

Glueball working group meeting
18 February 2003, Hamburg

Small angle scintillating counters

R. Mizuk
ITEP, Moscow

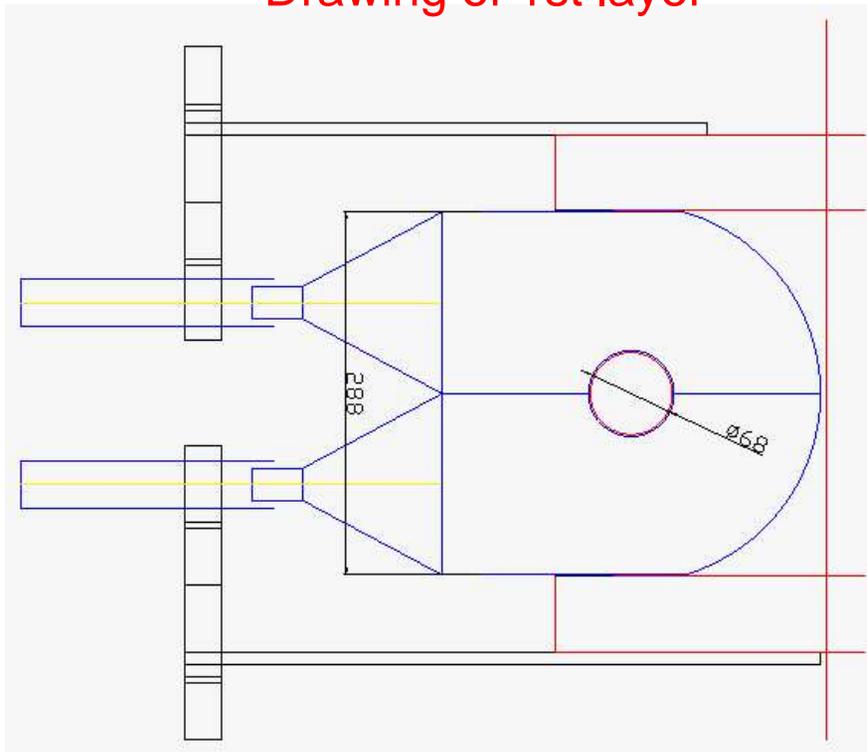
Design

Performance

Design of scintillating counters

∃ 4 scintillating counters which form two superlayers

Drawing of 1st layer



- in front of mu absorber →
- 10mm scintillator
- 10mm Pb in front ($2X_0$)
- ECAL PMTs, bases, readout,..

