Dissertation
submitted to the
Combined Faculties for the Natural Sciences and for Mathematics
of the Ruperto-Carola University of Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

presented by

Diplom-Physiker:	Oliver Brosch
born in:	Jugenheim an der Bergstraße

Oral examination: May 12, 2004

A Kaon Trigger for FOPI

Development and Evaluation of a
Trigger System for Strange Particles

Referees: Prof. Dr. Reinhard Männer
Prof. Dr. Norbert Herrmann

Ein Kaon-Trigger für FOPI – Entwicklung und Bewertung eines Triggersystems für seltsame Teilchen

In Stößen schwerer Ionen bei hohen Energien am SIS/GSI kann Kernmaterie großer Dichte und Temperatur erzeugt werden. Im FOPI-Experiment werden die Spuren der bei den Kollisionen entstehenden geladenen Teilchen nachgewiesen. Durch deren Analyse, insbesondere im Hinblick auf die seltsamen K-Mesonen, können unsere Kenntnisse über die Struktur der Materie und die Vorgänge beim Entstehen von Neutronensternen und Schwarzen Löchern vertieft werden. Kaonen werden jedoch selten beobachtet, und so sind die erzielbaren physikalischen Ergebnisse meist mit großen Fehlern behaftet. Um die Ausbeute an Kaonen deutlich zu erhöhen, wurde im Rahmen dieser Arbeit ein Trigger entwickelt, der das Zeit raubende Auslesen der kompletten Detektordaten für nicht interessante Ereignisse verhindern kann. Dafür wurde ein auf der Hough-Transformation basierender Algorithmus geschaffen. Dieser rekonstruiert Teilchenspuren aus einem kleinen Teil der Daten der Driftkammer CDC. Die geometrische Verknüpfung mit den Informationen des neuen, hoch auflösenden Flugzeitdetektors GRPC erlaubt die Bestimmung der gefundenen Teilchenarten. Um den Anforderungen durch die Datenbandbreite und die Rechenintensität zu genügen, sind für die Implementierung spezielle Prozessoren nötig. Etwa 5 bis 6 der an der Universität Mannheim entwickelten, FPGA-basierten MPRACE-Karten können diese Leistung kostengünstig erbringen. Die konsequente Parallelisierung der einzelnen Programmschritte ermöglicht das Ausnutzen der vollen Leistungsfähigkeit von MPRACE und damit eine Rechenzeit von weniger als $100\,\mu s$ pro Ereignis. Ausführliche Simulationen des Triggersystems zeigen, dass in Experimenten mit leichten Kernen wie Nickel bei Strahlenergien von $1.93\,GeV/u$ die K^+-Ausbeute etwa um den Faktor 6 und die K^--Ausbeute um den Faktor 11 gesteigert werden kann.

A Kaon Trigger for FOPI – Development and Evaluation of a Trigger System for Strange Particles

During collisions of heavy ions at high energies at SIS/GSI, nuclear matter can be exposed to high densities and temperatures. The FOPI experiment detects the traces of the charged particles produced in the nuclear reaction. Their analysis, in particular with respect to the strange K mesons, can extend our knowledge about the structure of nuclear matter and the processes during the evolution of neutron stars and black holes. However, kaons are rarely observed, hence the derived physics results suffer from large uncertainties. In order to significantly enhance the kaon yield, a trigger, which is able to prevent the time consuming readout of the complete detector data for uninteresting events, was developed within the scope of this work. For that purpose, a Hough transform based algorithm was created. It reconstructs particle tracks from a small fraction of the data of the drift chamber CDC. A geometrical matching to the information from the new high-precision time-of-flight detector GRPC allows the determination of the found particles' species. In order to fulfill the requirements from the data bandwidth and the computing intensity, special-purpose processors are required. About 5 to 6 of the FPGA based MPRACE boards developed at Mannheim University can provide this performance cost-effectively. The consistent parallelization of the individual program steps makes the exploitation of the full power of MPRACE and thus a processing time of less than $100\,\mu s$ per event possible. Detailed simulations of the trigger systems show, that in experiments with light nuclei like nickel at beam energies of $1.93\,GeV/u$ the K^+ yield can be enhanced by a factor of 6, and the K^- yield by a factor of 11.

Table of Contents

Chapter 1: Introduction

Hot and dense nuclear matter is an important research field in modern experimental and theoretical physics. It can only be explored by means of heavy ion collisions at medium and high energies, which generate extreme conditions over an enlarged period of time. The results obtained from such studies promise a better understanding of the nuclear matter itself, but also of astrophysical objects like neutron stars or the conditions in the early universe shortly after the Big Bang.

1.1 Kaons and Strange Particles

Strange particles, and especially positively charged kaons, are of particular interest for the analysis of heavy ion collisions. Unlike protons and neutrons, the building blocks of atomic nuclei, kaons contain a strange quark. Due to the conservation of strangeness in strong interactions, this makes them a unique probe for the conditions that existed during their production.

1.1.1 Kaon Production

The strange and anti-strange quarks (s and $\bar{s}$) are the third lightest among the six quark flavors. Charged kaons are composed of an (anti-) strange and an (anti-) up quark ($K^+ = u\,\bar{s}$ and $K^- = s\,\bar{u}$) and are therefore the lightest strange mesons with a mass of $m_K \approx 494\,MeV/c^2$ (see table 1.1 for more details). They are thus likely to be found as products of heavy ion collisions at sufficiently high energies. From a variety of possible reactions the most probable is

$$N\,N \;\rightarrow\; K^+\,Y\,N \tag{1.1}$$

with Y being a strange hyperon, $Y \in \{\Lambda, \Sigma\}$, and N a nucleon, $N \in \{n, p\}$, or a Δ resonance, and with a threshold energy of $1.6\,GeV$ for the production of kaons. Anti-kaons can e.g. be produced from a threshold energy of $2.5\,GeV$ via the reaction

$$N\,N \;\rightarrow\; N\,N\,K^+\,K^-, \tag{1.2}$$

or, even more frequently, via

$$Y\,\pi \;\rightarrow\; K^-\,N \tag{1.3}$$

(where π is a pion or π meson), as suggested in [Hart03]. Kaons are also obtained at energies below these thresholds through multi-step processes or collisions with clusters, or via the

particle	quark composition	rest mass m_0 $[MeV/c^2]$	lifetime τ $[s]$
baryons			
p (proton)	$u\,u\,d$	938.27	$>10^{32}$
n (neutron)	$u\,d\,d$	939.57	885.7
Λ (lambda)	$u\,d\,s$	1,115.68	$2.6\cdot10^{-10}$
Σ^+ (sigma)	$u\,u\,s$	1,189.37	$8.0\cdot10^{-11}$
Σ^0	$u\,d\,s$	1,192.6	$7.4\cdot10^{-20}$
Σ^-	$d\,d\,s$	1,197.4	$1.48\cdot10^{-10}$
Ξ^0 (xi)	$u\,s\,s$	1,314.8	$2.9\cdot10^{-10}$
Ξ^-	$d\,s\,s$	1,321.3	$1.6\cdot10^{-10}$
mesons			
π^+ (pion)	$u\,\bar{d}$	139.570	$2.60\cdot10^{-8}$
π^0	$(u\,\bar{u}+d\,\bar{d})/\sqrt{2}$	134.977	$8.4\cdot10^{-17}$
π^-	$\bar{u}\,d$	139.570	$2.60\cdot10^{-8}$
K^+ (kaon)	$u\,\bar{s}$	493.677	$1.238\cdot10^{-8}$
K_S^0	$(d\,\bar{s}+\bar{d}\,s)/\sqrt{2}$	497.65	$8.9\cdot10^{-11}$
K_L^0	$(d\,\bar{s}-\bar{d}\,s)/\sqrt{2}$	497.65	$5.2\cdot10^{-8}$
K^-	$\bar{u}\,s$	493.677	$1.238\cdot10^{-8}$

Table 1.1: Some properties of selected hadrons [Hagi02, Nave03].

modification of their effective masses in the dense hadronic medium [Li95b] (see section 1.2.2).

Kaons have a limited lifetime. The decay length of kaons is $\Gamma^{-1}=c\tau=3.713\,m$, and with the decay probability with respect to the flight distance s,

$$P(s) \;=\; \exp\left(-\frac{m_0\,s\,\Gamma}{p}\right), \tag{1.4}$$

only $1-P\approx65\%$ of the kaons with a transverse momentum of $p_T=0.5\,GeV/c$ are able to reach a detector situated around $R_{TOF}=1\,m$ apart from their origin (like the FOPI time-of-flight detector, see chapter 2.2.2), and accordingly even less for lower momenta [Devi00].

1.1.2 Strangeness

In strong interactions, strangeness is conserved. This means that during the expansion of the nuclear reaction fireball, the reaction cross sections for the positively charged K^+ mesons are mainly determined by elastic scattering. The consequence is, that a majority of them carries information about the conditions that existed during their production. Thus, the positively charged kaons are an excellent probe for studies of the hot and dense hadronic matter during the nuclear reaction [Seng01].

However, this is not true for anti-kaons, because they are also frequently absorbed e.g. in the reaction

$$K^{-} N \leftrightarrow \pi Y \tag{1.5}$$

(cf. eq. 1.3), that nevertheless preserves strangeness. The number of K^{-} detected outside the reaction zone is thus significantly reduced in comparison to the number of initially produced particles.

1.2 Compressed Nuclear Matter

Atomic nuclei at normal densities ρ_0 and low temperatures T_0 are not the only possible manifestation of nuclear matter. For example in a neutron star, elementary particles are compressed to extremely high densities, whereas in the early stages of the universe, ultra-high temperatures were reached. One of the aims of modern nuclear physics is to systematically explore the nuclear phase diagram in order to arrive at a valid equation of state, and to be able to explain the behavior of nuclear matter under extreme conditions.

1.2.1 Nuclear Phase Diagram

In heavy ion collisions at SIS energies, nuclear matter can be compressed to a few times the normal baryon density of $\rho_0 \approx 0.16\, fm^{-3}$. At even higher densities, a phase transition into the so-called Quark-Gluon Plasma (QGP) is expected to take place [Harr96]. In this state, the quarks and gluons, that are normally bound together by the strong force to form a.o. protons and neutrons, are deconfined. Moreover, the chiral symmetry of the Quantum Chromo-Dynamics (QCD) [Pich94, Pich99], which is the theory of particles subject to the strong force, should be restored. In contrast at normal densities, the chiral symmetry is broken [Bern95], which allows the existence of finite quark masses. In the high density phase of a heavy ion collision at SIS energies, the chiral symmetry may already partially be restored [Ecke95, Lutz00].

In the curse of a heavy ion collision, the reaction zone expands until the system reaches the freeze-out point, where particle production stops and collisions become rare. A schematic plot of the nuclear phase diagram can be seen in fig. 1.1.

