

# Project report on Monte Carlo simulations of proposed LET instrument at ISIS target station

## 2

Kaspar Hewitt Klenø  
Niels Bohr Institute  
University of Copenhagen  
kaspar@fys.ku.dk  
Supervisors:  
Kim Lefmann, Risø  
Robert Feidenhans'l, University of Copenhagen  
May 6, 2006

### 1 Introduction

The purpose of this project, has been to investigate the importance of 'jitter' in the beam choppers of the proposed LET instrument, so as to better understand the instrument before the design is finalized.

Jitter is to be understood here as imperfections in the phase alignment of the choppers, due to engineering limitations. The aim has been to predict which level of jitter can be accepted, before the precision of the instrument begins to degrade significantly, thus enabling more precise design requirements.

These simulations have been done using the Mcstas software, a neutron ray-trace simulation package[1].

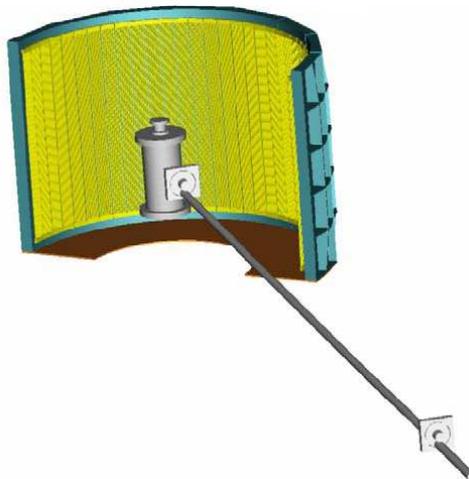


Figure 1: The LET spectrometer showing the large detector bank and multiple high speed disk choppers[2]

## 1.1 About LET

LET (Low Energy Transfer) is a cold neutron multi-chopper spectrometer, located at target station 2 of the ISIS pulsed neutron source, at the Rutherford Appleton Laboratory.

From the instrument homepage: *LET will be a versatile direct geometry spectrometer operating over a wide 0-80 meV energy range, with an energy resolution ranging from 5  $\mu$ eV at an incident energy of 1 meV, or 200  $\mu$ eV at 20 meV. It will have a large detector solid angle populated with MAPS type position sensitive detectors (PSD) that will make it possible to map a vast swathe of Q-E space in a single measurement.*[2]

Of special interest for this project, is the chopper system shown in figure 2

A quick rundown of the choppers, is necessary to understand the project:

- Res1 and Res2

These are the high speed resolution choppers, responsible for narrowing the energy resolution, as can be seen by the results in figure 4, and are positioned as the first and last choppers. Both are double choppers, with the second chopper counter rotating, which required some component code alteration. Res2's unusual windows also called for some custom code, as detailed in the next section.

- FO

The frame overlap chopper is responsible for removing slow neutrons from the previous pulse (otherwise they would cause spurions). Only a single pulse is simulated in the program, however we know that no slow neutrons from this pulse gets through, thus no slow neutrons from an earlier pulse would be transmitted either, were one to simulate this.

- PR

The pulse removal chopper thins out the pulses, so as to separate the desired pulses.

- CR

The contaminant removal chopper removes fast neutrons coming from the tail of the pulse (spurions) as seen in figure 3.[4]

## 1.2 jitter

Jitter is added to give more physically realistic beam choppers, i.e. choppers which are not perfectly in phase, but are rather off by a random error from the theoretical phase, at any given time. This error is chosen to be gaussian, but the true error will of course depend of the mechanics of the actual chopper. The size of the jitter is then defined as the width of the gaussian error.

## 2 Programming

In the field of neutron Monte Carlo simulations, substantial efforts are currently being made to perform virtual experiments. Virtual experiments are full, accurate simulations of neutron

instruments including realistic sample models. The goal of virtual experiments is to facilitate instrument design, experiment preparation, on-the-fly simulations, and data analysis, and represents a key focus area for the MCNSI [Monte Carlo Neutron Scattering Instruments<sup>1</sup>] joint research activity [at Risø, Denmark and 6 other laboratories in Europe][5].

In Monte Carlo ray-tracing, one generates a neutron with random parameters at the sample and follows it through the instrument. The ray-tracing package used in this work is McStas, developed at Risø.

The rest of this section will presume the reader has a basic knowledge of McStas. Refer to appendix for the actual code.

## 2.1 Building the instrument

The simulated instrument is build up quite analogous to the actual instrument:

- First the source, which simulates the ISIS moderator and is available as a downloadable component for McStas
- After 1.7 m 'down beam' the guide starts, continuing for 6.13 m
- At 7.83 m the Res1 set of counterrotating choppers is placed
- At 7.83 m a new 37 cm section of straight guide begins
- The frame overlap (FO) chopper at 8.2 m
- At 8.2 m another straight guide section, 3.55 m long
- The pulse removal (PR) chopper at 11.75 m
- At 11.75 m a another straight guide section, 3.91 m long
- The contaminant removal (CR) chopper at 15.66 m
- At 15.66 m, the last straight guide, 6.715 m long
- The split guide and the Res2 counterrotating choppers at 23.5 m.  
The split guide (as seen in figure 2) is build up by 4 sections of guide:
  - First a 1.1 m long pair, beginning immediately after the previous straight guide, at 22.375 m 'down beam', and narrowing from 19 to 10 mm.
  - 25 mm after this pair ends, the Res2 choppers are placed, at 23.5 m down beam
  - At 23.525, the second 1.1 m long guide pair begins, widening from 11 to 19 mm.
- 37.5 cm after this last guide section ends, is the sample position at 25 m down beam.

Various monitors are then interspaced throughout the instrument.

The whole idea behind the split guide and the counter rotating choppers, is to narrow down the energy resolution of the instrument. Refer to [6] for further discussion on this.

Although there is a large set of allowed frequencies for the instrument choppers, the following set has been used for all simulations in this project:

---

<sup>1</sup>mcnsi.risoe.dk

- Res1 and Res2 runs at 140 Hz
- PR and CR runs at 70 Hz
- FO is always fixed at 10 Hz

Refer to [4] for further discussion of allowed chopper frequencies.

## 2.2 Instrument code

Several parameters are needed in order to control the instrument, two of these are worth noticing:

- The phase of the choppers needs to be calibrated according to the energy (and thus speed) of the desired neutrons. This is selected with the parameter 'E\_foc'. The phase of the choppers, in dimension of time, is then calculated by:

$$pha = \frac{chopperdistance}{\sqrt{\frac{2 * E_{foc}}{m_{neutron}}}}$$

- Since the peak of neutron intensity is not emitted from the pulse at t=0, as can be seen in figure 7a, there need to be a phase offset in the choppers, so that they are focused on the maximum intensity of the source. With an ideal pulsed source, this would not be needed, as can be seen in figure 4a. Simulations shows that the highest intensity of 5 meV neutrons are received at the sample, with 'pha\_off' set to 120  $\mu$ seconds, as can be seen in figure 6, which seems quite reasonable, given the source profile. This value is then simply added to the previously calculated phase.

In the instrument code itself, the only special component needed, were the choppers, as explained in the next section<sup>2</sup>. Note that the Mcstas component 'Guide channeled' is used instead of the standard guide component, to allow for different m values for the top-bottom and the sides of the guide sections.

The split guide design for the Res2 choppers, are coded by placing two guide sections as they would be in the lab, and then using the GROUP command on them. (Without the GROUP command, Mcstas would require each neutron to pass both guide sections, which is not possible.

## 2.3 Chopper code

2 special chopper features were needed for this project:

### 1. Jitter

Jitter is calculated as a random floating number, with a normalized gaussian probability distribution, centered at zero. This is then multiplied with the global jitter parameter, giving the width of the distribution (default is zero for no jitter). This number is then added to the phase. A new random number is used each time a neutron hits a chopper. Note that phase here has dimension of time, and hence also jitter. Multiplying phase with angular frequency returns the ordinary radial phase.

---

<sup>2</sup>The modified chopper component have been incorporated as a standard mcstas component

## 2. Beamstop in chopper windows

The Res2 choppers have a beamstop in the middle of the chopper windows, the width of which is given by the parameter 'wc'. If a neutron hits outside the chopper windows *or* inside the beamstop, it is - of course - absorbed. This feature is due to the double funnel design.

As of release 1.9.1, these changes have been incorporated into the standard Mcstas disk chopper component. Refer to the Mcstas component manual for further details on this.[1]

## 3 Simulation results

I will now present the results of a series of McStas simulations of the instrument described in section 2. The graphs are either intensity vs. time of flight (TOF) or intensity vs. energy, unless otherwise stated. All simulations have been made with a source energy range of 4.9-5.1 meV<sup>3</sup>, again unless otherwise stated.

### 3.1 Simple source

This was a simulation using a simple pulsed source, where the entire pulse was emitted at t=0 (no tail). All neutrons are transmitted through the choppers, as would be expected, since they are all exactly in phase for all events, due to the absence of a tail. See fig 4 for the simulation results.

Note that after Res1, the width of the energy distribution is 38  $\mu\text{eV}$ , which is reduced to 3  $\mu\text{eV}$  after Res2. To first order the energy distribution after res1 is triangular, the exact formula is a bit more complicated:

The intensity,  $I(t)$ , must be proportional to how much of the guide opening is covered by the chopper window, which will then decline proportional to how much time has passed since chopper was perfectly aligned, at  $t_0$ :

$$I(t) = I_0[1 - k * (t - t_0)]$$

$k$  will then be given by the inverse of the time for the chopper to move from completely open, to completely closed:

$$k = 2 * \frac{\omega * r}{W}$$

Note that the extra factor 2 in  $k$  comes from the counter rotating chopper.