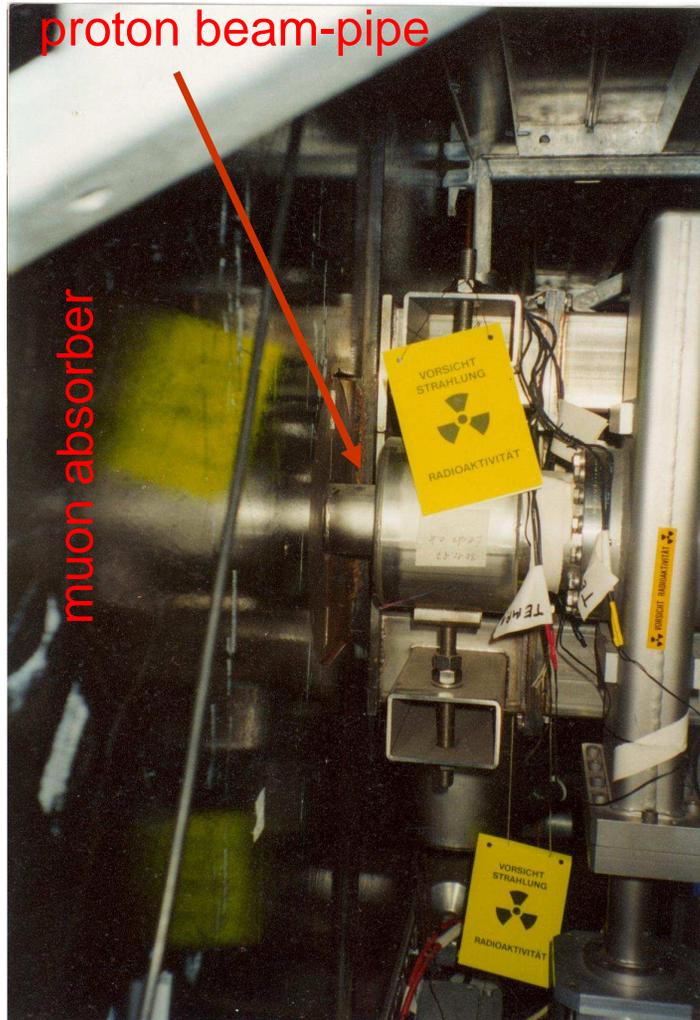
SVD: $\eta \in (1.5, 5.)$

ECAL: $\eta \in (2.5, 5.5)$

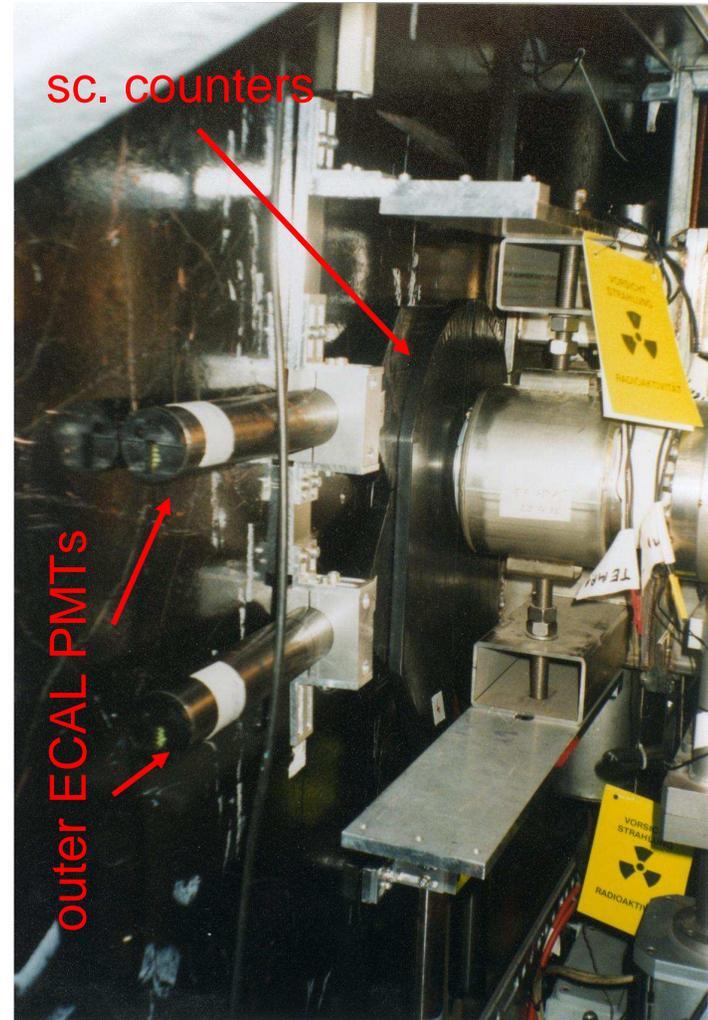
Sc. counters: $\eta \in (5.3, 6.8)$

Location of scintillating counters

Before installation of sc. counters

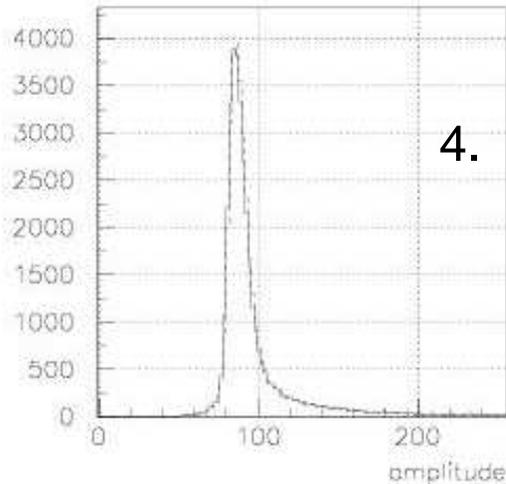
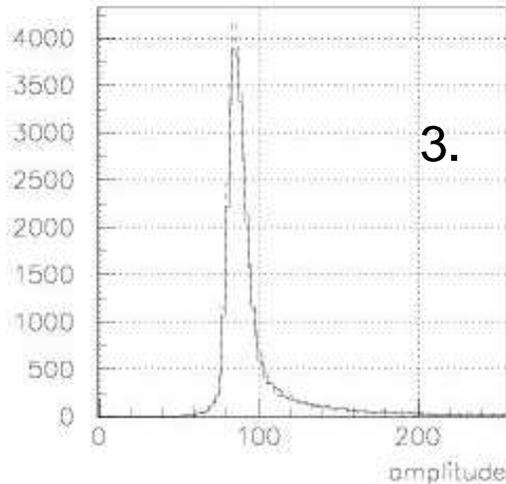