1.2.2 In-Medium Effects

The properties of hadrons may be influenced by the medium that surrounds them [Ko96]. This mechanism can be explained by a medium dependent modification of the hadron-nucleon potential [Li97, Croc98, Croc99], that arises from the breaking and partial restoration of the chiral symmetry of the QCD. As a consequence, the effective masses of hadrons change. Theoretical calculations (see fig. 1.2) predict, that the mass of the K^{+} meson increases slightly by approx. 10% at hadron densities of $\rho/\rho_0 \approx 3$ [Aich00, Flor00], at which the effective mass of the

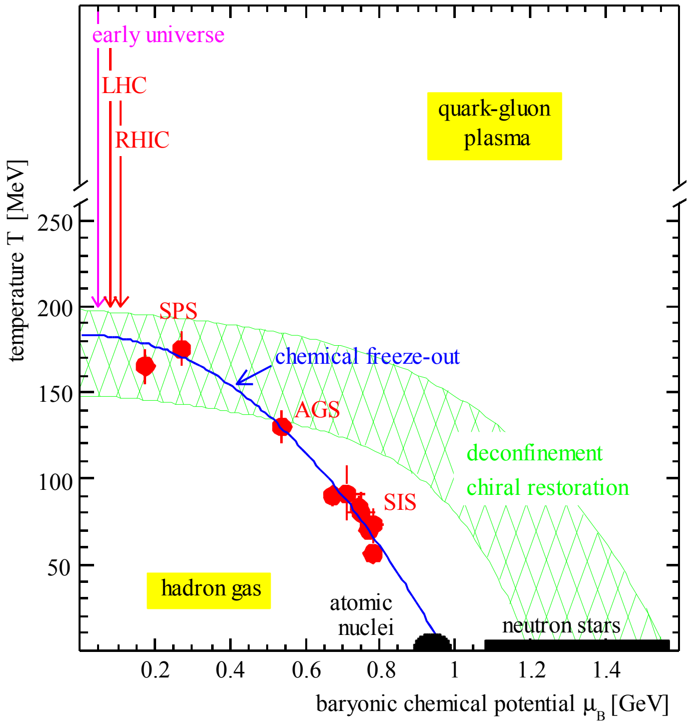


Fig. 1.1: Phase diagram of baryonic matter as a function of the baryonic chemical potential and the temperature. At high temperatures or particle densities, deconfinement and the transition into the Quark-Gluon Plasma is expected to take place.

K^- drops to around 50 % of its free value. The strength of the kaon-nucleon potential, and whether it is attractive or repulsive, also influences the phase space distribution of the hadrons in the final state of a heavy ion collision.

These effects have important consequences for the stability of neutron stars [Brow94, Li97b, Li97c, Seng99, Plet99]. If the dense hadronic matter lowers the production threshold for antikaons, they can already be formed in a collapsing star at densities around $\rho/\rho_0 \approx 3$ via

$$e^- \rightarrow K^- \nu_e. \tag{1.6}$$

This reaction is even further facilitated by the fact, that the electron mass is predicted to increase within a dense medium [Hard99]. The released binding energy causes the nuclear equation of state to soften [Brow92], thus allowing objects with smaller masses to collapse beyond the Schwarzschild radius to form a black hole. This limits the maximum possible mass for neutron stars to around 1.5 solar masses $M_\odot$ [Brow92b], which is compatible with observation.

1.2.3 Theoretical Approaches

Usually, the evolution of a heavy ion collision is emulated using microscopic transport models [Bass98] in order to draw conclusions from the observed phenomena. One parameter of particular interest is the compressibility of the hadronic matter, that determines whether the de-

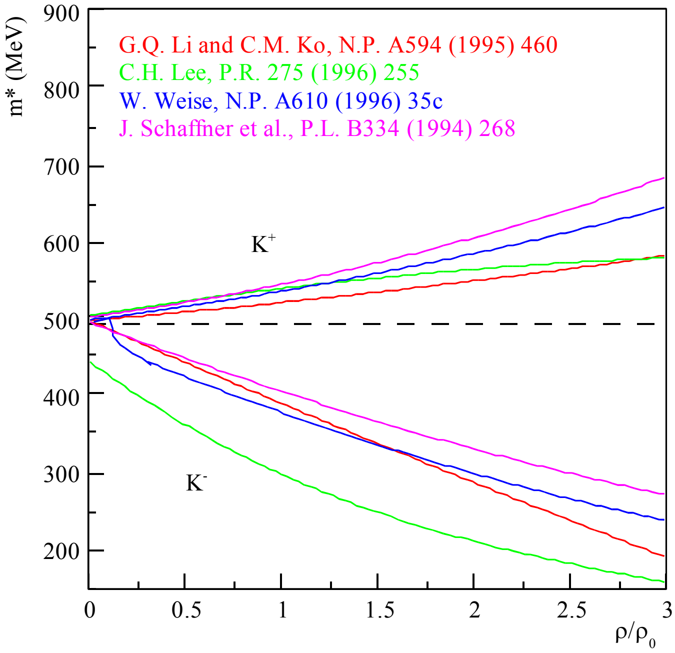


Fig. 1.2: Effective mass of K mesons as a function of the baryon density according to different theoretical predictions [Devi00b]. The mass of the kaon increases with density, whereas the anti-kaon mass is expected to drop substantially.

rived equation of state (EOS) is taken to be rather “soft” (corresponding to an incompressibility of $\kappa \approx 200\,MeV$) or “hard” (where $\kappa \approx 380\,MeV$). It allows the connection of the measured data through predicted in-medium effects to the maximum hadron density, that existed during the nuclear reaction. Fig. 1.3 shows two examples of such a comparison, where measured flow data (see chapter 1.3.2) is plotted together with model calculations using different values for the incompressibility. Calculations are frequently done using the many-particle Quantum Molecular Dynamics (QMD) [Aich91] or the one-particle Boltzmann-Uehling-Uhlenbeck (BUU) [Uehl33, Uehl34] models.

1.3 Observables

The hot and dense reaction zone cannot be observed directly. Instead, the experimenter has to rely on the detection of particles that were able to escape the fireball. However, certain properties of the obtained data allow conclusions about the conditions and processes during the early stages of the nuclear reaction.

The theoretical models (see section 1.2.3) are the connection between the observed particles with their properties and the fireball. Thus, observables are used for the determination of the model parameters, which correspond to physical properties of the nuclear matter. Other observables then serve as an evaluation of the model itself and the properties determined with its help.

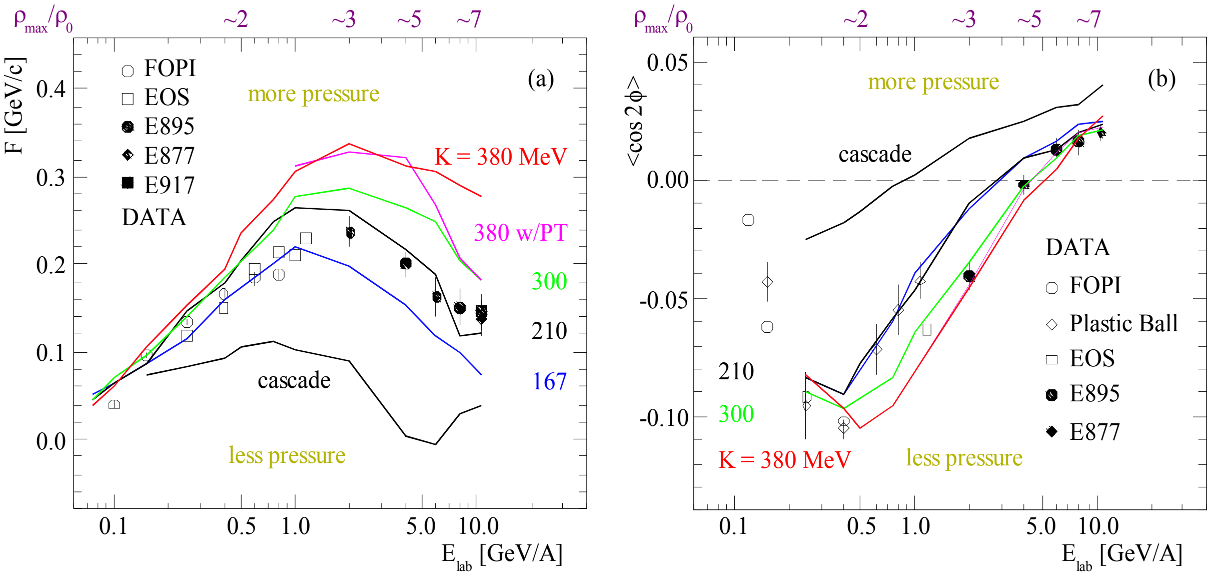


Fig. 1.3: Comparison of (a) sideward and (b) elliptic flow in Au + Au collisions as a function of the incident beam energy with theoretical predictions for different stiffnesses of the nuclear equation of state [Dani02].

1.3.1 Production and Ratios

In-medium effects (see section 1.2.2) strongly influence the production of kaons, because their effective masses are changed, and thus the energetic thresholds for their production are modified. Therefore, the production rate of kaons per participating nucleon $K^{\pm}/A_{part}$ depends on the system size, the collision centrality, the incident beam energy [Afan02], and the reaction time, which in turn depends on the other parameters. By variation of these event parameters the underlying mechanisms for the production of kaons can systematically be explored. On the other hand, kaon production also depends on the nuclear equation of state [Li95b, Hart00], which is responsible for an increase of the production cross-section at higher densities. Furthermore, enhanced strangeness production is proposed as one of the signatures of the QGP [Koch86, Harr96].

It has been suggested, that the influences from the in-medium modification of the K^+ mass and from the nuclear EOS nearly cancel out [Hart00, Aich00], so the underlying mechanisms are difficult to probe by means of kaon yields alone. However, recent results of the KaoS collaboration on subthreshold K^+ production in $Au+Au$ and $C+C$ collisions [Seng99, Stur01, Fuch01] clearly favor a soft EOS.

In the case of K^-, things become even more difficult: their yield and thus also the ratio of produced kaons and anti-kaons K^+/K^- [Li98, Croc00b, Seng00] turn out to be nearly constant, because virtually all observed K^- originate from the reaction $Y\pi \rightarrow K^- N$ [Aich00]. This channel is very frequently reversed during the nuclear reaction (see section 1.1.2), so only a fraction of the initially produced kaons can be observed.

Results of the FOPI collaboration on strangeness production and ratios have e.g. been reported in [Best96, Best97, Plet99, Wisn00, Devi00, Wisn00b, Devi02].

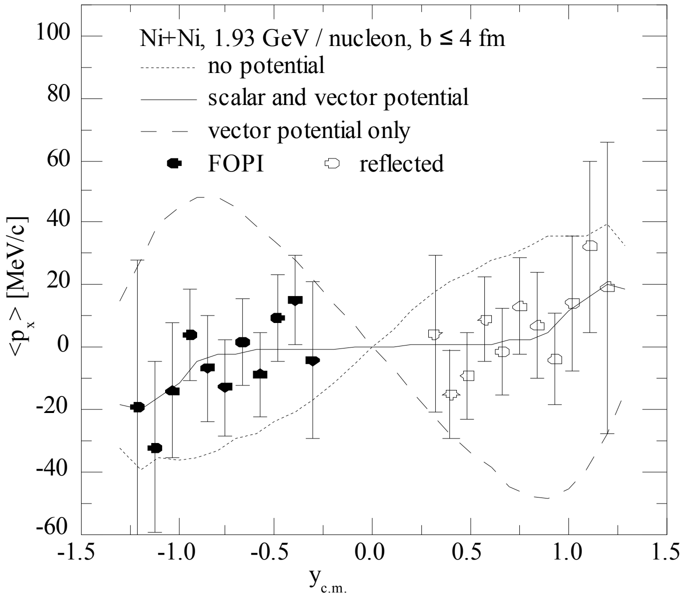


Fig. 1.4: Comparison of kaon sideward flow in Ni + Ni collisions as a function of rapidity with theoretical predictions, calculated with and without kaon-nucleon potentials [Ko96].

1.3.2 Collective Effects

The final phase space distribution of particles produced in heavy ion collisions is influenced by flow phenomena [Dani99, Dani01]. It is possible to distinguish between radial flow [Siem79, Lisa95], that describes the radial expansion of the fireball; sideward flow [Dani85, Croc00b], which can be expressed by the first Fourier coefficient of the azimuthal momentum distribution $F_1 = \langle \cos \varphi \rangle$ (or also $F_1 = \langle p_x \rangle$) of the reaction products; and elliptic flow [Stöc82], which corresponds to the second Fourier coefficient $F_2 = \langle \cos 2\varphi \rangle$. The observation of even higher order contributions has been predicted [Volo96].