Time on arrival difference can then be translated into an energy difference for the neutrons:

$$t - t_0 = d_{chopper} * (v^{-1} - v_0^{-1}) = d_{chopper} * \sqrt{2 * m_n} * \left(\sqrt{\frac{1}{E}} - \sqrt{\frac{1}{E_0}}\right)$$

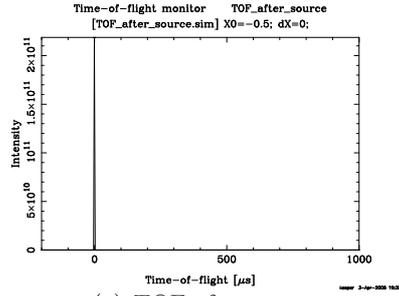
where  $d_{chopper}$  is the distance from source to chopper.

This all combines to:

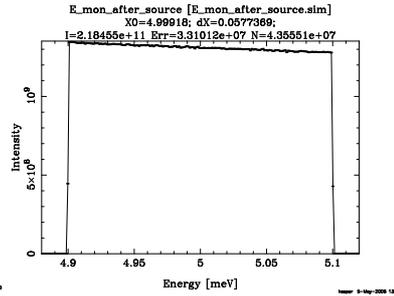
$$I(E) = I_0\left[1 - \frac{\omega * r}{\frac{W}{2}} * d_{chopper} * \sqrt{2 * m_n} * \left(\sqrt{\frac{1}{E}} - \sqrt{\frac{1}{E_0}}\right)\right]$$

---

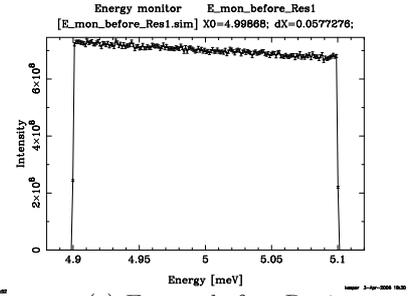
<sup>3</sup>See section 3.3.2



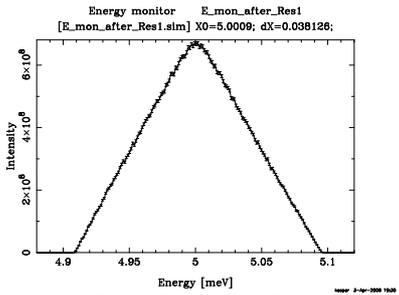
(a) TOF after source



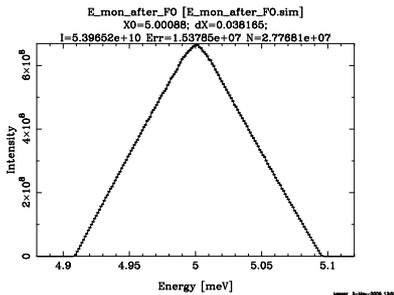
(b) Energy after source



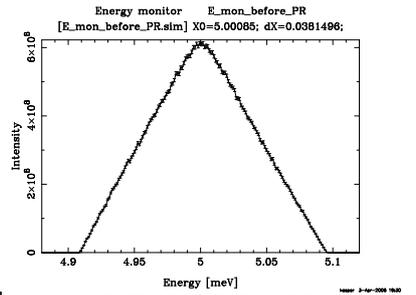
(c) Energy before Res1



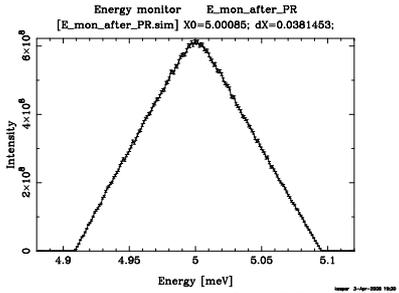
(d) Energy after Res1



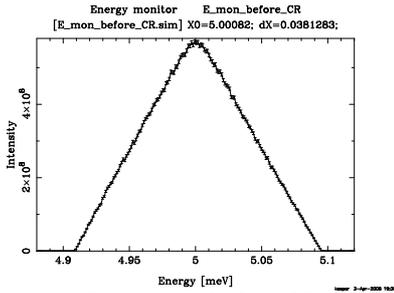
(e) Energy after FO



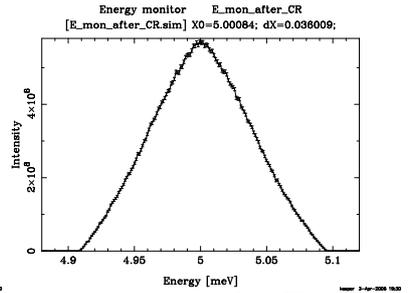
(f) Energy before PR



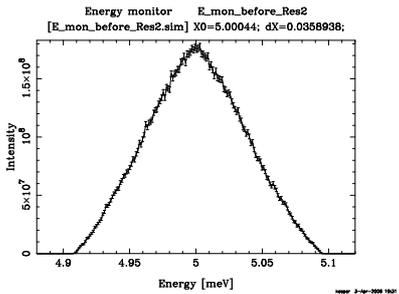
(g) Energy after PR



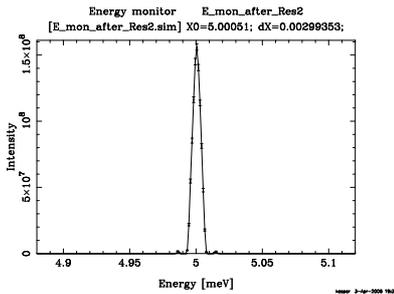
(h) Energy before CR



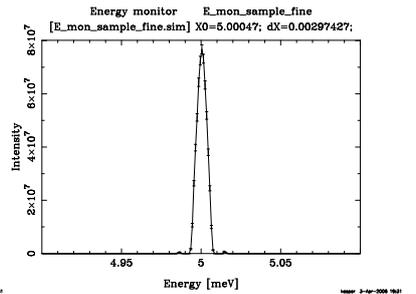
(i) Energy after CR



(j) Energy before Res2



(k) Energy after Res2



(l) Energy at sample

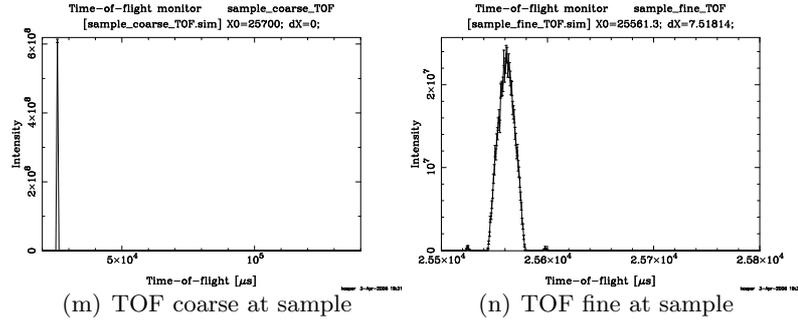


Figure 4: Simple source

Notice the small side peaks seen in graphs k,l and n. These are explained in section 3.3.1.

### 3.1.1 Guide system

Same as before, here the intensity of 5 meV neutrons is shown at the beginning and end of the guide.

As can be seen, the peak intensity is reduced from  $30 \times 10^8$  to  $2.3 \times 10^8$ , corresponding to a transmittivity of about 7.7 % for 5 meV neutrons. Since the simple source is used, 5 meV neutrons will not be touched by the choppers, so all reduction in intensity, will be due to absorption in the guide.

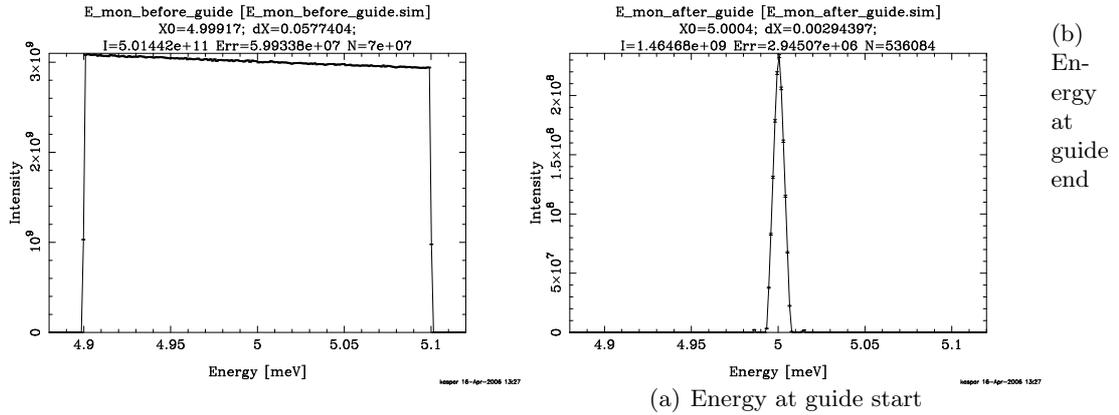
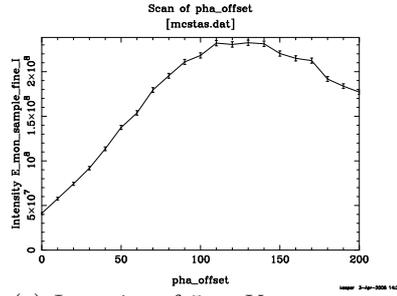


Figure 5: Intensity before and after the guide system, showing the transmittivity

### 3.2 Phase offset

A scan of the phase offset, from 0  $\mu$ seconds to 200  $\mu$ seconds in steps of 10  $\mu$ seconds, showing the intensity of 5 meV neutrons at the sample position, as a function of phase offset. As can be seen, the intensity is highest at 120  $\mu$ seconds phase offset. Thus 120  $\mu$ seconds phase offset will

be used in all subsequent simulations focusing on 5 meV neutrons.