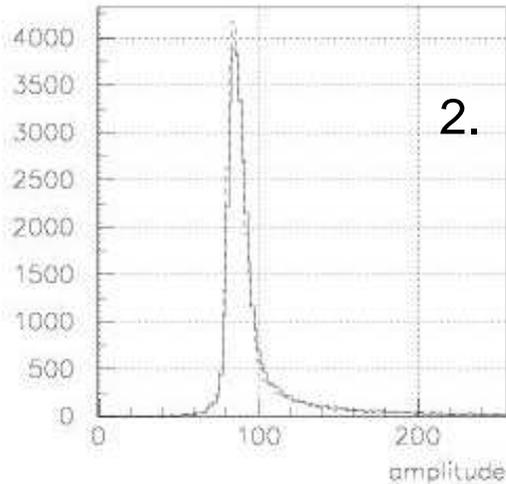
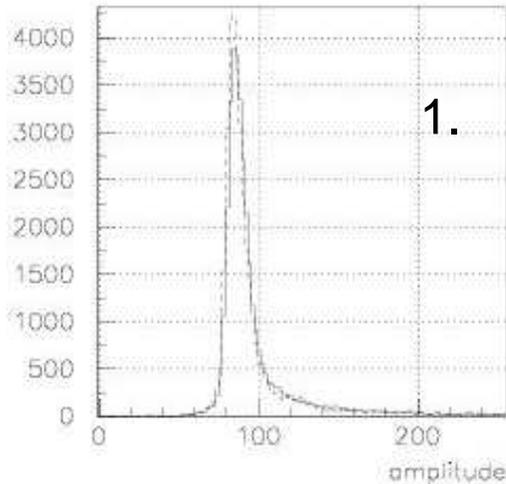


After installation of sc. counters



Stability of scintillating counters

Amplitude of counter #3 for all Glueball runs



runs 21019,21,22,23 (b1)
(solid hist)

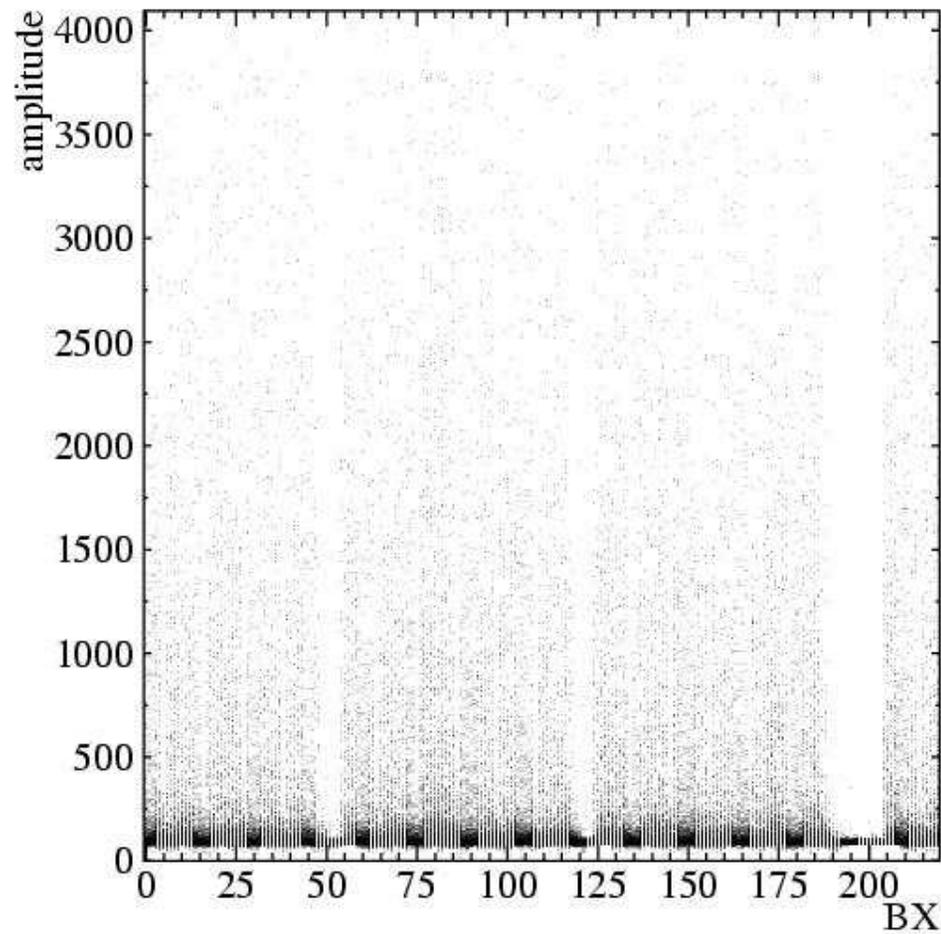
vs.

