark-count rate, have to be performed here.

We also looked at the effect of RMS noise. In order to obtain clean data, the reported data from the model had to be filtered. As is shown in figure 9, a series of cuts were applied. Events where the reported peak exceeded the maximum of the ADC digitization range were excluded, as were events with a ToT of less than 25 ns or an integral of more than −2 V × ns. The results of fits to the observables distributions are shown in figures 10(a) and 10(b). We used band-limited white noise by adding it to the front-end output. Both time and integral resolution show the expected degradation of resolution with increasing noise, as shown in figures 10(a) and 10(b).



(a) Time resolution vs. sampling rate.

(b) Integral resolution vs. sampling rate.

Parameterization: Bandwidth= 150 MHz; Trigger threshold = −30 mV; ADC resolution = 10 bit.

(c) Time resolution vs. bandwidth.

(d) Integral resolution vs. bandwidth.

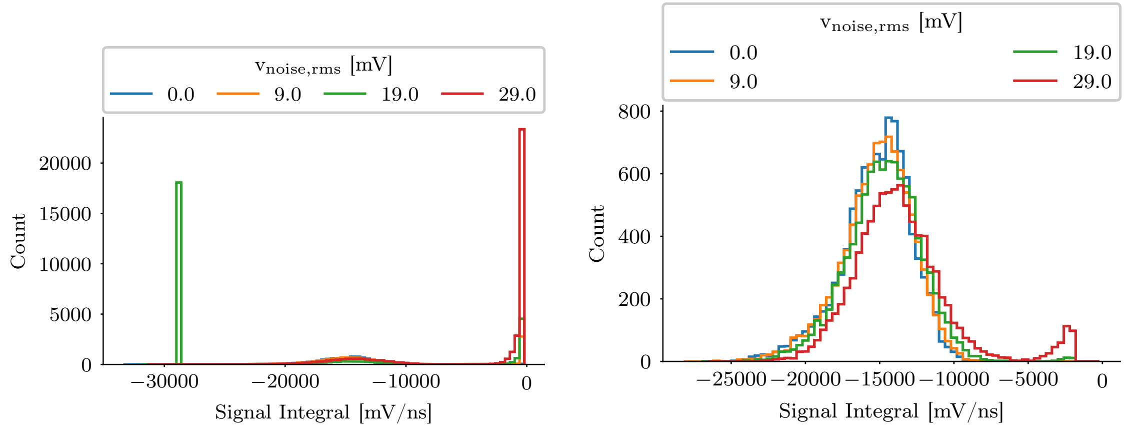
Parameterization: Sample rate= 1 GHz; Trigger threshold = −30 mV; ADC resolution = 10 bit.

(e) Time resolution vs. trigger threshold.

(f) Integral resolution vs. trigger threshold.

Parameterization: Sample rate= 400 MHz; Bandwidth = 150 MHz; ADC resolution = 10 bit.

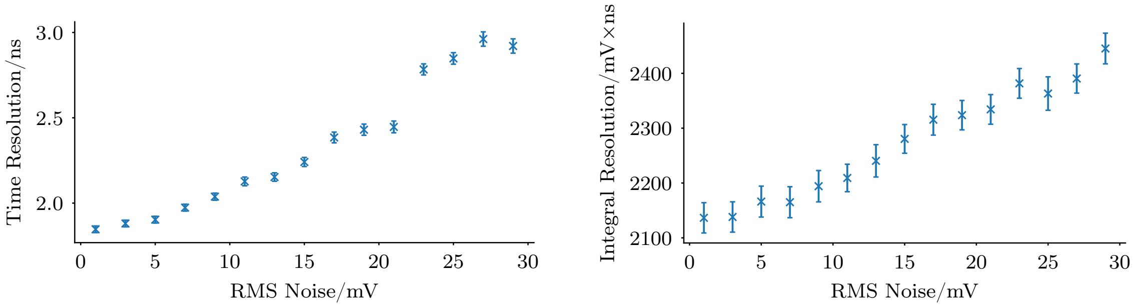
Figure 8. Influence of different parameters on the estimated timestamp and integral, regarding only cases where no noise was present in the system. The error bars denote the variation introduced by the different parameterizations that were simulated. Only noiseless cases with no dark counts were regarded for the plots.



(a) Unfiltered, reported signal integral for different rms noise values.

(b) The same data after application of the filters described in the text.

Figure 9. Reported signal integrals, filtered and unfitlered. High noise causes a lot of spurious misstriggers. Through proper datafiltering these can be largely omitted.



(a) Time resolution vs. RMS Noise.

(b) Integral resolution vs. RMS Noise.

Figure 10. Influence of the RMS noise on time and integral resolution, normalized by the applied trigger threshold.