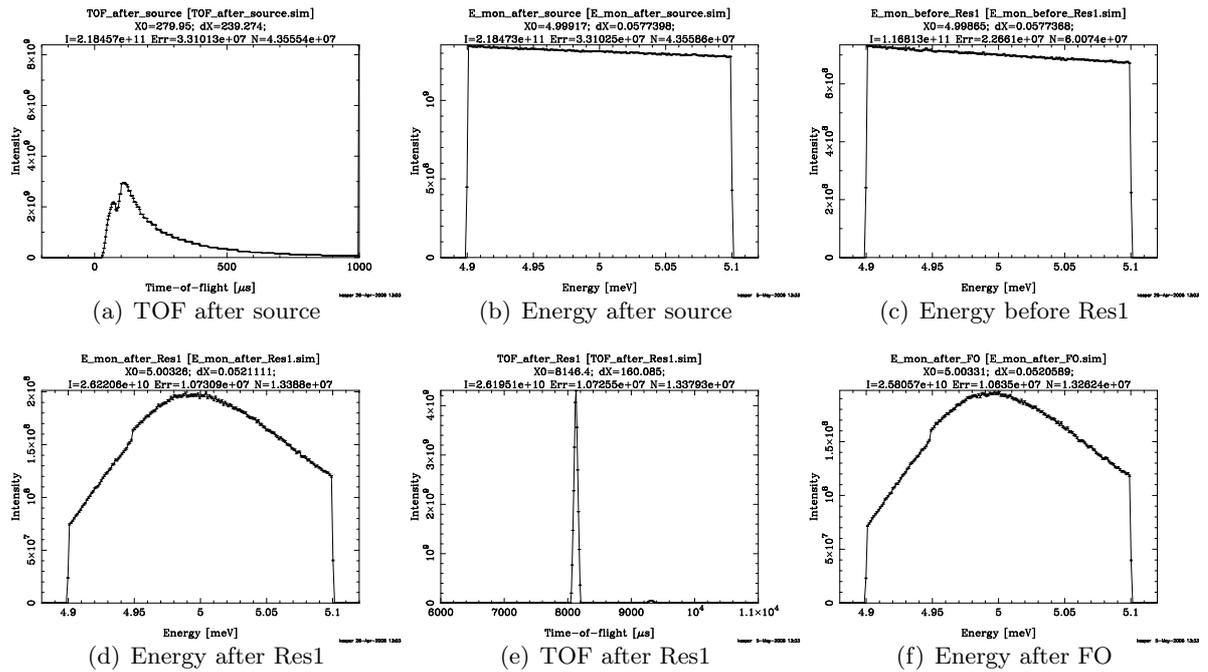
(a) Intensity of 5 meV neutrons at sample

Figure 6: Scan of phase offset

### 3.3 Normal source simulation

A simulation with no jitter and 120  $\mu$ seconds phase offset. Note the odd energy profile after Res1 (graph d), which is caused by the very uneven TOF distribution from the source, as seen in graph a.

Notice also the two side peaks in the last graph, explained in section 3.3.1



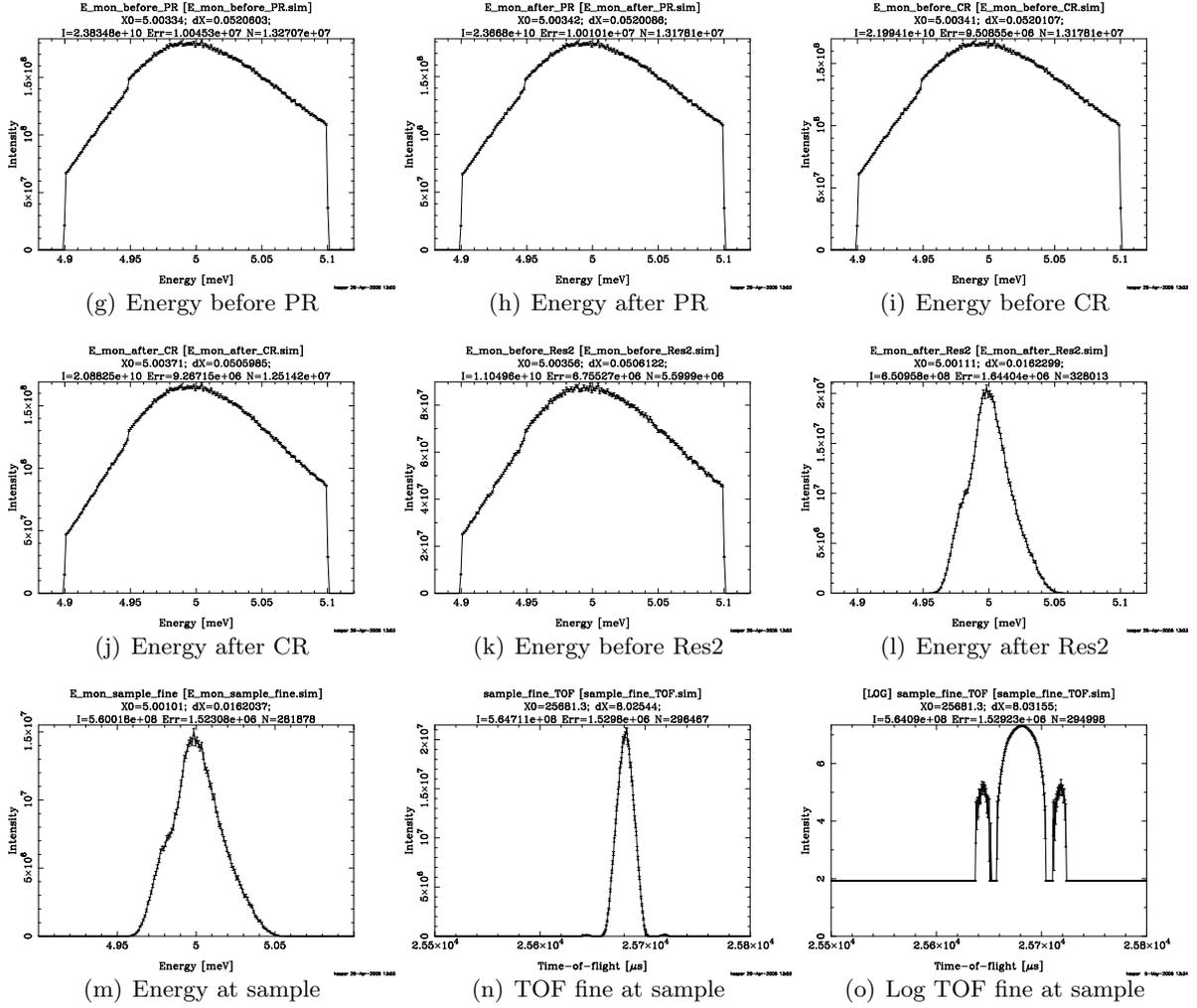


Figure 7: Normal simulation

### 3.3.1 Side peaks after Res2

The side peaks can be seen in the last graph of fig.7 (a log plot of the 2nd last graph), and can be seen to be about  $36 \mu$ s before and after the main peak. This offset in time, corresponds to an 11 mm spatial movement of the choppers, as the speed has been set to 140 Hz.

They were quite puzzling, but their explanation is rather simple: the chopper openings are triangular (the top part of those, which covers the guide openings, are of course trapezoidal) while the guide openings are rectangular. This allows a small wedge of the beam to be transmitted, when the chopper openings are aligned in the blank space between the two guides, which occurs when they have moved 11 mm.

This is all illustrated in the figure below:

The first graph is a snapshot taken by a PSD monitor, exactly at the time of the first side peak, which clearly shows the wedge profile. The two other graphs are x,t and y,t plots (with x being the horizontal axis perpendicular to the beam, and y being the vertical), showing the main peak and the two side peaks.

All three monitors are placed directly behind the Res2 choppers.