1. run 21018 (i2)
2. runs 21036,37 (b1)
3. run 21043 (b1)
4. runs 21111,12,13 (b1)
(dashed hist)

sc. counters were
working very stable

Signals from scintillating counters

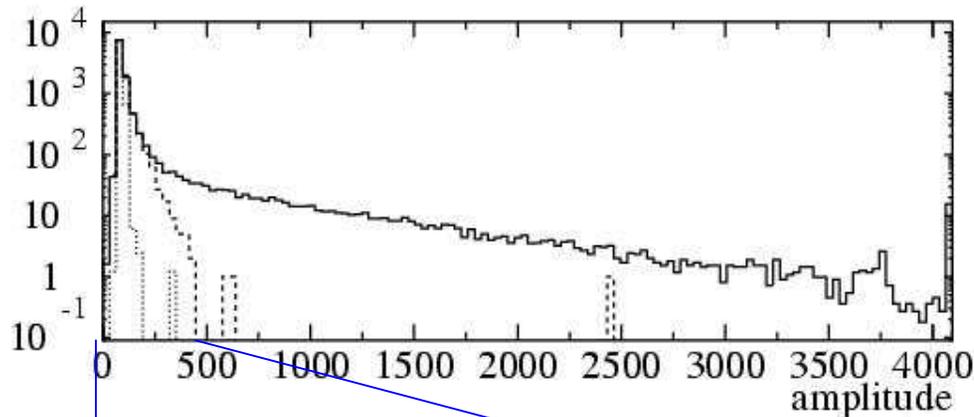
Amplitude vs. BX number for random trigger events in Glueball runs



Bunch structure
is clearly visible

Background

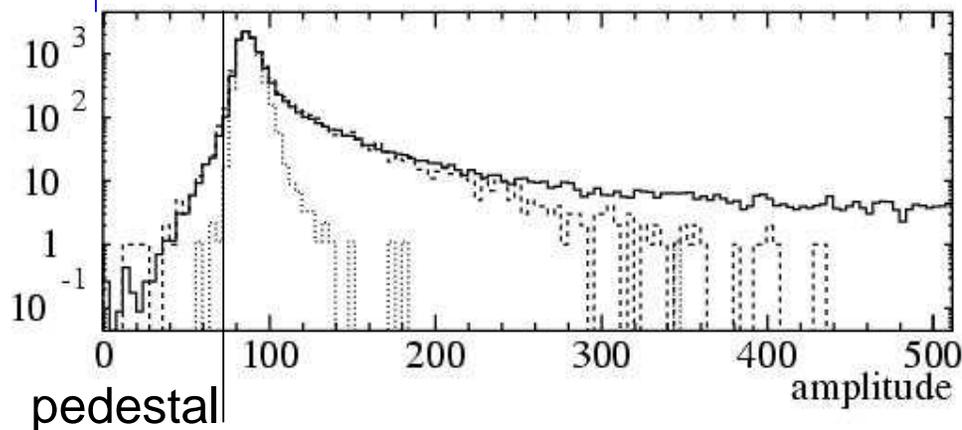
Amplitude averaged over all bunches (solid hist),
for 1st empty bunche (dashed) and for ≥ 4 subsequent empty bunches (dotted)



there is no tail
in empty bunches:
 $a_{3>270}$
0.5% for 1st empty bunch
8.7% for all bunches