In connection with strange mesons, some of these collective effects show deviations from the general behavior of the hadronic matter. The flow of strangeness is different from that of e.g. protons and pions [Herr96, Davi99, Aich00], due to the kaon-nucleon potential (section 1.2.2). This circumstance allows conclusions about this potential [Li95, Ritm95b], in-medium effects and the equation of state. A comparison of kaon sideward flow with theoretical calculations using different kaon-nucleon potentials is shown in fig. 1.4: in this case, the data points clearly favor the presence of both a scalar and a momentum dependent potential.

Collective effects in heavy ion collisions at SIS energies have e.g. been discussed in [Reis02, Kreß02]. Results from FOPI on strangeness dynamics have been published in some of the references mentioned in the previous section, and in addition e.g. in [Andr99, Croc00].

1.4 About this Work

The probing of fundamental properties of the QCD and the nuclear EOS by means of heavy ion collisions remains an important research field in hadron physics. The measured data and theoretical models developed so far do not form a unique picture, so many questions are still open. Strange particles, especially the positively charged kaons, are thanks to their long mean free path inside hadronic matter a promising probe for the conditions and processes during the nuclear reaction. Also the anti-kaons are of particular interest, due to their specific interactions inside the hot and dense medium. However, these investigations could provide results of greater evidence, if kaons and especially anti-kaons were less rarely produced. In the future, measurements with higher precision and better statistics are crucial.

In this thesis, a kaon trigger for the enhancement of the strangeness contents of FOPI experiments is introduced. The FOPI experimental setup, including its data acquisition, is described in chapter 2. Chapter 3 develops the trigger concept. The trigger algorithm, the framework necessary for its simulation and evaluation, and issues related to the trigger hardware, are addressed in detail in chapters 4 through 6. In chapter 7, performance results of the trigger algorithm, the whole trigger system, and implications on the hardware dimensions are presented. Chapter 8 closes with a discussion of the results. Finally, appendix A features the documentation of the user interface of the simulation framework, including important details of the implementation such as file formats, parameters, and other options.

Chapter 2: The FOPI Experiment

The FOPI experiment is located at the GSI (Gesellschaft für Schwerionenforschung mbH), a member of the HGF (Helmholtz-Gemeinschaft Deutscher Forschungszentren) and supported by the Hessian and German governments, in Darmstadt-Wixhausen, Germany. FOPI stands for "4π", indicating that its detectors cover almost the complete spherical angle of $\Omega = 4\pi$ in the center-of-mass system of the heavy ion reactions.

FOPI is a fixed target experiment that uses a heavy ion beam of medium energy – mostly in the range of several $100\,AMeV$ to $2\,AGeV$ ($1\,AGeV \triangleq 1\,GeV/u$) – as the projectiles. Its detectors are able to determine particle species, energies and angular distributions of charged fragments produced in the nuclear reactions.

The FOPI collaboration currently (as of May 2003) consists of approx. 55 scientists from 12 institutes in Germany, France, Poland, Russia, Croatia, Romania, Hungary and South Korea [FOPI03].

2.1 GSI Accelerator Facility

GSI experiments are driven by three different heavy ion accelerators [GSI01]. They are capable of producing energies of up to $2\,AGeV$, for ions ranging from carbon to uranium.

- The **UNILAC** is a universal linear accelerator with a total length of $120\,m$. Charged ions from two independent ion sources are injected into the UNILAC and accelerated to approx. $1.4\,AMeV$ in a first stage. After passing a stripper, that increases their charge state by removing outer electrons, the particles are further accelerated to between 3 and $13\,AMeV$. Alternatively, ions produced by a high charge ion source can directly be injected into the second stage. Accelerated particles are used for low energy experiments, or directed to the SIS for further acceleration. For SIS injection, usually ion energies of $11.4\,AMeV$ are used.
- The **SIS** (Schwerionen-Synchrotron) is a synchrotron for heavy ions and has a circumference of $216\,m$. The pre-accelerated ions from the UNILAC are stripped before injection into the SIS, and are then further accelerated to energies ranging from $50\,AMeV$ up to $2\,AGeV$. In the case of uranium ions, the maximum charge state of U^{92+} and an energy of $1.35\,AGeV$ can be reached by again stripping the accelerated ions up to their highest possible charge and re-injecting them into the SIS via the ESR. The output can either be directed to the high energy experimental area, where the FOPI experiment resides, or to the ESR.
- The storage ring **ESR** (Experimenteller Speicherring) has half the circumference of the SIS (i.e., $108\,m$) and features devices for electron cooling and statistical cooling. It is able to

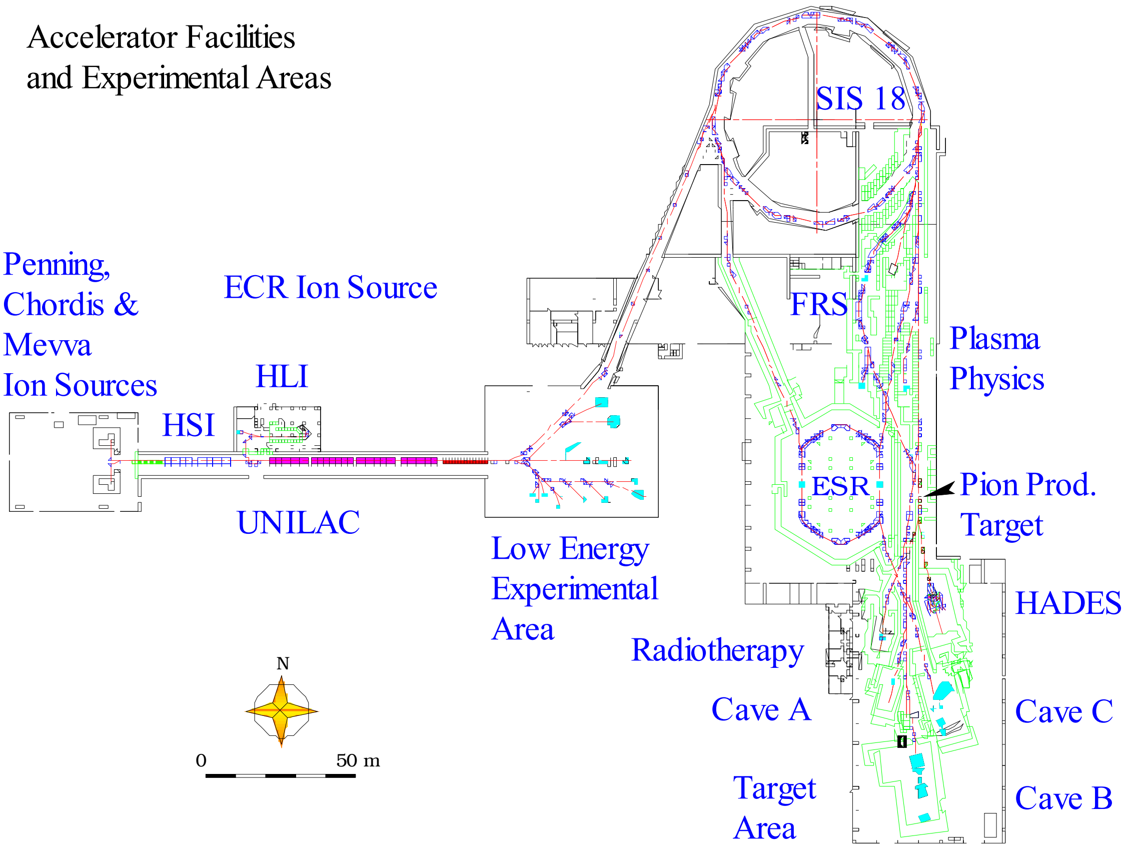


Fig. 2.1: Layout of the GSI accelerator facilities [GSI01]. The ion sources can be seen on the left, the SIS at the top, and the SIS experimental area at the bottom right. The FOPI experiment resides in "Cave B" of the target area.

store heavy ions for up to a period of a few seconds and can therefore be used for the accumulation of rare isotopes.

The layout of the accelerator setup and the experimental areas is sketched in fig. 2.1.

2.2 FOPI Detector Setup

FOPI's setup [Ritm95] (see fig. 2.2) is located in Cave B of the SIS experimental area at GSI. The central detectors are embedded in a superconducting solenoid, that supplies them with a magnetic field of $B=0.6\,T$, which is nearly homogeneous ($\Delta B/B<2\%$).

2.2.1 Central Drift Chamber (CDC)

The Central Drift Chamber (CDC) is the heart of FOPI's detector setup. It detects three-dimensional tracks of charged particles and energy losses along these tracks. The cylindric drift chamber is placed such, that its axis meets the beam axis, and that the target position is inside the chamber. Its inner and outer radius are $\rho_{IN}=20\,cm$ and $\rho_{OUT}=80\,cm$, respectively. It is di-

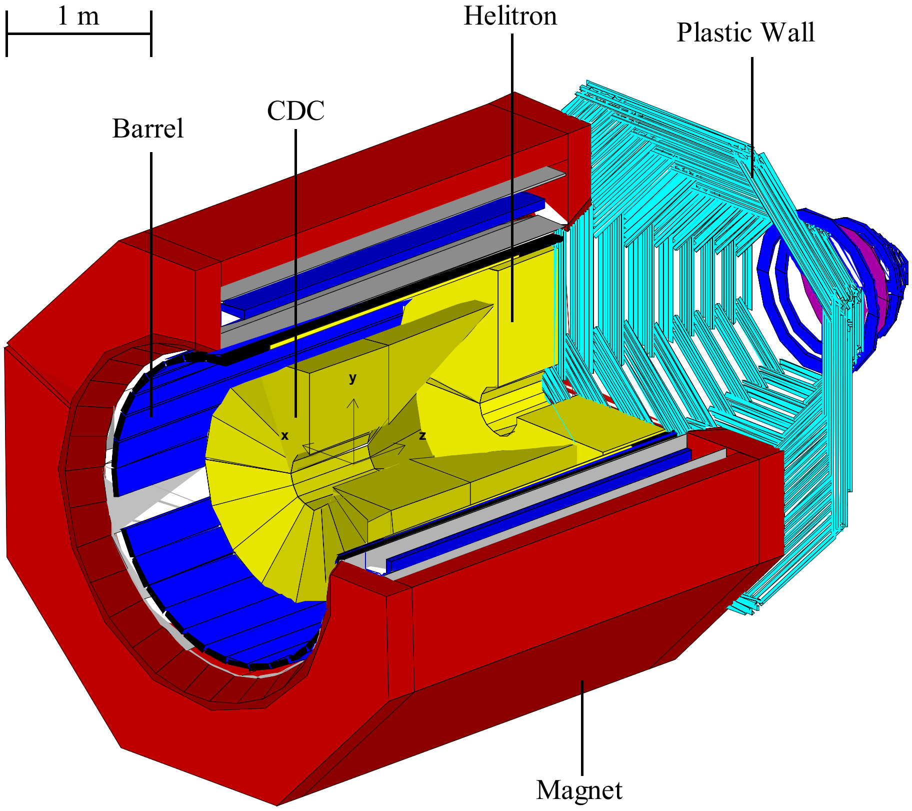


Fig. 2.2: Layout of the FOPI detector “Phase 2” [GEAN93]. The old TOF barrel (blue), the drift chambers (yellow), and the Plastic Wall (turquoise and blue/violet) are shown. The heavy ion beam is injected from the left, hitting the target at the origin of the small coordinate system.

vided into 16 sectors, rotation-symmetric around the beam axis. The sectors are tilted by $\varphi_{TILT}=8\,°$ and limited by field wires. The middles of the sectors are equipped with additional planes of wires, alternately field and sense wires. The field wires are put to high voltage of $-1.55\,kV<U_F<-1.15\,kV$. Each sector has 60 sense wires spaced by $d=1\,cm$. For the target and the CDC being in their standard positions, the CDC covers polar angles (deviation from the beam direction) of $32.5\,°<\vartheta_{CDC}<140\,°$. The length of the sense wires varies between $l_{IN}=86\,cm$ (innermost) and $l_{OUT}=190\,cm$, so the forward and backward ends of the CDC run inwards by $\varphi_{FWD}=30\,°$ and $\varphi_{REV}=60\,°$, respectively. All sense wires and drift wires have been exchanged during the 2001 / 2002 CDC upgrade. Profiles of the CDC layout are shown in fig. 2.3.