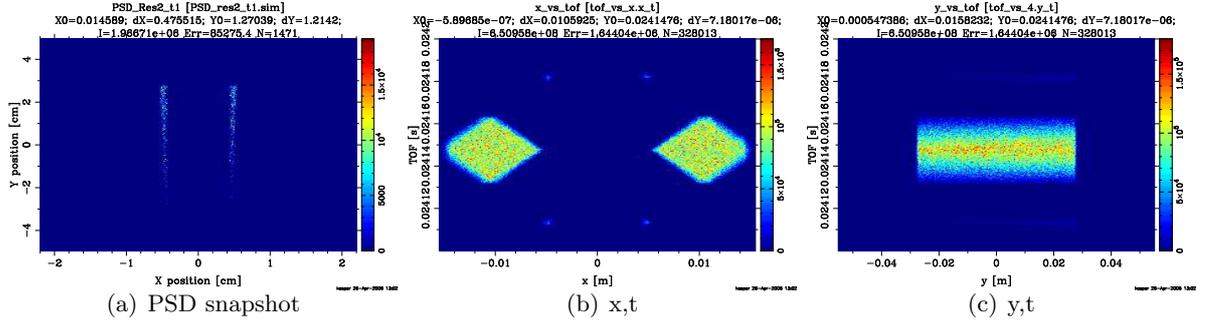


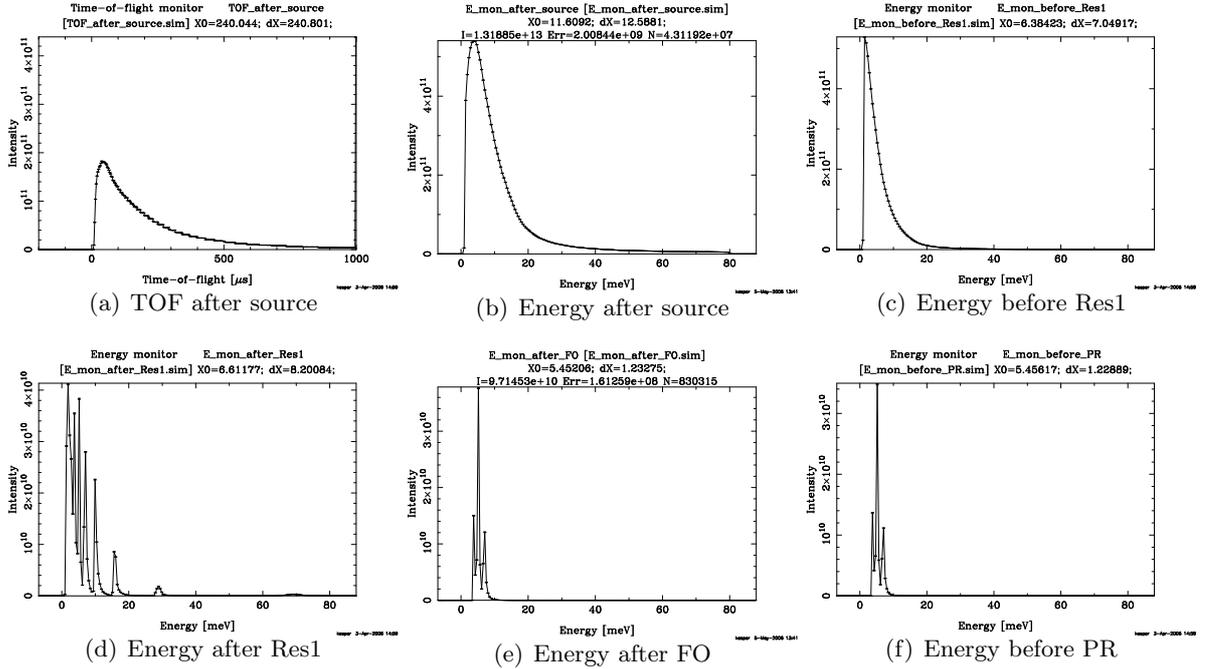
Figure 8: Side peaks after Res2

### 3.3.2 Broad energy range

Same as section 3.3, except the energy range is now 1-80 meV.

The simulation of the broader energy range is used to check whether the instruments lets through any spurions, as seen in fig. 3. As can be seen, this is not the case, and hence the narrow energy range will be used for most simulations, to save computing power.

Note how Res1 lets through multiple peaks, of which the subsequent choppers kills all but the desired, in accordance with the instrument design.



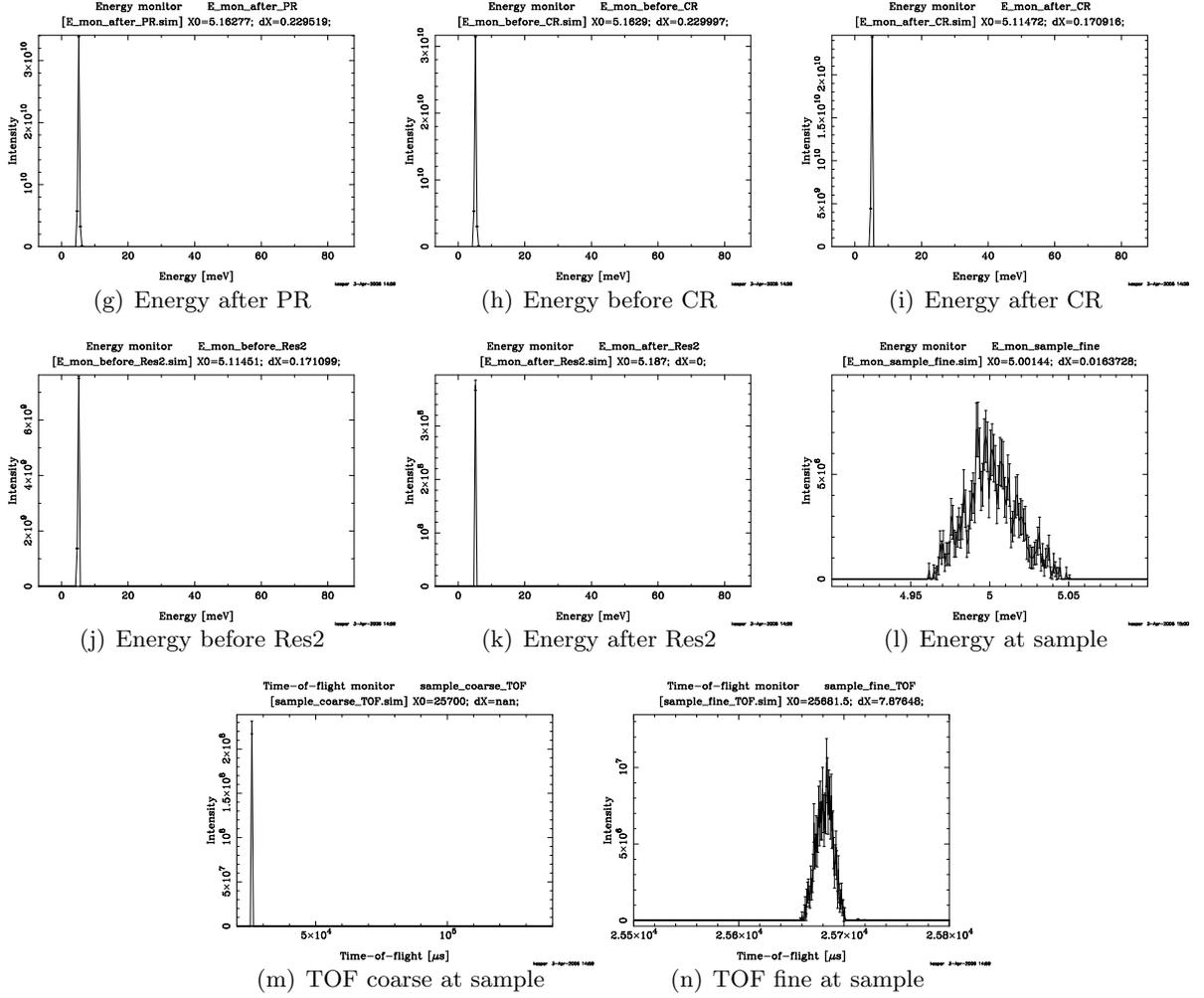
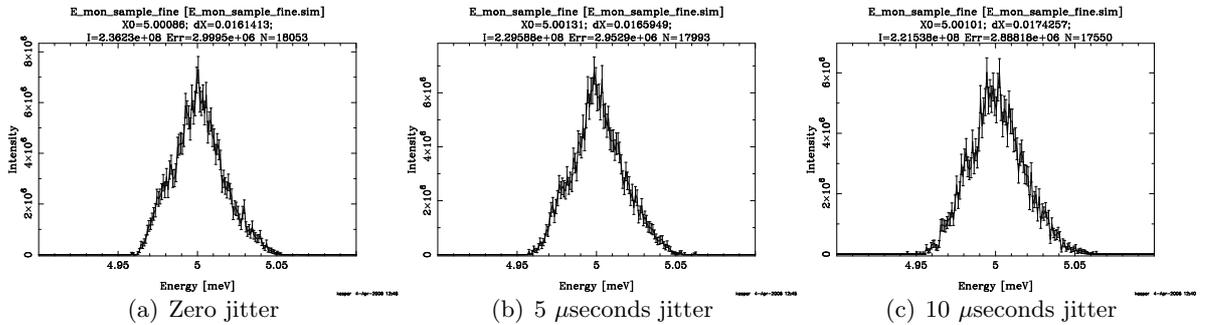


Figure 9: Energy range 1-80 meV

### 3.4 Jitter

Here the energy distribution at the sample position is seen for various values of jitter. Note that both intensity and resolution begins to seriously degrade at about 10-20  $\mu$ s of jitter. At 200  $\mu$ s the energy distribution is little more than noise.

Note also that the energy range has been broadened by a factor 2, in the last two simulations.



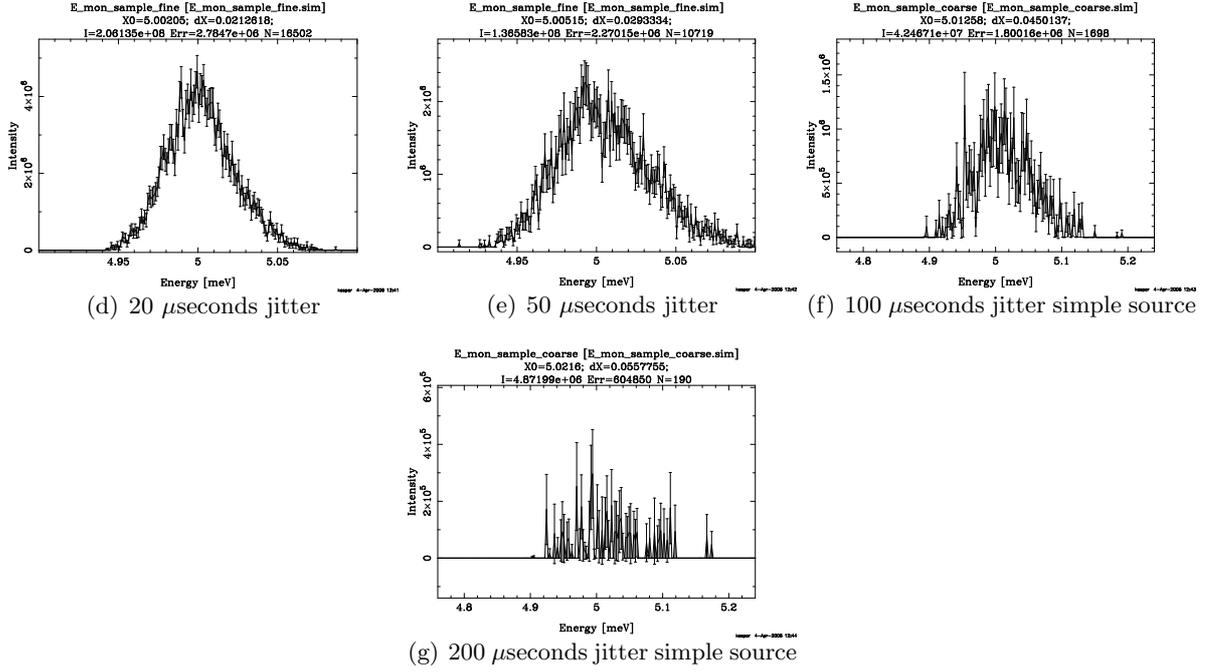


Figure 10: Energy distribution of the sample positin for different values of jitter.