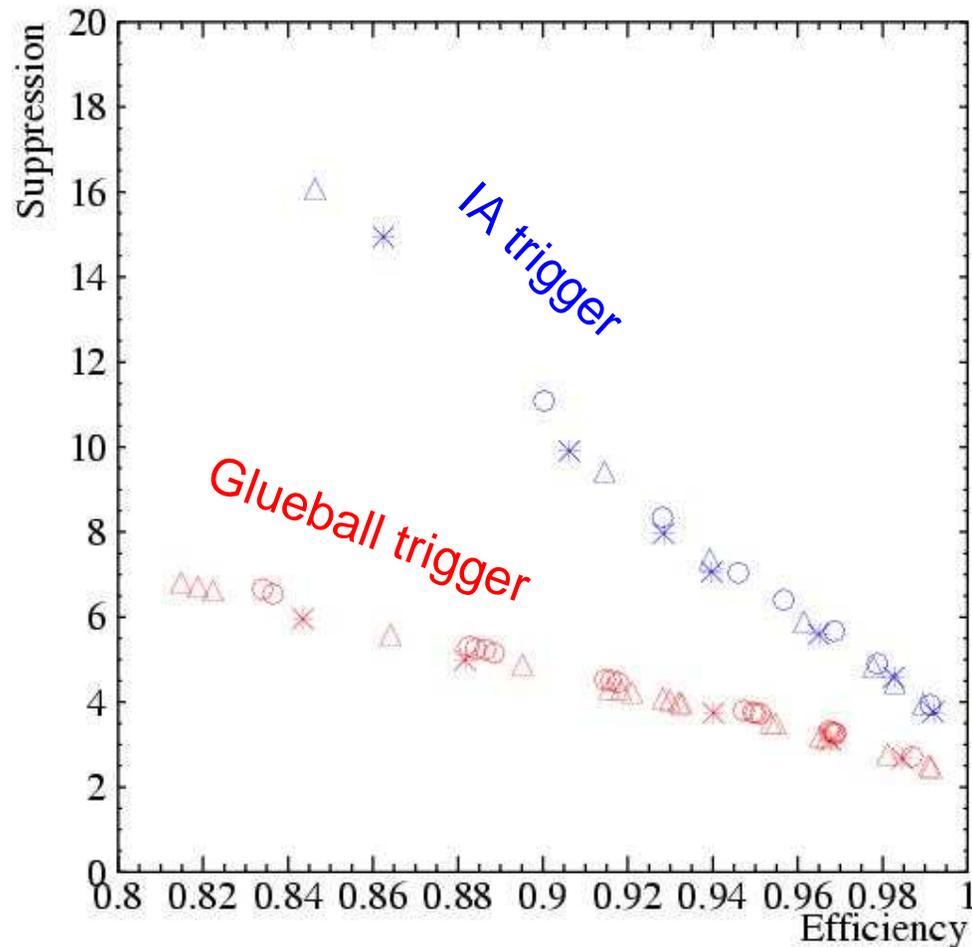
Long tail in amplitude
is attributed to interactions,
soft part is neutron bg
or electronics effect

Efficiency will be calculated
w.r.t empty bunches



Suppression vs. efficiency

Efficiency is calculated w.r.t. empty bunches, suppression w.r.t. all triggered ev



(all Glueball runs)

Veto condition:

Δ : $a1 < th1 \ \&\& \ a2 < th2$

O : $a3 < th3 \ \&\& \ a4 < th4$

$*$: $(a1 < t1 \ \&\& \ a2 < t2) \ || \ (a3 < t3 \ \&\& \ a4 < t4)$

Single plane works
as good as two planes
coincidence.

Suppression vs. efficiency (2)

We recommend to use a3, a4

th3	th4	eff	sup
113	94	81.5	6.8
137	110	89.5	4.9
184	140	95.5	3.5
290	215	99.2	2.5

Amplitudes available in Arte (since 04-01-r3) in table RSAC (filled by CARE) in usevnt.C

```
if(ArteTable<RSAC>::size() == 1) {  
    int a1 = ArtePointer<RSAC>(1)->a1;  
    int a2 = ArtePointer<RSAC>(1)->a2;  
    int a3 = ArtePointer<RSAC>(1)->a3;  
    int a4 = ArtePointer<RSAC>(1)->a4;
```

.....