The chamber is filled with a gas mixture consisting of 88% argon, 10% isobutane and 2% methane slightly above atmospheric pressure. Drift gas, magnetic and electric fields lead to a constant drift velocity for ionized gas particles of $v_d\approx 43.7\,\mu m/ns$ under a Lorentz angle (between drift direction and the normal to the sense wire plane) of $\alpha_L\approx 32.5\,°$. Thus, the maximum drift time is $t_d\approx 5\,\mu s$ after an event.

If a charged particle crosses the CDC volume, it ionizes a certain amount of gas molecules characteristic for its mass and energy. The ionized track drifts towards the sense wires, where it causes voltage peaks that are read out on both sides of the wires. The particle track can later

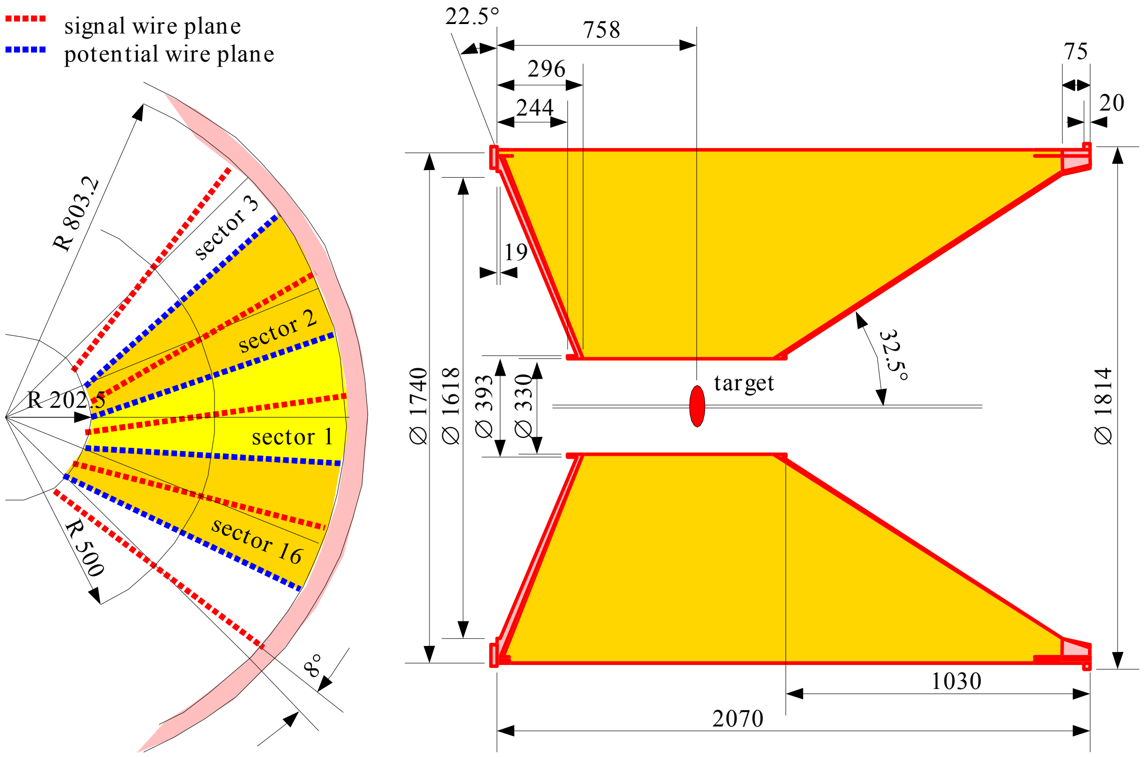


Fig. 2.3: Sketch of the geometry of the Central Drift Chamber as viewed from the front (left) and from the side (right) [Kuts99]. The 16 sectors are tilted by 8°, and the length of the sense and the field wires increases with the distance from the beam axis.

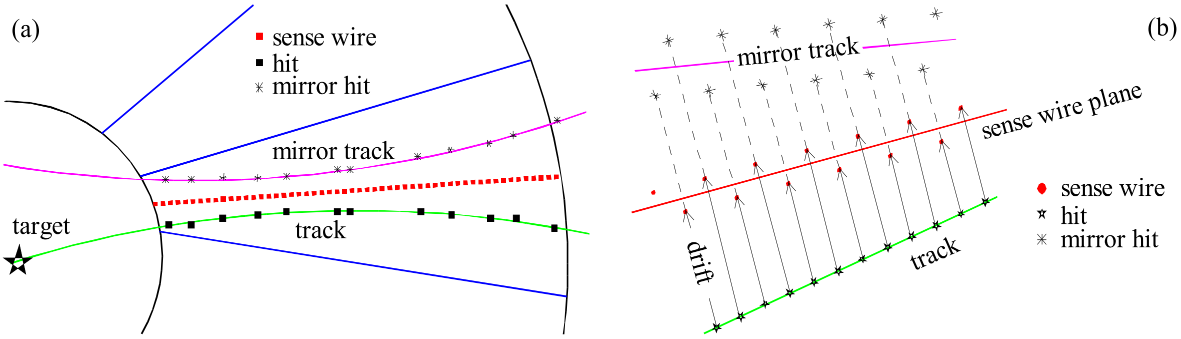


Fig. 2.4: Tilting and staggering in the CDC [Andr98]. Both techniques are used to distinguish tracks from mirror tracks. Tilting (a) prevents mirror tracks from appearing to originate from the target. Staggering (b) causes the mirror hits to alternate around the mirror track.

be reconstructed using the drift time from each sense wire (components in the plane perpendicular to the beam direction), and the ratio of the charges detected on both ends (component along the beam direction). This process is illustrated for two dimensions in fig. 2.5.

For each particle track, a drift chamber offers two possible reconstructions, because it can not decide, whether a charge drifted towards a sense wire from the one or the other side. However,

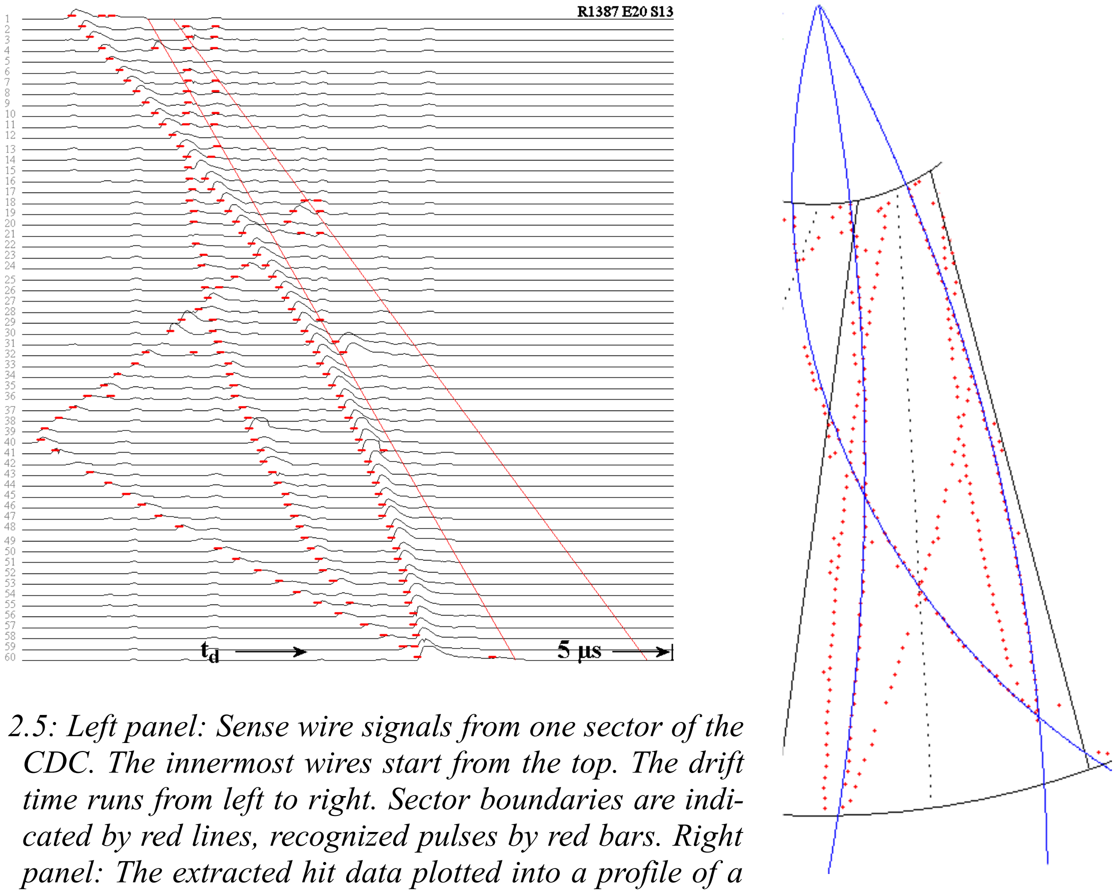


Fig. 2.5: Left panel: Sense wire signals from one sector of the CDC. The innermost wires start from the top. The drift time runs from left to right. Sector boundaries are indicated by red lines, recognized pulses by red bars. Right panel: The extracted hit data plotted into a profile of a sector of the CDC. Reconstructed particle tracks are represented by blue lines.

due to the tilted sense wire planes, the wrong solution does not appear to emerge from the center-of-collision (see fig. 2.4). In addition, if the track crosses a sector boundary, the wrong reconstruction does not continue in the neighboring sector. Instead, a second track segment appears on the other side of the boundary at a different location.

In fact, a second mechanism is built into the CDC to avoid reconstruction of the wrong tracks. The sense wires do not lie exactly in a plane, but are shifted alternately to either side out of plane by $d_{STG} = 100\,\mu m$ ("staggering", see fig. 2.4). While real tracks are not affected by this method, mirror hits are oscillating around an assigned track. However, this feature is difficult to exploit, because the amplitude of the staggering is close to the geometrical resolution of the CDC.

The CDC momentum resolution as measured for charged pions varies from $\Delta\, p_T / p_T \approx 4\,\%$ for $p_T < 0.5\, GeV/c$ to $\Delta\, p_T / p_T \approx 12\,\%$ for $p_T \approx 2\, GeV/c$ [Andr01].

2.2.2 GRPC Time-of-Flight Detector

A time-of-flight (TOF) detector measures time (relative to the start of an event) and location of particles that traverse it. The GRPC barrel is a new subdetector for FOPI, whose setup is currently being completed. It uses the technology of multi-gap Glass Resistive-Plate Cham-

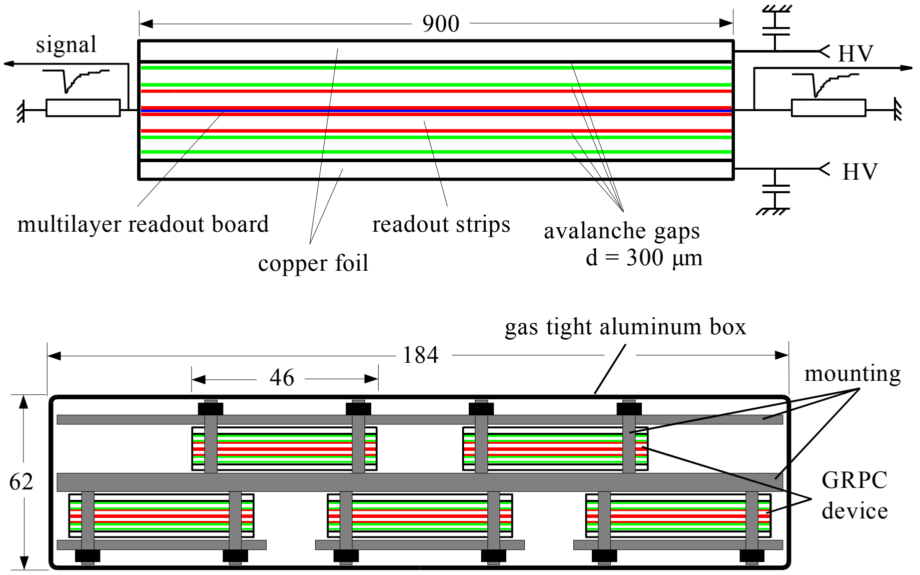


Fig. 2.6: Sketch of a single GRPC device [Petr02] (side view, top), and a GRPC super-module (front view, bottom), that contains five GRPC devices.