Next a scan of jitter, showing the intensity of 5 meV neutrons at the sample position. As can be seen, jitter have negligible effect up to about 10  $\mu$ seconds, and then increases steadily until it becomes catastrophic at around 100  $\mu$ seconds. Note that the relative error on the first two graphs are not larger than that on the other graphs; the scale of the y-axis is simply different. Graph e was created using the simple source, as mentioned in section 3.1, and shows that the intensity drop off is more straight, when the tail of the moderator is absent.

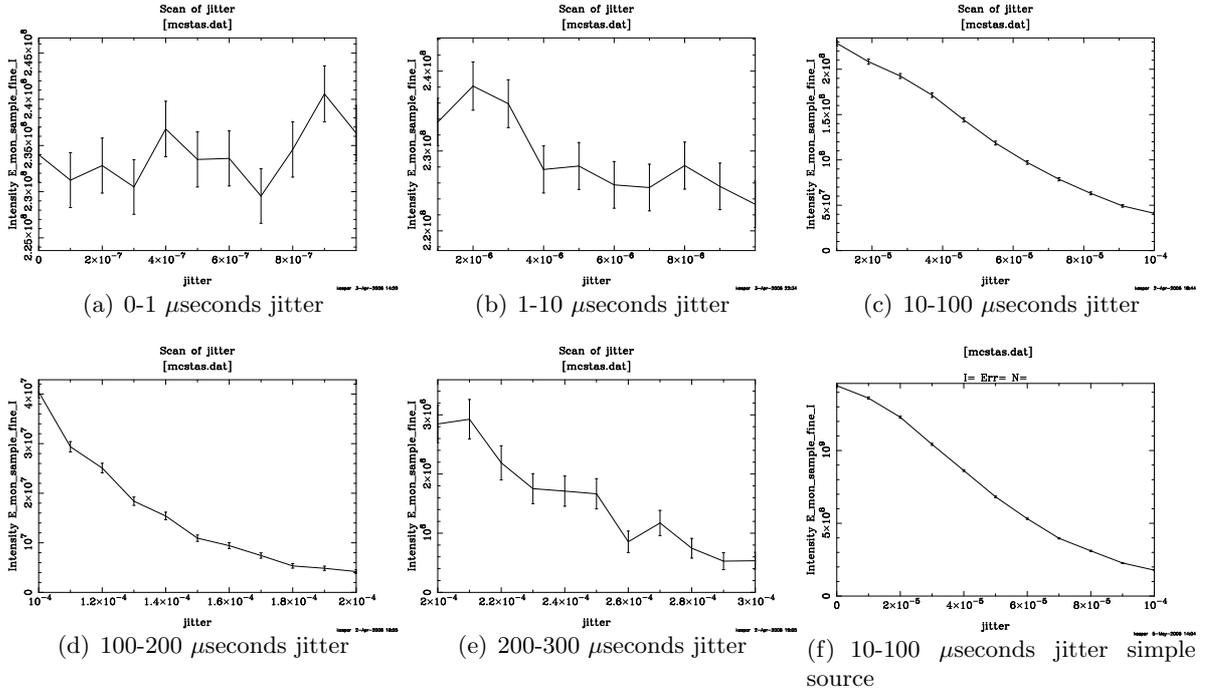
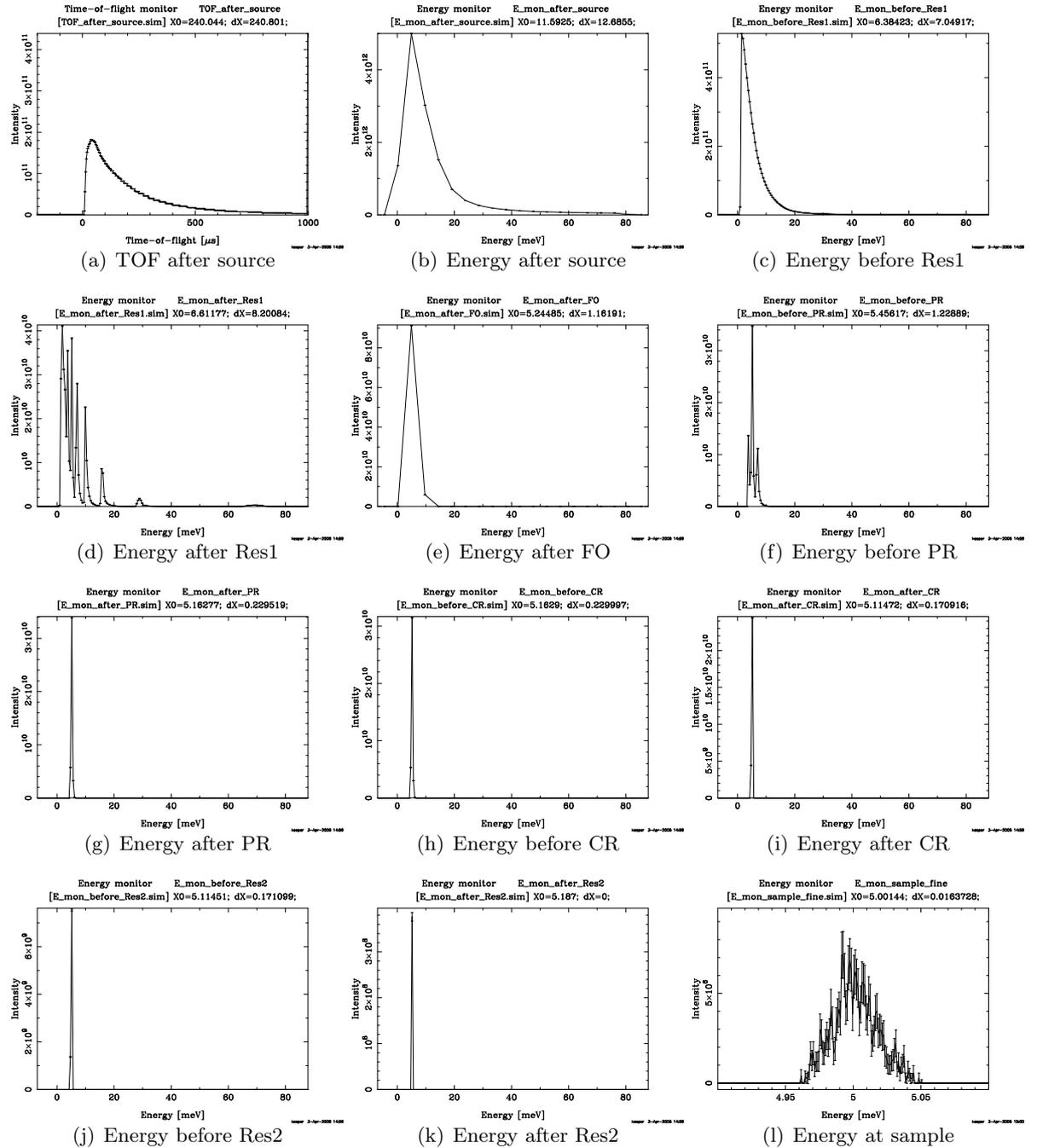


Figure 11: Scan of jitter

### 3.4.1 Jitter in the broad energy range

A simulation of jitter with a broad energy range, is needed to see whether the instrument will begin letting through spurions, before the jitter reaches a level where the resolution is seriously degraded anyway.

Again the energy range of 1-80 meV is used, this time with at jitter of 20  $\mu$ seconds. The only noticeable jitter effect, is a slight broadening of the distribution at the sample position, thus spurions does not appear at this level of jitter.



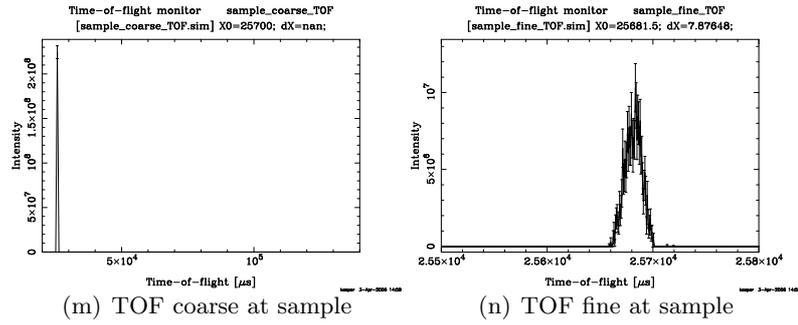


Figure 12: Energy range 1-80 meV

## 4 Conclusion

The program seems to fulfill its purpose, making realistic simulations of the effect of jitter in beam choppers.

On the basis of this, it seems fair to conclude that jitter at or below 10  $\mu$ seconds can be accepted. Above that, the effects of jitter steadily increases, resulting in degrading intensity and resolution, as seen in fig. 13.