4 Comparison to existing approaches

There are many different approaches to simulate parts of a particle detector system. Most prominent the Monte-Carlo simulators for the phyiscal processes, such as Geant4. Additionally there are models for sensor effects, such as [9] or equivalent circuit models such as reported in [13] and [11]. There are also mathematical signal models such as used in [14]. While we use all those individual models, we are not aware of a system that, like our own implementation, stitches together these models to analyse the performance of a complete read-out chain by performing repeated transient simulations. Typical approaches use the outcomes of individual investigations of the subcomponents to smear distributions such as those shown in figures 7 and 9 to achieve an estimation of achievable performance. Our approach reaches beyond this by modelling more of the electronics characteristics and digital logic, giving a more indepth picture. Unfortunately there was no room for comparative studies to be performed in parallel to our work.

5 Conclusion and outlook

With this work we wanted to present the basic concepts of a software framework for simulating the effects of particle detector read-out electronics on the observables that are of interest for the physics analysis, together with application to the SHiP SBT detector. As demonstrated above the approach is able to perform numerous simulations in a relatively short time frame. By combining models from the different domains of a particle detector, ranging from sensor responses to analog electronics for shaping to digital signal processing, a read-out system can be studied across parameters in all those domains. While the models have to be implemented for each individual detector implemention, the combination of the models in a single framework allows for fine-tuning of the entire single channel system.

Using the SHiP SBT as an example, we have shown that we can produce meaningful data, that can be used to make predictions for possible hardware implementations. We are already working on an updated model that has improved analog electronics models and conduct simulations using an adjusted Geant4 model.

To further exploit these capabilities, implementation of optimization algorithms around the above described funcitonalities would be a logical next step. Given our detector agnostic functions, we can easily develop models for a PET detector and use the same software to perform design space explorations for a PET read-out system.

In the future, we will use our software for the development of possible techniques and algorithms that could help with the improvement of the performance of the read-out for the SHiP SBT detector. Especially the time resolution could potentially be improved using a suitable interpolation approach, similar to [14]. We plan to verify our simulations with live measurements on a SHiP SBT detector cell in 2024.

Acknowledgments

We want to thank our collaborators from within the SHiP Collaboration for the support of this work. Espacially we thank Andrii Kotenko and Vladyslav Orlov for important contributions to the Geant4 Model of the SBT detector cell.

References

[1] Geant4 collaboration, *Geant4 — a simulation toolkit*, *Nucl. Instrum. Meth. A* **506** (2003) 250.

[2] The MathWorks Inc., *MATLAB Version: 9.12.0.2039608 (R2022a) Update 5*, The MathWorks Inc. (2022).

[3] The MathWorks Inc., *HDL Coder Toolbox*, The MathWorks Inc. (2022).

[4] W. Bonivento et al., *Proposal to Search for Heavy Neutral Leptons at the SPS*, arXiv:1310.1762.

[5] SHiP collaboration, *The SHiP experiment at the proposed CERN SPS Beam Dump Facility*, *Eur. Phys. J. C* **82** (2022) 486 [arXiv:2112.01487].

[6] J. Alt et al., *Performance of a first full-size WOM-based liquid scintillator detector cell as prototype for the SHiP Surrounding Background Tagger*, 2024 *JINST* **19** P05024 [arXiv:2311.07340].

[7] Hamamatsu Photonics, *MPPC S14160-3050HS Silicon Photo Multiplier*, (2020).

[8] F. Lyons, *GitHub — fairlyons/ship_wom_testbeam at 4cells*, https://github.com/fairlyons/ship_wom_testbeam.

[9] J. Pulko et al., *A Monte-Carlo model of a SiPM coupled to a scintillating crystal*, 2012 *JINST* **7** P02009.

[10] J. Grieshaber, *Calculation and Simulation of Silicon Photomultiplier Signals*, Bachelor Thesis, Albert-Ludwigs-Universität Freiburg, Freiburg, Germany (2022).

[11] F. Villa et al., *SPICE Electrical Models and Simulations of Silicon Photomultipliers*, *IEEE Trans. Nucl. Sci.* **62** (2015) 1950.

[12] B. Efron and R.J. Tibshirani, *An Introduction to the Bootstrap*, Monographs on Statistics and Applied Probability, Chapman & Hall/CRC (1994) [DOI:10.1201/9780429246593].

[13] A. Rivetti, *CMOS: Front-End Electronics for Radiation Sensors*, Taylor and Francis (2017) [DOI:10.1201/b18599].

[14] L. Jokhovets et al., *Improved Rise Approximation Method for Pulse Arrival Timing*, *IEEE Trans. Nucl. Sci.* **66** (2019) 1942.