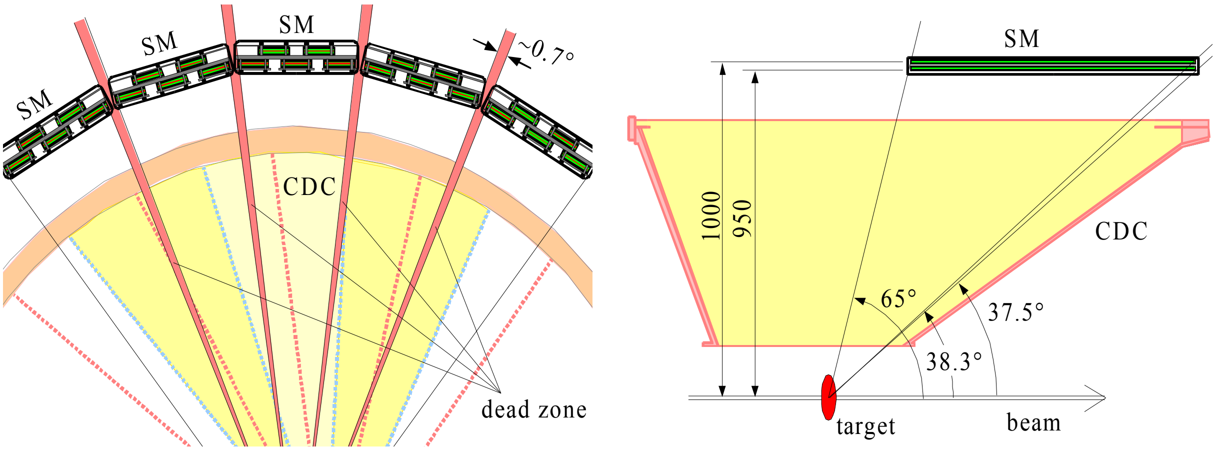


Fig. 2.7: Mounting of the GRPC super-modules (SM) around the CDC as viewed from the front (left) and the side (right). 33 super-modules cover 90–95% of the full azimuth.

bers (GRPC) [Petr02], that provide a clearly better resolution than the old TOF subdetectors (see following section for an overview).

A GRPC consists of a stack of several glass plates with gaps of $d = 300\,\mu m$ in between (see fig. 2.6). The plates are enclosed between high voltage electrodes ($U = 9.6\,kV$) that produce an electric field in the gaps. The system is kept under a mixture of 85% $C_2F_4H_2$, 5% isobutane and 10% SF_6 at atmospheric pressure. The TOF signal is read from an electrode in the middle of the glass stack. The electrode is divided into 16 strips, that are read out at both ends.

Five chambers are mounted in an aluminum box of size $184 \times 62 \times 900\,mm^3$ (see fig. 2.6), and 33 boxes are put around the CDC in parallel to the beam axis (fig. 2.7). The distance between a chamber and the beam axis is $95\,cm < R_{GRPC} < 100\,cm$. The GRPCs cover polar angles of $37.5\,° < \vartheta_{GRPC} < 65\,°$ and about $90-95\,\%$ of the azimuthal angle.

If a charged particle crosses the stack of glass plates, it ionizes the gas, which then avalanches towards the readout electrodes. The time difference of the signals read from both ends of the electrodes gives the longitudinal position on the respective strip. The position across the chamber can be determined by the charge distribution among neighboring strips. As was measured for different prototypes, the accuracy of the time signals is in the order of $\sigma_{TOF} < 70\,ps$, depending on the mechanical GRPC setup.

2.2.3 Other Detectors

The FOPI setup includes several other detectors, that are not used for the trigger system presented in this work. However, they are shortly mentioned for completeness:

- The **TOF Barrel** is located around the CDC in backward direction of the GRPCs. It measures the time-of-flight and location of particles that have crossed the CDC beyond the coverage of the GRPCs. It originally consisted of two parts:

  The **Plastic Barrel** is an array of 180 scintillators. The current version covers a polar angle of $65\,° < \vartheta_{BAR} < 135\,°$ and has a time resolution of around $\sigma_{TOF} < 150\,ps$.

  The **Čerenkov Barrel** was located around the Plastic Barrel and also consisted of 180 elements, filled with an appropriate saline solution. This detector was removed from the setup in order to provide enough space for the new GRPC barrel.

- The **Helitron** is FOPI's second drift chamber, located in forward direction relative to the CDC. It detects the tracks of particles emerging at polar angles of $7\,° < \vartheta_{HEL} < 30\,°$. The inner and outer radii of the Helitron are $\rho_{IN} = 20\,cm$ and $\rho_{OUT} = 100\,cm$, respectively, and its length is $l = 60\,cm$. The Helitron is divided into 24 sectors. Other than in the CDC, the sense wires in the Helitron are mounted in radial direction and have all the same length.

- The **Plastic Wall** (PlaWa) is located in forward direction behind the Helitron. It measures the energy loss and time-of-flight of particles, whose tracks are detected in the Helitron. It consists of an outer and an inner part:

  The **Outer Plastic Wall** is an umbrella-shaped array of 512 scintillators. It covers polar angles of $7\,° < \vartheta_{OPW} < 30\,°$.

  As indicated by its name, the **Zero Degree** detector or Inner Plastic Wall covers small polar angles of $1\,° < \vartheta_{ZER} < 7\,°$. It is an array of 252 scintillators.

- The **Cluster Detector** or ΔE Shell covers the same polar angle as the Plastic Wall. It is used for measurements of the energy loss of clusters (heavy fragments) and fast particles, and is also divided into two parts:

  Polar angles of $1\,° < \vartheta_{ROS} < 6\,°$ are covered by the **Rosace**, an array of 60 thin scintillator paddles located in front of the Zero Degree Detector.

  The polar range $6\,° < \vartheta_{PAR} < 30\,°$ is covered by the **Parabola**, an ensemble of 16 gas ionizing chambers, each with 8 anodes, thus leading to a granularity of 128 elements. The Parabola are located between the Helitron and the Plastic Wall.

- The **Start Counter** is a thin scintillator foil located in front of the target and is used to trigger the start of an event. Its time signal has a resolution of $\sigma_{STC} = 50\,ps$ and serves as a reference for the TOF detectors. Its energy signal can be used to reject multiple events.

FOPI was built starting in 1992 in three steps: “FOPI Phase 1” included only the Forward Wall (PlaWa and Cluster), “FOPI Phase 2” added the Solenoid, the drift chambers and the Barrel. Finally the FOPI upgrade, which started in 1998, enhances FOPI's data acquisition performance and introduces the GRPCs.

2.3 Data Acquisition

During an experiment, the particles emitted from the interaction zone produce electric signals in the detectors. As seen in the previous section, these signals occur due to avalanches in the chambers or in the photo multipliers attached to the scintillators. The task of the data acquisition (DAQ) is to select appropriate events, reduce the amount of data collected in the drift chambers, and to store the remaining data for later analysis. During the FOPI upgrade, most DAQ components were replaced or modified in order to improve performance. This section describes some aspects of the new DAQ system.

2.3.1 Components

The FOPI DAQ system consists of many specialized electronics components. A rough knowledge of this system is necessary in order to understand and solve FOPI's trigger problem (see chapter 3). These are the components connected to the detectors which are relevant for this work:

- Actually, a basic trigger functionality is currently included in FOPI's DAQ, and is referred to as the first level trigger. It uses the charged particles' multiplicities in the Plastic Wall together with information from the start counter in order to enrich the read out data with more central events.

- The readout chain for the drift chambers (both CDC and Helitron) starts with Flash Analog / Digital Converter (FADC) modules. Each module features eight converters, that digitize the signals from both ends of four sense wires at a sampling frequency of $f_{FADC} = 100\,MHz$ and an effective accuracy of $b_{eff} = 10\,bit$. 24 FADC modules are located in a single FADC crate.

  Half of each FADC crate (twelve modules with together 96 channels) is read out by a single Scanning and Sampling Module (SSM) at a frequency of $33\,MHz < f_{SSM} < 40\,MHz$. The SSM performs a rough hit detection on the FADC data by separating the so-called hitlets from the background noise and thus reducing the amount of data that is passed to the next module in the readout chain.

  Two SSMs transfer their data via a GTB (Geräte Transport Bus, German for device transport bus, a differential SCSI-like interface used in GSI experiments, that can transfer 16-bit data at $f_{GTB} = 20\,MHz$) to a SAM3 (Steuerungs- und Auslesemodul, German for control and readout module). The SAM3 performs double hit recognition and data reduction on the hitlets that are supplied by the SSMs. Only arrival times, charge deposit, rise and integra-

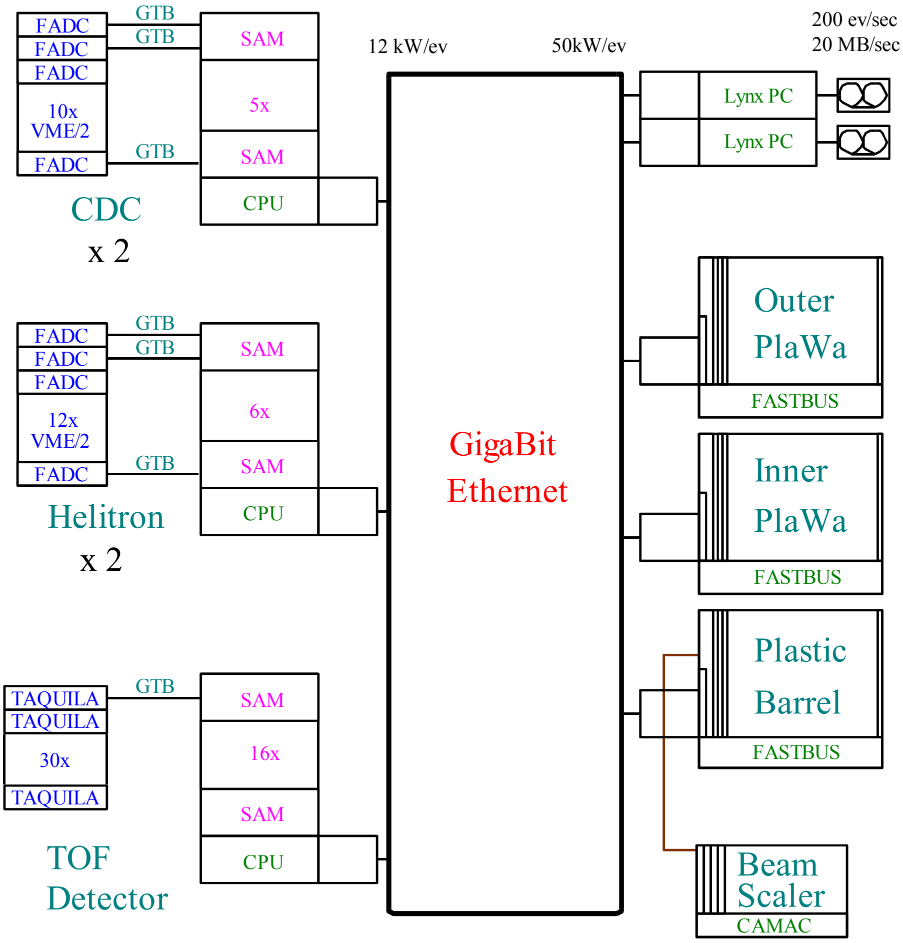


Fig. 2.8: Layout of FOPI's upgraded DAQ system [Stoc02]. The relevant modules are explained in the text.

tion times remain after the reduction. The SAM3 is also used to control and program the SSMs and the FADCs.