## A Appendix

### A.1 Instrument code

```
DEFINE INSTRUMENT LET(g_w=0.04, g_h=0.09, jitter=0, E_min=4.9, E_max=5.1, Res=140, PR=70,  
E_foc=5, pha_offset=120)
```

```
DECLARE
```

```
{  
double v_foc, TOF_Res1, TOF_PR, TOF_CR, TOF_Res2, TOF_sample;  
}
```

```
INITIALIZE
```

```
{  
v_foc = SE2V*sqrt(E_foc);  
printf("\nv_foc: %f\n\n",v_foc);  
}
```

```
TRACE
```

```
COMPONENT Origin = Arm()  
  AT (0,0,0) ABSOLUTE
```

```
COMPONENT source = ISIS_moderator(  
  Face = "hydrogen", E0 = E_min, E1 = E_max, dist = 1.7, xw = g_w,  
  yh = g_h, modXsize = 0.1, modYsize = 0.1, CAngle = 0,SAC = 0)  
  AT (0, 0, 0) RELATIVE Origin /*EXTEND  
{  
t=0;  
}*/
```

```
COMPONENT TOF_after_source = TOF_monitor(  
  nchan = 400, filename = "TOF_after_source",xmin = -g_w*0.55,  
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, t0 = -200,  
t1 = 1000)  
  AT (0, 0, 1e-6) RELATIVE Origin
```

```
COMPONENT E_mon_after_source = E_monitor(  
  filename = "E_mon_after_source", xmin = -g_w*0.55,  
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_min-(E_max-E_min)/10,  
  Emax= E_max+(E_max-E_min)/10, nchan=200)  
  AT (0, 0, 2e-6) RELATIVE Origin
```

```
COMPONENT E_mon_before_guide = E_monitor(  
  filename = "E_mon_before_guide", xmin = -g_w*0.55,  
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_min-(E_max-E_min)/10,  
  Emax= E_max+(E_max-E_min)/10, nchan=200)  
  AT (0, 0, 1.7-1e-6) RELATIVE Origin
```

```

COMPONENT PSD_before_Guide1 = PSD_monitor(
  filename = "PSD_before_Guide1", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55)
AT (0, 0, 1.7-0.5e-6) RELATIVE Origin

COMPONENT Guide1 = Guide_channeled(
  w1 = g_w, h1 = g_h, w2 = g_w, h2 = g_h, l = 6.13-3e-6, alphax = 4.38, alphay = 4.38 ,
  W=3e-3, mx = 2, my = 3)
AT (0, 0, 1.7) RELATIVE Origin

COMPONENT PSD_after_Guide1 = PSD_monitor(
  filename = "PSD_after_Guide1", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55)
AT (0, 0, 7.83-3e-6) RELATIVE Origin

COMPONENT E_mon_before_Res1 = E_monitor(
  filename = "E_mon_before_Res1", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_min-(E_max-E_min)/10,
  Emax= E_max+(E_max-E_min)/10, nchan = 200)
AT (0, 0, 7.83-2e-6) RELATIVE Origin

COMPONENT TOF_before_Res1 = TOF_monitor(
  nchan = 400, filename = "TOF_before_Res1", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, t0 = 0.95*7.83/v_foc*1e6,
  t1 = 1.1*7.83/v_foc*1e6)
AT (0, 0, 7.83-1e-6) RELATIVE Origin

COMPONENT PSD_before_Res1 = PSD_monitor(
  filename = "PSD_before_Res1", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55)
AT (0, 0, 7.83-0.5e-6) RELATIVE Origin

COMPONENT Res1 = DiskChopper(theta_0=0,compat=1,abs_out=0,
  R = 0.305, omega = Res*2*PI, n=6, w=0.04004359436, j=jitter, t_0=7.83/v_foc+pha_offse
  AT (0, 0, 7.83) RELATIVE Origin

/*COMPONENT TOF_between_Res1 = TOF_monitor(
  nchan = 400, filename = "TOF_between_Res1", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, t0 = 7000, t1 = 9000)
AT (0, 0, 7.83+1e-6) RELATIVE Origin*/

COMPONENT Res1_counter = DiskChopper(theta_0=0,compat=0,abs_out=0,
  R = 0.305, omega = -Res*2*PI, n=6, w=0.04004359436, j=jitter, t_0=-7.83/v_foc-pha_off
  AT (0, 0, 7.83+2e-6) RELATIVE Origin

```

```

COMPONENT E_mon_after_Res1 = E_monitor(
  filename = "E_mon_after_Res1", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_min-(E_max-E_min)/10,
  Emax= E_max+(E_max-E_min)/10, nchan = 200)
  AT (0, 0, 7.83+3e-6) RELATIVE Origin

/*COMPONENT PSD_after_Res1 = PSD_monitor(
  filename = "PSD_after_Res1", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55)
  AT (0, 0, 7.83+4e-6) RELATIVE Origin*/

COMPONENT TOF_after_Res1 = TOF_monitor(
  nchan = 400, filename = "TOF_after_Res1",xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, t0 = 6000, t1 = 11000)
  AT (0, 0, 7.83+4e-6) RELATIVE Origin

COMPONENT Guide2 = Guide_channeled(
  w1 = g_w, h1 = g_h, w2 = g_w, h2 = g_h, l = 0.37-5e-6, alphax = 4.38, alphay = 4.38 ,
  W=3e-3, mx = 2, my = 3)
  AT (0, 0, 7.83+5e-6) RELATIVE Origin

COMPONENT FO = DiskChopper(theta_0=0,compat=0,abs_out=0,
  R = 0.305, omega = 10*2*PI, n=1, w=0.044937221097, j=jitter,
  t_0=8.2/v_foc+pha_offset*1e-6, IsFirst=0)
  AT (0, 0, 8.2) RELATIVE Origin

COMPONENT E_mon_after_FO = E_monitor(
  filename = "E_mon_after_FO", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_min-(E_max-E_min)/10,
  Emax= E_max+(E_max-E_min)/10, nchan=200)
  AT (0, 0, 8.2+1e-6) RELATIVE Origin

COMPONENT Guide3 = Guide_channeled(
  w1 = g_w, h1 = g_h, w2 = g_w, h2 = g_h, l = 3.55-5e-6, alphax = 4.38, alphay = 4.38 ,
  W=3e-3, mx = 2, my = 3)
  AT (0, 0, 8.2+2e-6) RELATIVE Origin

COMPONENT E_mon_before_PR = E_monitor(
  filename = "E_mon_before_PR", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_min-(E_max-E_min)/10,
  Emax= E_max+(E_max-E_min)/10, nchan = 200)
  AT (0, 0, 11.75-2e-6) RELATIVE Origin

COMPONENT TOF_before_PR = TOF_monitor(
  nchan = 400, filename = "TOF_before_PR",xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, t0 = 0.95*11.75/v_foc*1e6,

```

```

t1 = 1.05*11.75/v_foc*1e6)
  AT (0, 0, 11.75-1e-6) RELATIVE Origin

COMPONENT PR = DiskChopper(theta_0=0,compat=0,abs_out=0,
  R = 0.35, omega = PR*2*PI, n=2, w=0.07427690045, j=jitter, t_0=11.75/v_foc+pha_offset
  AT (0, 0, 11.75) RELATIVE Origin

COMPONENT E_mon_after_PR = E_monitor(
  filename = "E_mon_after_PR", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_min-(E_max-E_min)/10,
  Emax= E_max+(E_max-E_min)/10, nchan = 200)
  AT (0, 0, 11.75+1e-6) RELATIVE Origin

COMPONENT Guide4 = Guide_channeled(
  w1 = g_w, h1 = g_h, w2 = g_w, h2 = g_h, l = 3.91-5e-6, alphax = 4.38, alphay = 4.38 ,
  W=3e-3, mx = 2, my = 3)
  AT (0, 0, 11.75+2e-6) RELATIVE Origin

COMPONENT TOF_before_CR = TOF_monitor(
  nchan = 400, filename = "TOF_before_CR",xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, t0 = 0.95*15.66/v_foc*1e6,
t1 = 1.05*15.66/v_foc*1e6)
  AT (0, 0, 15.66-2e-6) RELATIVE Origin

COMPONENT E_mon_before_CR = E_monitor(
  filename = "E_mon_before_CR", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_min-(E_max-E_min)/10,
  Emax= E_max+(E_max-E_min)/10, nchan = 200)
  AT (0, 0, 15.66-1e-6) RELATIVE Origin

COMPONENT CR = DiskChopper(theta_0=0,compat=0,abs_out=0,
  R = 0.35, omega = Res*PI, n=6, w=0.05511346046, j=jitter, t_0=15.66/v_foc+pha_offset*
  AT (0, 0, 15.66) RELATIVE Origin

COMPONENT E_mon_after_CR = E_monitor(
  filename = "E_mon_after_CR", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_min-(E_max-E_min)/10,
  Emax= E_max+(E_max-E_min)/10, nchan = 200)
  AT (0, 0, 15.66+1e-6) RELATIVE Origin

COMPONENT Guide5 = Guide_channeled(
  w1 = g_w, h1 = g_h, w2 = g_w, h2 = 0.06, l = 6.715-4e-6, alphax = 4.38, alphay = 4.38
  W=3e-3, mx = 2, my = 3)
  AT (0, 0, 15.66+2e-6) RELATIVE Origin

/*COMPONENT PSD_Guide5 = PSD_monitor(
  nx=128, ny=128, filename="PSD_Guide5",

```

```

xmin = -g_w*0.55, xmax = g_w*0.55,
ymin = -g_h*0.55, ymax = g_h*0.55)
  AT (0, 0, 22.375-1e-6) RELATIVE Origin*/

COMPONENT E_mon_before_split = E_monitor(
  filename = "E_mon_before_split", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_min-(E_max-E_min)/10,
  Emax= E_max+(E_max-E_min)/10, nchan = 200)
  AT (0, 0, 22.375-1e-6) RELATIVE Origin

COMPONENT splitguide_211 = Guide_channeled(
  w1 = 0.019, h1 = 0.06, w2 = 0.01, h2 = 0.05512, l = 1.1, alphax = 4.38, alphay = 4.38
W=3e-3, mx = 2, my = 3)
  AT (0.00975, 0, 22.375) RELATIVE Origin
  GROUP double_funnel_21

COMPONENT splitguide_212 = Guide_channeled(
  w1 = 0.019, h1 = 0.06, w2 = 0.01, h2 = 0.05512, l = 1.1, alphax = 4.38, alphay = 4.38
W=3e-3, mx = 2, my = 3)
  AT (-0.00975, 0, 22.375) RELATIVE Origin
  GROUP double_funnel_21

COMPONENT TOF_before_Res2 = TOF_monitor(
  nchan = 400, filename = "TOF_before_Res2", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, t0 = 0.95*23.5/v_foc*1e6,
t1 = 1.05*23.5/v_foc*1e6)
  AT (0, 0, 23.5-2e-6) RELATIVE Origin

COMPONENT E_mon_before_Res2 = E_monitor(
  filename = "E_mon_before_Res2", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_min-(E_max-E_min)/10,
  Emax= E_max+(E_max-E_min)/10, nchan = 200)
  AT (0, 0, 23.5-1e-6) RELATIVE Origin

/*COMPONENT PSD_split1= PSD_monitor(
nx=128, ny=128, filename="PSD_split1",
xmin = -g_w*0.55, xmax = g_w*0.55,
ymin = -g_h*0.55, ymax = g_h*0.55)
  AT (0, 0, 23.5-2e-7) RELATIVE Origin*/

COMPONENT Res2 = DiskChopper(theta_0=0,compat=0,abs_out=0,
  R = 0.35, omega = Res*2*PI, n=2, w=0.031, wc=0.011, j=jitter, t_0=23.5/v_foc+pha_offs
  AT (0, 0, 23.5) RELATIVE Origin

COMPONENT PSD_Res2_t0 = PSD_monitor_t0(
  nx=256, ny=256, filename="PSD_res2_t0",

