## Calibration software for the LKr

### Michal Zamkovsky

Charles University in Prague

02.09.2014

Michal Zamkovsky (Charles University in Pra NA62 Collaboration meeting in Ferrara

- Migrate the NA48 software (FORTRAN) to NA62 (C++)
- 2 Have it operational for the 2014 technical run
- Get the NA48 software running for checking and backup
- Obtain same results for 2012 SLM data
- Build a digital filter

# Strategy

Select 4 of 8 samplings and fit three parameters of Double Fermi-Dirac function:

$$f(x)=a\left(rac{1}{e^{rac{-t+b}{g}}+1}+rac{f}{e^{rac{t-b-e}{d}}+1}-h
ight)+c$$

- Obtain the time value at pulse peak
- Ilign several pulses in time and fit again with more parameters
- Propagate these parameters to fit individual pulses again iteratively
- Soop over several DAC values and all patterns
- Compute the pulse width for each channel, which is the shape parameter used to group all channels into a few (order 10) groups, where each group will have its own digital filter constants

## Time aligned pulses



02.09.2014 4 / 8

# Time aligned pulses with Double Fermi-Dirac function fit





- As for NA48 we should (eventually) separate:
  - read all calibration bursts to accumulate pulses in an intermediate store
- analyze the pulses in different ways for DFD fitting and Digital filtering
  With DFD fitting we can already check bad channels and linearity, beginning with a few DACs and then covering the full range with about 12 DACs

### Status

#### What is done

- OFD local and global pulse fitting
- Production version to read a full calibration cycle (several DAC values)
- 3 The new software is so far checked with 2012 data

### What has to be done

- Add all 8 patterns to production version
- Ocompute the pulse width for each channel and make groups of digital filter constants
- Independently migrate the pedestal measurement and monitoring programme PERK. Alan will first get a simplified version running in the NA48 framework, for migration to NA62. We will need pedestal runs from a pseudo-random trigger
- Alan will resurrect Unal's logic to compute digital filter constants, using PUCK output and the channel width classification
- The various constants should be stored in a data base so they are available for comparison and monitoring, reconstruction and TSL

Thank you for your attention!