Finally, the subevent builder RIO3 (a commercially available PPC computer with VME interface) collects the data from different SAM3 units and passes it to the event builders (see below).

- The GRPC readout chain starts with the front end electronics (FEE) modules, that amplify the pulses of 16 GRPC channels with a gain factor of up to $V_{FEE}=300$ and are mounted directly at the GRPCs.

  The next module is called TAQUILA and combines the following functionalities: a Time-to-Amplitude Converter (TAC) converts arrival times to pulse heights just like its name suggests. This amplitude as well as the amplitude of the original pulse are digitized by separate ADCs, and the data of 16 channels are collected in a FIFO buffer. Additional control logic sends the information (time, amplitude and channel number) via a GTB connection to a SAM3 unit upon request.

The remaining part of the readout chain is (at least in terms of employed hardware) identical to the one used for the drift chambers. However, no data reduction needs to be done by the SAM3 units.

- All RIO3 subevent builders send their data via a Gigabit Ethernet switch to two global event builders (also PPC computers running a real-time operating system). Attached to them are the final nodes of the readout system: the storage devices.

A more detailed description of the new DAQ system can be found in [Stoc02]. Fig. 2.8 shows a sketch of the DAQ layout.

2.3.2 Performance

Although many components of the readout system have been replaced during the FOPI upgrade program, a significant dead time remains. This is mainly due to the large amount of data to be extracted from the drift chambers and the additional data reduction that has to be done.

A new event cannot start before all FADC scanners have completely been read out by the SSMs. In addition, the dead time cannot be smaller than the slowest stage in the whole readout pipeline. This introduces a dead time in the order of $1.2\,ms < t_{DEAD} < 2.5\,ms$. In the current version of the SSM units, scanning of the subsequent event's data has to wait until the data of the current event is completely transferred to the SAM3 nodes. This further increases the dead time to values of $t_{DEAD} > 2.5\,ms$. The event rate for the complete FOPI detector system is in the order of $f_{FOPI} \approx 200\,Hz$.

2.3.3 Interfaces

In addition to the interfaces that connect the different modules of the readout chains, the FOPI DAQ system also provides interfaces that allow the connection of external devices, e.g. for monitoring and other purposes. They can also be used for feeding a trigger module (see chapter 3), which is the subject of this thesis. Within this context, two of these interfaces are important: one interfaces to the CDC data, the other to the GRPC readout.

The FADC modules do not only provide access to the digitized sense wire data, which is delivered upon request to the SSMs. In addition, a reduced sense wire signal is available in real-time, i.e. within the first $t_d = 5\,\mu s$ after an event, at a front output of the FADC modules. This signal is a discriminated combination of the signals of both ends of a sense wire. It is high for $t_{DCR} = 50\,ns$ whenever the combined sense wire signal rises above an adjustable threshold, and low otherwise. If the threshold is set carefully, this signal can be used to obtain the arrival times of all sense wire pulses. However, additional information like charge deposit and distribution cannot be derived from this signal. Fig. 2.9 illustrates the generation of this signal, a block diagram of an FADC module is shown in fig. 2.10.

The GTB, which connects the GRPC's TAQUILA readout module to the SAM3 unit, can be intercepted by a GTB slave interface, the SNOOP module. This module makes the TOF data available to other recipients by simply duplicating the data stream. The full TOF data can thus be accessed within $t_{GTB} = 50\,\mu s$ after an event. The interconnections and a block diagram of the SNOOP module are depicted in fig. 2.11.

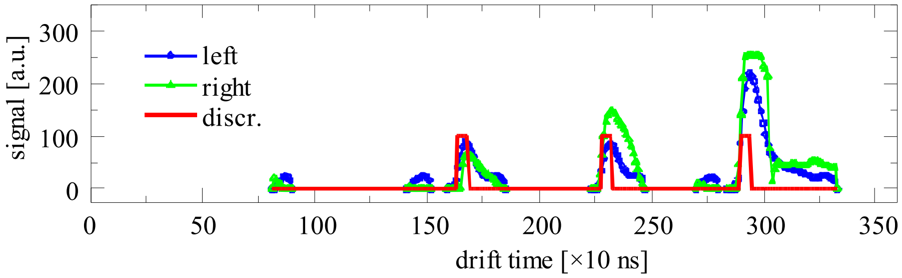


Fig. 2.9: Digitized signals read from both ends of a single sense wire, after zero-suppression. The signal shown in red is the output of the FADC discriminator for the same wire.

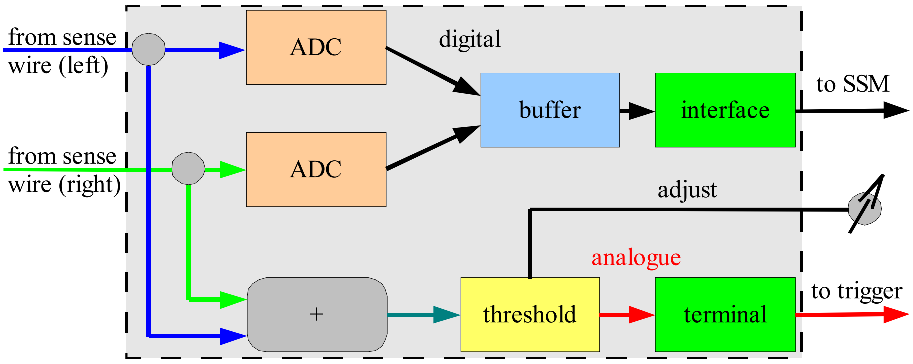


Fig. 2.10: Block diagram of relevant functional units inside the FADC modules. The FADC receives sense wire signals from both ends and provides a digitized output upon request, and in addition an analogue discriminated signal in real-time.

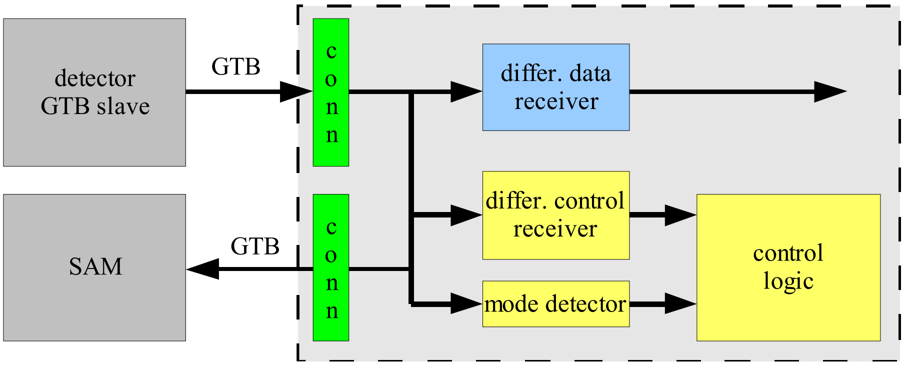


Fig. 2.11: Block diagram of the SNOOP interface that resides inside the GTB.

2.4 Event Reconstruction

Once an experiment is finished and its data is available in the event database, the relevant physical quantities have to be extracted. As the first step, it is necessary to reconstruct single events from these data, i.e. to identify the particles and determine their properties. Recalling the collected data, one finds among others

- arrival times of pulses from the drift chambers

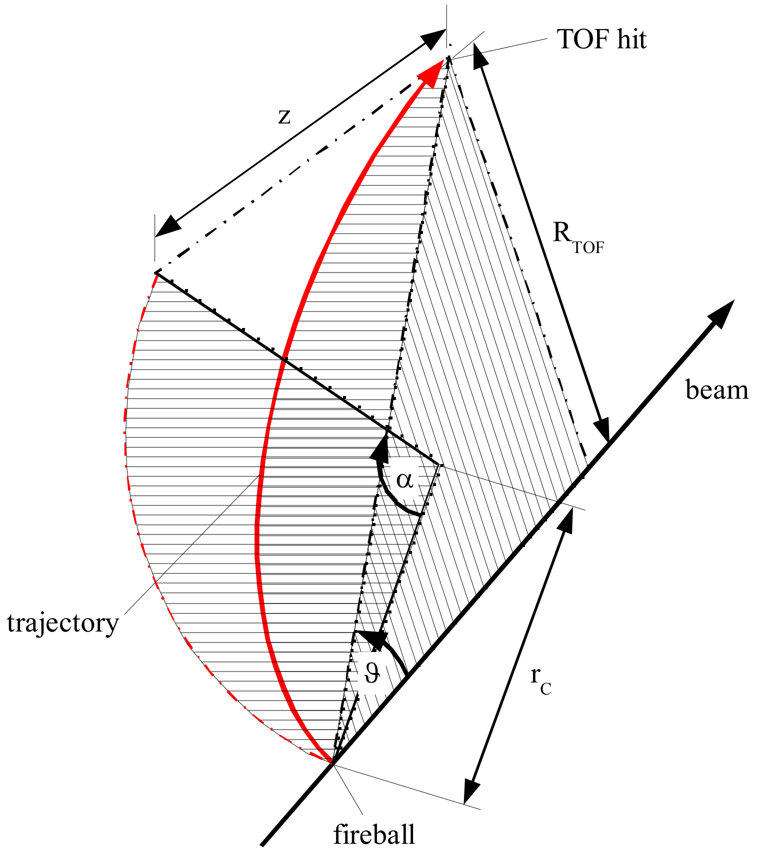


Fig. 2.12: Geometry used for the reconstruction of particles and their trajectories. Refer to text for details.

- numbers of the wires on which the pulses occurred
- arrival times and z coordinate of pulses from the TOF detectors
- numbers of the channels which detected the pulses.

Any of the mentioned information can as well be extracted from the data available at the interfaces discussed in the previous section.

A particle can be identified by only two properties, namely its electric charge q and its invariant mass m_0. The following calculation describes the connection of m_0 to quantities more directly connected to the data from the detectors, namely the particle's time-of-flight t, the z coordinate of its hit on the TOF detector, and the radius r_C of its trajectory. r_C and the correlation of tracks and TOF hits have to be extracted from the detector data by well sophisticated algorithms (see chapter 3.1.2).

Let B be the magnetic field inside the detector and ϑ the polar angle of the TOF hit with respect to center-of-collision and beam axis, so that

$$\tan\vartheta = \frac{R_{TOF}}{z} \qquad (2.1)$$

(see fig. 2.12 for a sketch of the geometric construction). Then the momentum in eV/c of a particle with charge $q=\pm e$ can easily be calculated by

$$p = \frac{c B r_C}{\sin \vartheta}. \tag{2.2}$$

Considering now a projection of a charged particle's track (which due to the magnetic field parallel to the beam axis has the shape of a helix; its projection is a circle) to a plane perpendicular to the beam axis: let α be the angle between the normals to the track through the center-of-collision and through the TOF hit of this track. R_{TOF} is the distance between the TOF hit and the beam axis, then

$$\alpha = 2\sin^{-1}\left(\frac{R_{TOF}}{2 r_C}\right), \tag{2.3}$$

and $d=\alpha r_C$ is the length of the projected track. The total length of the track results to $s=(d^2+z^2)^{-1/2}$, and the particle's average velocity (ignoring the energy loss due to ionization along the track) $v=s/t$. Now

$$\beta = \frac{v}{c} \quad \text{and} \quad \gamma = \frac{1}{\sqrt{1-\beta^2}} \tag{2.4}$$

are the representations of the velocity commonly used in relativistic calculations, with c being the speed of light. Then the particle's rest mass in eV/c^2 is

$$m_0 = \frac{p}{\beta\gamma}. \tag{2.5}$$

Chapter 3: A Kaon Trigger for FOPI

As explained in the previous chapters, FOPI's goal, the investigation of strangeness production and propagation (chapter 1.1.1), is constrained mainly by the performance of the detector system (chapter 2.3.2). Even after the electronics' upgrade, the wish for a further improvement of the statistics for strange particles remains. On the other hand, only few out of the events that FOPI collects are useful for the later analysis. The majority of recorded event data does not contain any strangeness. A kaon trigger is a means to at least minimize the waste of time caused by such events. It supplies the capability to skip uninteresting events at a very early stage of the data acquisition. This chapter describes, what this should mean in detail, and how this issue can be solved for the FOPI experiment.