```

```

xmin = -g_w*0.55, xmax = g_w*0.55,
ymin = -g_h*0.55, ymax = g_h*0.55, tmin=2.4105e4, tmax=2.412e4)
  AT (0, 0, 23.5+2e-7) RELATIVE Origin

COMPONENT TOF_betweeen_Res2 = TOF_monitor(
  nchan = 3200, filename = "TOF_betweeen_Res2", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, t0 = 24000, t1 = 24300)
  AT (0, 0, 23.5+5e-7) RELATIVE Origin

COMPONENT Res2_counter = DiskChopper(theta_0=0, compat=0, abs_out=0,
  R = 0.35, omega = -Res*2*PI, n=2, w=0.031, wc=0.011, j=jitter, t_0=-23.5/v_foc-pha_of
  AT (0, 0, 23.5+1e-6) RELATIVE Origin

COMPONENT PSD_Res2_t1 = PSD_monitor_t0(
  nx=256, ny=256, filename="PSD_res2_t1",
  xmin = -g_w*0.55, xmax = g_w*0.55,
  ymin = -g_h*0.55, ymax = g_h*0.55, tmin=2.4105e4, tmax=2.412e4)
  AT (0, 0, 23.5+12e-7) RELATIVE Origin

COMPONENT x_vs_tof = Monitor_nD(
  options = "x limits [-0.0155,0.0155], t limits [0.0241 0.0242]",
  filename = "tof_vs_x", xwidth = 0.031, yheight = g_h,
  bins = 256)
  AT (0, 0, 23.5+15e-7) RELATIVE Origin

COMPONENT y_vs_tof = Monitor_nD(
  options = "y limits [-0.05512/2,0.05512/2], t limits [0.0241 0.0242]",
  filename = "tof_vs_4", xwidth = 0.031, yheight = g_h,
  bins = 256)
  AT (0, 0, 23.5+16e-7) RELATIVE Origin

COMPONENT E_mon_after_Res2 = E_monitor(
  filename = "E_mon_after_Res2", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_min-(E_max-E_min)/10,
  Emax= E_max+(E_max-E_min)/10, nchan = 200)
  AT (0, 0, 23.5+2e-6) RELATIVE Origin

COMPONENT splitguide_221 = Guide_channeled(
  w2 = 0.019, h2 = 0.05, w1 = 0.011, h1 = 0.05488, l = 1.1, alphax = 4.38, alphay = 4.3
  W=3e-3, mx = 2, my = 3)
  AT (0.00975, 0, 23.525) RELATIVE Origin
  GROUP double_funnel_22

COMPONENT splitguide_222 = Guide_channeled(
  w2 = 0.019, h2 = 0.05, w1 = 0.011, h1 = 0.05488, l = 1.1, alphax = 4.38, alphay = 4.3
  W=3e-3, mx = 2, my = 3)
  AT (-0.00975, 0, 23.525) RELATIVE Origin

```

```

GROUP double_funnel_22

COMPONENT E_mon_after_guide = E_monitor(
  filename = "E_mon_after_guide", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_min-(E_max-E_min)/10,
  Emax= E_max+(E_max-E_min)/10, nchan=200)
  AT (0, 0, 24.625+1e-6) RELATIVE Origin

COMPONENT sample_coarse_TOF = TOF_monitor(
  nchan = 200, filename = "sample_coarse_TOF", xmin = -0.1, xmax = 0.1,
  ymin = -0.1, ymax = 0.1, t0 = 20000, t1 = 140000)
  AT (0, 0, 25-2e-6) RELATIVE Origin

COMPONENT sample_fine_TOF = TOF_monitor(
  nchan = 400, filename = "sample_fine_TOF", xmin = -0.1, xmax = 0.1,
  ymin = -0.1, ymax = 0.1,
  //t0 = 0.95*25/v_foc*1000000, t1 = 1.05*25/v_foc*1000000
  t0 = 2.55e4, t1 = 2.58e4)
  AT (0, 0, 25-1e-6) RELATIVE Origin

COMPONENT E_mon_sample_coarse = E_monitor(
  filename = "E_mon_sample_coarse", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_min-(E_max-E_min)/10,
  Emax= E_max+(E_max-E_min)/10, nchan = 200)
  AT (0, 0, 25) RELATIVE Origin

COMPONENT E_mon_sample_fine = E_monitor(
  filename = "E_mon_sample_fine", xmin = -g_w*0.55,
  xmax = g_w*0.55, ymin = -g_h*0.55, ymax = g_h*0.55, Emin = E_foc-0.1,
  Emax= E_foc+0.1, nchan = 200)
  AT (0, 0, 25+1e-6) RELATIVE Origin

END

```

## A.2 Chopper code

```
/******  
*  
* McStas, neutron ray-tracing package  
*   Copyright 1997-2006, All rights reserved  
*   Risoe National Laboratory, Roskilde, Denmark  
*   Institut Laue Langevin, Grenoble, France  
*  
* Component: DiskChopper  
*  
* %I  
* Written by: Peter Willendrup  
* Date: March 9 2006  
* Version: $Revision: 1.6 $  
* Origin: Risoe  
* Release: McStas 1.9.1  
* Based on Chopper (Philipp Bernhardt), Jitter and beamstop from work by  
* Kaspar Hewitt Klenoe (jan 2006), adjustments by Rob Bewey (march 2006)  
*  
*  
* %D  
* Models a disc chopper with n identical slits, which are symmetrically disposed  
* on the disc. Possibly, the slits can have central absorbing inserts ('beamstops').  
* For more complicated geometries, see component manual example of DiskChopper GROUPing.  
*  
* If the chopper is the 1st chopper of the instrument, it sets t time with phase  
*  
* Example: DiskChopper(R=0.2, theta_0=10, f=2500.0, n=3, t_0=0, IsFirst=1) First chopper  
*          DiskChopper(R=0.2, theta_0=10, f=2500.0, n=3, t_0=0, IsFirst=0)  
*  
* %P  
* INPUT PARAMETERS:  
*  
* theta_0: (deg)   Angular width of the slits.  
* h:        (m)    Slit height (if = 0, equal to R). Auto centering of beam at h/2.  
* R:        (m)    Radius of the disc  
* omega:    (rad/s) Angular frequency of the Chopper  
*           (algebraic sign defines the direction of rotation)  
* n:        (1)    Number of slits  
*  
* Optional parameters:  
* IsFirst: (0/1)   Set it to 1 for the first chopper position in a cw source  
*               (it then spreads the neutron time distribution)  
* n_pulse: (1)    Number of pulses (Only if IsFirst)  
* j        (s)    Jitter in the phase  
* theta_1: (deg)   Angular width of optional beamstop in chopper windows  
* abs_out: (0/1)   Absorb neutrons hitting outside of chopper radius?
```

```

* t_0:      (s)      Time 'delay'
* phi_0:    (deg)    Angular 'delay' (suppresses t_0)
* w:        (m)      'width' of slits for compatibility with Chopper.comp
* wc:       (m)      'width' of beamstops for compatibility with Chopper.comp
* compat:   (1)      Chopper placement compatible with original Chopper.comp
*
* %D
* Example values: theta_0=10 h=0.2 R=0.5 f=2500 n=3 pha=0 n_pulse=1 j=0 wc=0
*
* %E
*****/

DEFINE COMPONENT DiskChopper
DEFINITION PARAMETERS ()
SETTING PARAMETERS (theta_0, R, h=0, omega, n=3, j=0, theta_1=0, t_0=0, IsFirst=0, n_puls
OUTPUT PARAMETERS (Tg, T, To, Tc, delta_y)
STATE PARAMETERS (x, y, z, vx, vy, vz, t, s1, s2, p)

DECLARE
%{
  double Tg,To,Tc,delta_y;
%}

INITIALIZE
%{
  /* Compatibility with old Chopper.comp interface */
  if (w) {
theta_0=2*RAD2DEG*asin((w/2)/R);
  }
  if (wc) {
theta_1=2*RAD2DEG*asin((wc/2)/R);
  }
  if (compat) {
delta_y = R;
  } else {
delta_y = R-h/2;
  }

  if (n<=0 || theta_0 < 0 || R <=0)
  { fprintf(stderr,"DiskChopper: %s: n, theta_0 and R must be > 0\n", NAME_CURRENT_CO
    exit(-1); }
  if (n*theta_0 >= 360)
  { fprintf(stderr,"DiskChopper: %s: n * theta_0 exceeds 2PI\n", NAME_CURRENT_COMP);
    exit(-1); }
  if (theta_1 >= theta_0)
  { fprintf(stderr,"DiskChopper: %s: theta_1 must me lower than theta_0\n", NAME_CURR
    exit(-1); }

