

Chicago, EFI/EDG, July 17th 2008.
J-F Genat

Simulations of electronics timing techniques.

1- Simulation

This package simulates five different electronics timing strategies used in fast photo-detection, using Matlab.

- Single threshold on leading edge
- Constant fraction as zero-cross of (delayed – attenuated) input
- Multiple thresholds on leading edge
- Pulse sampling and reconstruction

The main program (simtime.m) can be found in the root directory, each method being called from specific subdirectories.

Parameters are set in the .param files (see below)

1- Input signals

A database of ‘stat’ different signals is built using ‘database’.

The detector pulse (wave.m) is a triangular waveform to which noise is added (shot noise)

convolved by a transfer function $\frac{1}{\tau} \exp(-t/\tau)$ where τ is linked to the detector and

electronics analog bandwidth ‘bw’, by the relation $bw = .35/\tau$. White noise is also added.

Noise contributions can be tuned through the detector shot noise ‘Ishot’ in detector_params.m, and the thermal noise resistance ‘R’ in elec_params.m

Respective and global signal to noise ratios are reported

The signal amplitudes are distributed according to the input signal statistics in terms of number of photo-electrons ‘npe’.

For pulse sampling, a reference pulse and its derivative are added to the database as two extra signals, since the signal average waveform has to be known.

Parameters of the simulations taken into account are:

- The (fixed) math time step ‘tstep’
- The signals rise and fall time ‘trise’ and ‘tfall’
- The analog bandwidth ‘bw’
- The temperature ‘T’ (Kelvins)
- The number of photoelectrons ‘npe’
- The gain ‘G’

- The shot noise “Ishot”
- The white noise thermal resistance “R”

All units are default SI.

Specific timing method parameters such as the number of thresholds “th_n”, the thresholds “th” for threshold techniques, the constant fraction “frac”, the delay “delay”, the threshold for constant fraction, the sampling frequency “fs”, the sampling jitter “apjit”, and the quantization “nadc” for pulse sampling.

2- Parameters

A typical set of parameters corresponding to Micro-channel Plate detectors (MCP) can be used :

Parameter	File
Simulation and detector parameters:	
	detector_params
tstep=500e-15;	Math time step
trise=200e-12;	Rise time
tfall=0;	Fall time
npe=50;	Photo-electrons
Ishot=300e-12;	Shot noise current
T=300;	Absolute temperature
Electronics parameters	
	elec_params
bw=1e9;	Bandwidth
G=1e5;	Gain
nadc=8;	ADC number of bits
apjit=2e-12;	Sampling jitter
fs=40e9	Sampling frequency
Single threshold	
	edge_params
th_n=1	Single threshold
th=0.4;	Threshold (fraction of the average pulse amplitude)
Constant fraction	
	cfp_params
frac=0.5;	Constant fraction
thresh=0.4;	Trigger threshold

delay=180e-12;

Delay

Multiple-thresholds

th_n = 4
th=[0.1 0.2 0.3 0.4];

edge_params

Four thresholds
Thresholds (fractions of the average pulse amplitude)

Figure 1 illustrates Multiple-threshold and Sampling methods.

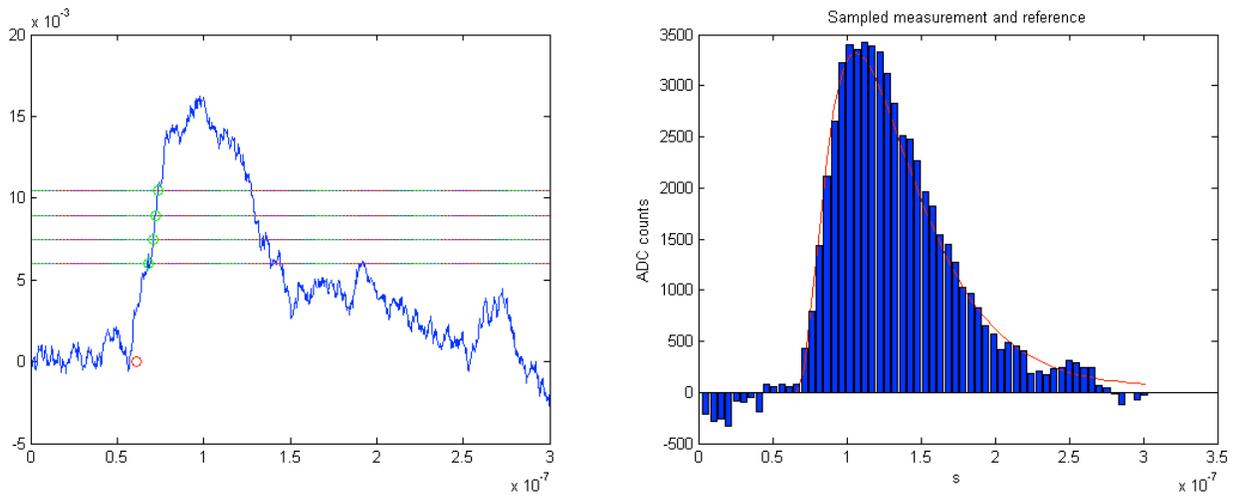


Figure 1 Multiple-threshold and Sampling (for Silicon strips signals).

2- Running the simulation

After having set the proper parameters in the parameter files depending on the method to be used:

- detector_params
- elec_params

for all methods,

- cfd_params
- edge_params
- sampling_params

specific to the method to be used,

The user should first set the database using for instance the command:

```
[fy, snn, sn_mcp, sn_elec, t0, i0 ,i0_0, maxsig, tstep]=database(300)
```

The database function returns a set of 300 signals $fy(1:300)$ and two references $fy(301:302)$ allowing running any of the implemented timing methods. It returns also the mean signal to noise ratios from the actual signal distribution. The signals are randomly shifted in time within $\pm \tau$, and in amplitude according to “npe”.

[A set of databases are provided such as NP50_SN79_1G_300.mat for:

50 Photo-Electrons
79 Signal to Noise ratio
1GHz analog bandwidth
300 samples]

Databases can be saved and reloaded using the Matlab commands save ‘database’ and load ‘database’. The “load” statement within the simtime main program controls the database to be used.

(It looks that Matlab cannot handle a “load” command outside an internal call to a function/directory, so a database loaded from the main directory is not seen inside simtime/directories)

The user can visualize the signals using: **dataplot(stat)**, where “stat” is the number of signals to be generated.

The user selects the one of the following commands:

Single-threshold, Multiple-threshold

```
[stdt, mean_snn]=edge_stat(fy, snn, i0, t0, stat);
```

Constant fraction

```
[stdt, mean_snn]=cfd_stat(fy, snn, maxsig, i0, t0, stat);
```

Sampling:

```
[stda, stdt, mean_snn]=sampling_stat(fy, snn, i0, t0, stat);
```

The simulation returns the spread in time stdt, the spread in amplitude, stda if sampling is used, the actual average signal to noise ratio mean_snn, and displays the corresponding histograms.

**For all comments and remaining identified bugs, report to genat@hep.uchicago.edu
Thanks !**