3.1 High-Level Trigger Systems

Experiments in high energy physics (HEP) typically produce large amounts of data at a high frequency, thus making extraordinary high demands on the data acquisition and storage system. In many cases, the demands exceed the financial or even technical possibilities. However, only a small fraction of this data, data from only a part of the detector, or only selected events are necessary for the physical analysis. A trigger can reduce the data rates by identifying the relevant information.

3.1.1 Trigger Idea

While a simple trigger system is capable of determining e.g. the start time or the particle multiplicity of an event in order to exclude multiple or peripheral events (like the FOPI start counter and first level trigger, see chapters 2.2.3 and 2.3.1), a high-level trigger system (HLT) can perform a much more detailed, albeit in many cases compared to off-line analysis still simple, analysis of the event's physical properties. It is possible to make a HLT e.g. identify certain particle species or coincidences. It depends on the given experiment, whether then a considerably reduced amount of data per event is passed to the storage devices, or whether the information only of events selected by the trigger criterion is further processed by the DAQ chain.

In the case of strangeness experiments at FOPI, a higher yield of strange particles is desired (see chapters 1.4 and 2.3.2). Since only a small fraction of the measured events contain any strangeness, the task of a high-level kaon trigger is to avoid the data acquisition for the remaining events or at least to abort it at an early stage. It must therefore accomplish a particle identification using an adequate subset of the detector data, and supply a result within less than the targeted event frequency.

Particle identification can be done with algorithms similar to the ones used for the later detailed physics analysis. Yet their complexity may be considerably lower, because the only result of interest is the one given by the trigger criterion, which is the event's strangeness content in the case of FOPI's kaon trigger. Due to the typically large number of detector channels and high event rates, a specialized hardware platform, commonly allowing parallel or distributed processing, is usually necessary for a trigger system to provide a low latency.

3.1.2 Tracking Approaches

For a detailed physical event analysis, the produced particles and their properties have to be identified. This can be done, if the momenta and velocities of the particles are known. They can be calculated from the associated trajectories' radii and time-of-flight (see chapter 2.4). While the latter can directly be read from the respective detector (like FOPI's GRPC barrel), the trajectories have to be reconstructed from the data of a tracking detector (like the FOPI CDC) using an appropriate feature extraction algorithm. Various different algorithms exist, that can be used in high energy physics tracking applications, and many of them have already successfully been applied. They can be divided into two major classes: local and global approaches.

Local methods start with a few points, the track initialization. The algorithm tries to find other points that are appropriate to spatially extend the initial tracklet by following the rules that define a valid particle track (the track model). If more of them exist continuously, they complete the initialization to form a track candidate; if not, the tracklet is discarded. Typical members of this group are:

- The Track Following method, which tries to extend a track “seed” (usually two or three hits in a low occupancy region) by including hits that fall into a window given by the orientation of the seed and the track model.
- The Track Road method, which starts from both ends of a possible particle track. Additional hits are included, if they meet a road of a certain width around the track candidate, that is compatible with the track model and the initial hits.
- Track Segment methods are an extension to the track following method. Elements that are added to the initial seed are also clusters of few hits and likewise have an orientation.
- The track following method can be implemented using a Kalman Filter [Kalm60, Früh87]. This approach recursively refines the extracted track parameter by alternately predicting the path of the particle and correcting the prediction by a new measurement, i.e. by assigning an additional hit, that is compatible with the prediction, to the particle track. The final result is identical to the least-square's fit to the hits that form the particle track. Yet in every recursion step, only the coordinates of the current hit have to be processed.

The disadvantage of all local methods consists in that most of the detector hits must be touched multiple times, many of them for each attempt to complete a particle track. Moreover, with increasing hit multiplicity, the number of futile cycles grows faster than linearly.

Global methods do not compute single points or groups of points individually. Instead, all points undergo the same treatment, whose first step often is a parameter transformation. This transformation produces a list of track candidates, or at least a structure, in which tracks can be found much more easily than in the original detector image. Important global methods are:

- The Combinatorial method. All combinations of hits that form a possible particle track are tested against the track model. The track is accepted, if it is rated as good according to criteria defined in the model. This method can only be applied to detector images with only a few hits, because the computing effort grows extremely fast with the detector occupancy.
- Pattern Matching compares the detector image to a list of patterns, that correspond to possible particle tracks. Each match indicates the existence of the respective track. This approach is only viable, if the number of possible particle tracks is sufficiently low to handle the pattern list. It is also difficult to implement, if the data is noisy, and especially if detector hits are missing.
- The Histogramming method is an improvement over the pattern matching approach. A histogram features a counter for each possible particle track, and a lookup table provides the information, which detector hits are associated with that track. The histogram is filled by incrementing the respective counters for each hit. Local maxima in the histogram denote the particle tracks, which are contained in the detector image with the highest probability. The height of a maximum corresponds to the quality of the recognized track. This method is robust against a certain level of noise and missing data.
- Methods based on the Hough Transform [Houg62, Nogu86] are a special case of a histogramming method. The histogram represents a discrete parametrization of all possible particle tracks and can be filled by a parameter transformation of the detector image. The Hough transform method is introduced in detail later (see section 3.3.1).
- In the Conformal Mapping [Hans88, Yepe96] approach, the detector hits are transformed into a space, where circular particle tracks become almost straight lines or parabolas. The following track recognition can be done by a comparatively simple and fast algorithm.
- Algorithms utilizing an Artificial Neural Network [Gyul91, Kola95]. A typical neural network of the “feed-forward” type consists of an appropriate number of nodes arranged in layers: an input layer, mostly one or two “hidden” layers and an output layer, which for classification problems can be represented by a single node. The nodes of each layer feed their outputs to one or more nodes of the subsequent layer. A node's output is determined only by the sum of its weighed inputs minus a threshold. The network is trained with real or realistic data, i.e. the weights and thresholds are adjusted to produce the correct classification for the set of input patterns. The network is subsequently capable of correctly classifying similar input patterns.

A comprehensive introduction to tracking algorithms is given in [Bock90]. See [Best92, Pink95, Klug97] for a short description of some of these algorithms and examples of their application.

3.1.3 Hardware Platforms

HLT systems usually work on the same level as or within a detector's data acquisition, which is responsible for data reduction. However, the HLT has a more sophisticated task, which requires an understanding of the data. Event classification and particle identification, which typically starts with a track reconstruction, is the basis for a significantly more efficient data reduction. These tasks require powerful computing platforms, and the specific HLT demands make special or even application specific hardware components necessary.

Critical boundary conditions, which limit the choice of an appropriate hardware platform, are, beside the complexity of the trigger task, the required input bandwidth, which is given by the amount of detector data necessary to compute the trigger algorithm, and the event rate. Commercial off-the-shelf (COTS) hardware is limited by e.g. the speed of the PCI bus, which is below $133\,MB/s$ or, for high end systems with a $64\,bit/66\,MHz$ bus, still below $0.5\,GB/s$. Available I/O technology is even slower: the data rate of e.g. Gigabit Ethernet may further be reduced by protocol latencies. The event rate also sets the boundary condition for the computing power of the trigger hardware. Aspects of data transfers, like memory bandwidths and latencies, have also to be taken into account. In some cases, the DAQ chain must be halted during the processing of the trigger algorithm. This circumstance dictates a maximum latency of the trigger result, which further increases the required computing power and input bandwidth.

Some of the algorithms described in the previous section are inherently parallel, or have at least steps, that can easily be executed in parallel. The first is true e.g. for an artificial neural network, while the second holds for the treatment of the different track candidates in local algorithms, as well as for the coordinate transformations in global approaches. A dedicated hardware platform will naturally exploit as many parallelization opportunities as possible.

Suitable hardware may be based on DSPs, that feature dedicated functional units for certain operations like FFT or matrix operations. Reconfigurable FPGAs (see section 3.4.2 for more details) can be programmed to represent any digital circuit that is allowed by the amount of existing logic units. In mass production, ASICs may be an additional option. Especially for neural network algorithms dedicated chips, like CNAPS® [Adap94] or TOTEM [Anze95], and whole systems, like the neuro-computer SYNAPSE-1 [Rama95], have been developed. Recently, with the availability of fast data switches and cluster software, processor farms, that consist of a large number of standard PCs, can be used in high performance computing.

3.1.4 Existing High-Level Triggers

Some examples of recent trigger developments are briefly addressed in the following. These particular systems have been chosen for two reasons: first, they represent a broad spectrum of the hardware platforms and algorithms discussed above, and second, they are similar in central aspects to the situation in FOPI: all example systems reconstruct particles from the data of a tracking detector in order to select events or regions-of-interest for permanent storage.

- The ATLAS experiment [ATLA94], that is currently being installed at the future Large Hadron Collider (LHC) facilities [Schm02] at the European Organization for Nuclear Research (CERN) in Geneva, Switzerland, will employ a high level trigger [ATLA02] for online event selection. In the level-2 trigger (LVL2) part of the HLT, the event rate has to be reduced from the LVL1 rate of $100\,kHz$ down to the LVL2 accept rate of $3\,kHz$. The LVL2 requests a few percent of the detector data, that belong to regions of interest (RoI) defined by the LVL1 trigger, for analysis. The LVL2 processing units (L2PU) are projected as a farm of standard PCs connected by a fast switching network.

One of the ATLAS trigger scenarios is the complete reconstruction of B-physics events (events that contain traces of B meson decays), for which a tracking algorithm [Sess98, Hink99] for the full TRT subdetector [ATLA97] has been contributed by the Mannheim ATLAS group. It is based on a Hough transform and has also been optimized [Hink00] for implementation on an FPGA based hardware platform (see section 3.4.3). The input event rate for the full scan TRT algorithm is around $9\,kHz$. Performance results are presented in [Bain00], recent timing measurements are reported in [Hink03].

- ALICE [ALIC96] is the second of the four large-scale experiments projected for the LHC at CERN, and is dedicated to the analysis of heavy ion collisions. The TPC [ALIC00] is the largest tracking subdetector, that produces roughly 90% of the data. It delivers an event rate of $200\,Hz$ for central $Pb+Pb$ collisions. Other subdetectors, like the TRD [ALIC01] and the ITS [ALIC99] can be used for further refinement of the tracking results. The ALICE third level trigger [ALIC02] has to select physically relevant events or RoIs for permanent storage. A Hough transform is executed for tracking within single detector slices, inter-detector tracking is done using a Kalman filter. The HLT is designed as a farm of COTS dual-processor PCs that are connected via a fast network technology. An additional PCI card (RORC) provides, among others, FPGA coprocessor functionality.
- STAR [Acke03] is a heavy ion experiment, that is running at the Relativistic Heavy Ion Collider (RHIC) [Hahn03] at the Brookhaven National Laboratory (BNL) in Upton, NY, USA. The third level trigger [Adle03] for the tracking subdetectors like the main instrument TPC [Ande03] can deal with a $50\,Hz$ event rate for central $Au+Au$ collisions. Its sector part (SL3) is associated with TPC sectors and performs cluster finding. The results are passed to a tracking node, which employs a conformal mapping method for event reconstruction. The reconstruction efficiency exceeds 80% (75% for $p_T \approx 1\,GeV/c$ particles), the accuracy is $\Delta p_T / p_T \approx 10\%$. A global part (GL3) then collects the data and makes the trigger decision. The HLT hardware consists of a dedicated CPU farm (custom boards carrying an Intel® i960® processor) for the cluster finder connected by a fast network (Myrinet®). The tracking is done by Compaq® DS-10 and DS-12 (466 and $600\,MHz$ ALPHA® CPU) machines running Linux.
- The H1 experiment [HONE97, HONE97b] at the electron-proton collider HERA at DESY in Hamburg, Germany, uses a multi-level trigger architecture to reduce the event rate from the bunch crossing rate down to only $5\,Hz$ taping rate. Its second level trigger [Köhn97] makes a decision based on the outputs of the numerous L1 subtriggers, that deliver a rough event classification for the different H1 subdetectors. For a part of the L2 calculations, an artificial neural network based on CNAPS® chips is used.