```

```

        if (h && h>R) {
fprintf(stderr,"DiskChopper: %s: h must be < R\n", NAME_CURRENT_COMP);
        exit(-1); }
        if (IsFirst && n_pulse <=0)
        { fprintf(stderr,"DiskChopper: %s: wrong First chopper pulse number (n_pulse)\n", NAME_CURRENT_COMP);
          exit(-1); }
        if (!omega) {
fprintf(stderr,"DiskChopper: %s WARNING: chopper frequency is 0!\n", NAME_CURRENT_COMP);
omega = 1e-15; /* We should actually use machine epsilon here... */
        }
        if (!(abs_out==1)) {
fprintf(stderr,"DiskChopper: %s WARNING: chopper will NOT absorb neutrons outside radius
abs_out=0;
        }
        if (IsFirst && theta_1) {
fprintf(stderr,"DiskChopper: %s WARNING: Options IsFirst and beamstops not compatible. Beamstop
theta_1=0;
        }

        theta_0*=DEG2RAD;
        theta_1*=DEG2RAD;

        /* If slit height 'unset', assume full opening */
        if (h == 0) {
h=R;
        }

        /* Calculate t_0 from phi_0 and vice versa, 'direction' moderated by sign of omega */
        t_0 *=omega/fabs(omega);
        if (phi_0) {
if (t_0) {
        fprintf(stderr,"DiskChopper: %s WARNING: t_0 AND phi_0 specified. Using phi_0 setting\n", NAME_CURRENT_COMP);
        }
        phi_0*=DEG2RAD;
        /* 'Delay' should always be a delay, taking rotation direction into account: */
        t_0=omega*phi_0/(omega*omega);
        } else {
phi_0=t_0*omega;
        }

        /* Time from opening of slit to next opening of slit */
        Tg=2*PI/fabs(omega)/n;

        /* How long can neutrons pass the Chopper at a single point */
        To=theta_0/fabs(omega);

        /* How long are neutrons absorbed by beamstop in chopper windows? */

```

```

        Tc=theta_1/fabs(omega);
    %}

TRACE
    %{
        double toff;
        double yprime;

        yprime = y+delta_y;
        PROP_Z0;

        /* Is neutron outside the vertical slit range and should we absorb? */
        if ((x*x+yprime*yprime)>R*R && abs_out==1) {
            ABSORB;
        }
        /* Does neutron hit inner solid part of chopper in case of h!=R? */
        if ((x*x+yprime*yprime)<(R-h)*(R-h)) {
            ABSORB;
        }

        if (IsFirst)
        {
            t=atan2(x,yprime)/omega + To*randpm1()/2.0 - t_0 + j*randnorm() + floor(n_pulse*rand01())
            p*=n*theta_0/2/PI;
        }
        else
        {
            toff=fabs(t-atan2(x,yprime)/omega - t_0 + j*randnorm());
            /* does neutron hit beamstop? */
            if (fmod(toff+Tc/2.0,Tg)<Tc) ABSORB;
            /* does neutron hit outside slit? */
            if (fmod(toff+To/2.0,Tg)>To) ABSORB;
        }
        SCATTER;

    %}

MCDISPLAY
    %{

        int j,k,l,kmin,kmax;
        /* Arrays for storing geometry of slit/beamstop */
        double X[5];
        double Y[5];
        double thetas[4];
        double Radius;

```

```

double theta;
magnify("xy");
circle("xy", 0, -delta_y, 0, R);
/* Drawing the slit(s)/internal beamstop(s): */

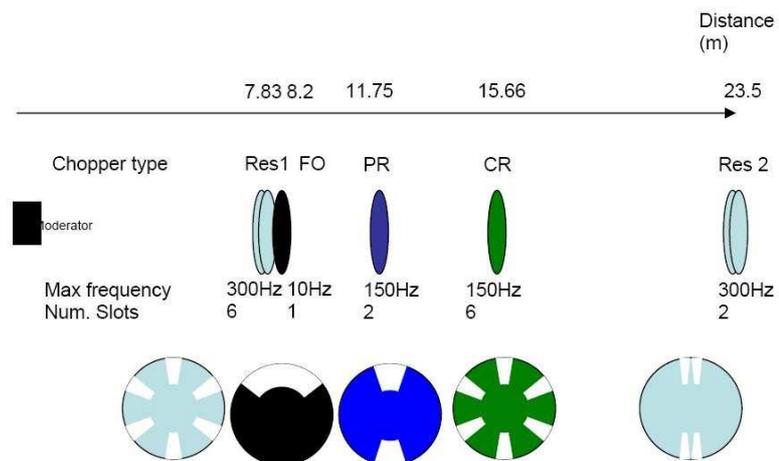
for (j=0; j<n; j++) {
  /* Angular start/end of slit */
  thetas[0] = j*(theta_0+(Tg-To)*omega) - theta_0/2 + phi_0;
  thetas[3] = j*(theta_0+(Tg-To)*omega) + theta_0/2 + phi_0;
  /* Angular start/end of beamstop (possibly identical) */
  thetas[1] = thetas[0] + (theta_0 - theta_1)/2;
  thetas[2] = thetas[1] + theta_1;
  /* Draw two 'multilines' from start of slit to start of beamstop
     and end of beamstop to end of slit. */
  for (l=0; l<2; l++) {
    for (k=0; k<5; k++) {
if (k==0) {
  theta = thetas[2*l];
  if (l==1 && (!theta_1)) {
    Radius = R-h;
  } else {
    Radius = R;
  }
} else if (k==4) {
  theta = thetas[2*l+1];
  if (l==0 && (!theta_1)) {
    Radius = R-h;
  } else {
    Radius = R;
  }
} else {
  theta = thetas[2*l]+(k-1)*(thetas[2*l+1]-thetas[2*l])/2;
  Radius = R-h;
}
X[k] = Radius * sin(theta);
Y[k] = Radius * cos(theta)-delta_y;
    }
    multiline(5,
(double)X[0], (double)Y[0], (double)0,
(double)X[1], (double)Y[1], (double)0,
(double)X[2], (double)Y[2], (double)0,
(double)X[3], (double)Y[3], (double)0,
(double)X[4], (double)Y[4], (double)0);
    }
  }
}
END

```

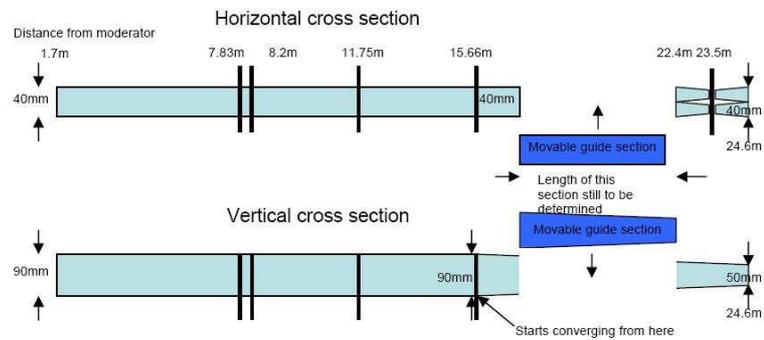
## References

- [1] Mcstas documentation is available at the homepage: <http://www.mcstas.org>
- [2] LET homepage: <http://www.isis.rl.ac.uk/targetstation2/instruments/let/>
- [3] R. Bewley (ISIS), LET specification, report from Jan. 2006
- [4] R. Bewley (ISIS), Multi-rep rate on LET version 3, report from 14th september 2005
- [5] P. Willendrup, U. Filges, L. Keller, E. Farhi, K. Lefmann: Validation of a realistic powder sample using data from DMC at PSI, accepted for Physica B, 2000
- [6] R. Bewley (ISIS): Simulations of the double funnel construction for LET. Comparison with a single funnel, report from 2005

LETs Chopper arrangement 15<sup>th</sup> September 05



LETs guide arrangement 15<sup>th</sup> September 05



Notes on guide: The horizontal cross section is 40mm the whole way apart from at the double funnel section. There will be small gaps in the guide at the chopper positions which are marked on. These gaps are to be kept to a minimum and are to be determined in co-operation with chopper manufacturer. The vertical guide is 90mm high until it reaches the chopper at 15.66m, where it linearly converges to 50mm at 24.6m. The length of the movable guide section is still to be determined.

Figure 2: LET chopper and guide layout[3]

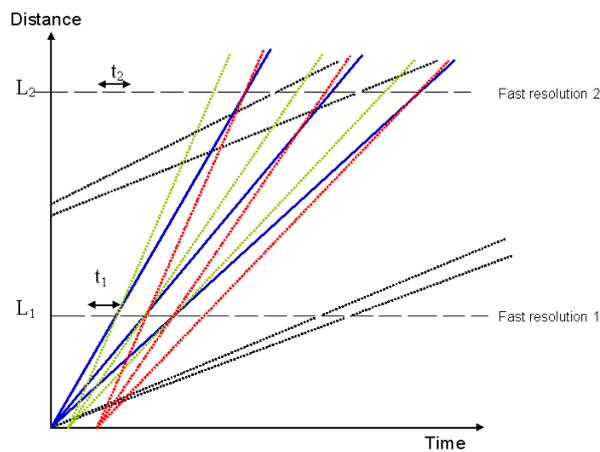


Figure 3: The desired pulses (in blue) and spurions (red, green and black), passing through LET with only the resolution choppers present[4]

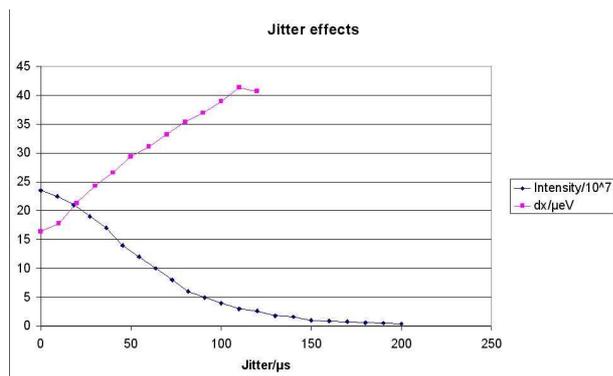


Figure 13: A plot of intensity and resolution, as a function of increased jitter.