  The H1 trigger system has been upgraded to allow a more sophisticated event selection based on a track analysis. The Fast Track Trigger (FTT) [Bair01] reconstructs up to 48 three-dimensional tracks per event in the central drift chamber in three stages: the first stage finds lower resolution track segments by a histogramming method, the second stage links the refined segments and performs a fit, and finally the third stage extracts the physics quantities. The FTT runs on a custom computing board [Meer02], that carries an Altera® FPGA with embedded Content Addressable Memory (CAM) which is used for the tracking, and DSPs for the fitting routines.

These examples point out approaches, that might be viable also for FOPI. There are, however, reasons, why an individual solution must nevertheless be developed. The tracking algorithm has to take into account the unique geometry of the CDC. The desired extension of the particle recognition up to higher momenta (around $2\,GeV/c$) requires an additional matching with the TOF data from the GRPC barrel for better accuracy. Moreover, an increase of the event rate in order to enhance FOPI's strangeness yields cannot be done after a complete CDC readout, because the corresponding event rate is already achieved in FOPI without a trigger. The CDC readout itself turns out to be the bottleneck in this case. Last but not least, the integration of a solution similar to the systems introduced above would require a redesign of substantial parts of FOPI's data acquisition, which is simply beyond the financial possibilities. Thus, a customized system appropriate for strangeness enhancement in FOPI is developed in the subsequent sections.

3.2 System Specification

Before developing a high level trigger system for the FOPI experiment, dedicated to the enhancement of its strangeness yield, the present conditions of the data acquisition and the trigger's implications on the overall experiment's performance have to be examined closely. This includes the definition of the data, that must at least be made available to the trigger, and the data that it technically can be supplied with. It further requires a careful consideration of the influence the trigger performance exerts on the data collectible during a FOPI experiment. Trigger performance in this context can be associated with quantities like timing, recognition efficiency, and fake rejection.

3.2.1 Timing Constraints

In order to have the statistics on some strange particle species in events of a certain type – defined by the reaction partners, the impact energy, and the collision centrality – improved by a factor E, the following is a rough estimation of the minimum timing requirements the trigger module must meet.

Without a trigger, the DAQ is able to collect N events within the time

$$T_{DAQ} = N\,T\,, \tag{3.1}$$

where T ("big time") is the average DAQ time for a single event. It is presumed, that the data from exactly one of these events contains the signatures of a kaon, which implies, that N depends on the event type. If a trigger is employed, the time the system needs for the same number of events must be recalculated. Assuming that the trigger decision time is simply added to the DAQ time, one can compute as follows: Let g ("good") be the probability for detecting an existing kaon with the trigger algorithm, and b ("bad") the probability for erroneously detecting a kaon which is not contained in the data. Let t ("small time") be the time the trigger needs to compute the decision for a single event. Five different types of events can then be distinguished and are summarized in fig. 3.1 and table 3.1.

According to this classification, the time that is needed to collect the same N events with the enabled trigger system sums up to

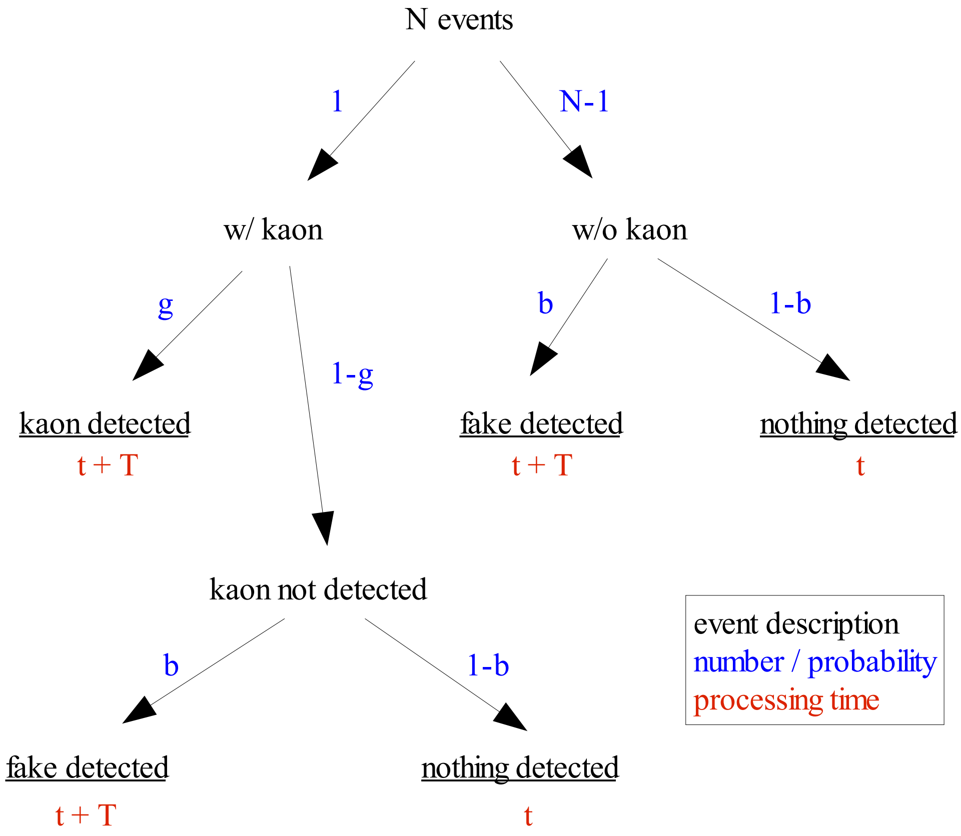


Fig. 3.1: Classification of different event types. The distinction is done with respect to kaon contents within the event and its detection.

No.	contains kaon	kaon detected	fake detected	# events	time
1	yes	yes	–	g	$t+T$
2	yes	no	yes	$(1-g)b$	$t+T$
3	yes	no	no	$(1-g)(1-b)$	t
4	no	–	yes	$(N-1)b$	$t+T$
5	no	–	no	$(N-1)(1-b)$	t

Table 3.1: Summary of the different event types (cf. Fig. 3.1) with frequency of occurrence and necessary processing time.

$$\begin{aligned} T_{TRIG} &= [g+(1-g)\,b](t+T)+(1-g)(1-b)\,t+(N-1)[b\,(t+T)+(1-b)\,t] \\ &= Nt\;+\;[g+(N-g)\,b]T\,. \end{aligned} \tag{3.2}$$

The number of kaons, that can be found in these events with trigger, is reduced to

$$N_K \;=\; g \;+\; (1-g)\,b. \tag{3.3}$$

Thus, for the same number of kaons as without trigger, the time

$$T_K = \frac{T_{TRIG}}{N_K} \tag{3.4}$$

is needed. The kaon enhancement factor then is

$$E = \frac{T_{DAQ}}{T_K} = \frac{N_K T_{DAQ}}{T_{TRIG}} = \frac{N[g+(1-g)\,b]\,T}{N\,t+[g+(N-g)\,b]\,T}. \tag{3.5}$$

Solving for the allowed trigger computing time *t* leads to

$$t = \left(\frac{g+(1-g)\,b}{E} - \frac{g+(N-g)\,b}{N}\right) T\,. \tag{3.6}$$

The condition $t > 0$ gives an upper boundary for the kaon enhancement factor

$$E < N \frac{g+(1-g)\,b}{g+(N-g)\,b}. \tag{3.7}$$

Using the optimistic assumptions $g \gg b$ and $N\,b \gg 1$, the above terms reduce to

$$t \approx \left(\frac{g}{E} - b\right) T \quad \text{and} \quad E < \frac{g}{b}. \tag{3.8}$$

Events of the type $\mathrm{Ni}+\mathrm{Ni}@1.93\,AGeV$ have a probability of approx. $1/5{,}000$ for containing a negatively charged kaon within the acceptance of the CDC [Herr04]. So in the above formulas $N=5{,}000$ for these particles. As a first guess, *g* and *b* may be $80\,\%$ and $3\,\%$, respectively. The DAQ time for these events can be estimated as $T=1.5\,ms$ [Leif04] (cf. chapter 2.3.2). To achieve an enhancement factor of $E>8$, the trigger algorithm must accordingly be faster than $t=100\,\mu s$ for an average event.

In the case of events of type $Au+Au@1.5\,AGeV$, assuming $N=1{,}400$ [Herr04], $b=10\,\%$, and $T=2\,ms$, a processing time of $t=100\,\mu s$ is sufficient for an enhancement of the anti-kaon contents by a factor of $E>5$.

3.2.2 Prerequisites

Unfortunately, it is not possible to compute a trigger decision from the data that was just read from the CDC electronics. This is due to the slow FADC readout, which alone is responsible for a latency of approx. $T_{SCAN}=0.8\,ms$ (see chapter 2.3.2). With this constraint the size of the event database may be reduced, but its total strangeness contents were then diminished as well.

However, some circumstances are helpful to nevertheless reach the goal.

- Particle identification only requires a particle track's radius and its time-of-flight, so a small amount of the total event data may already be sufficient to produce useful trigger decisions. The complete data also contains information about e.g. pulse heights and energy loss, which are redundant due to the available TOF data.

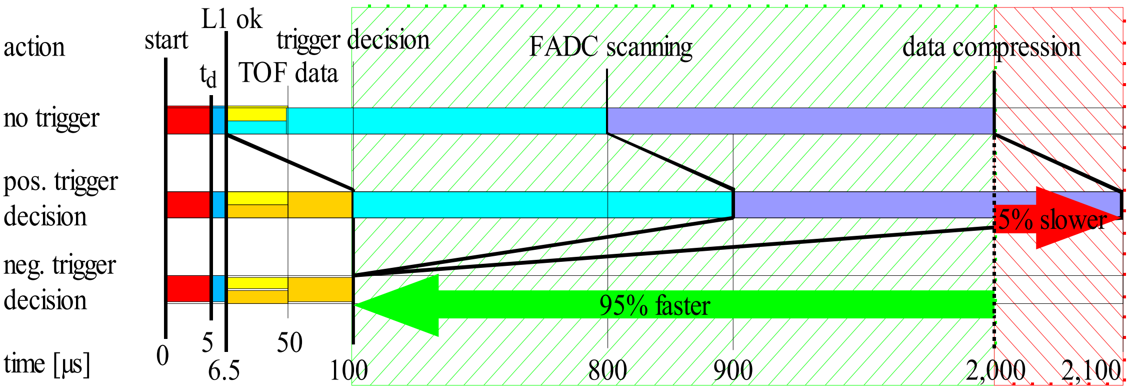


Fig. 3.2: Timing diagram for the DAQ of a complete event, without and with the use of a trigger. Differences arise only after the CDC drift and the ok from the level-1 trigger. While a positive trigger decision increases the total time by only 5%, the time needed for events rejected by the trigger is reduced by roughly 95%. The time axis is not to scale for better visibility.

- The CDC's FADC units supply a logical output with a time-to-hit signal (chapter 2.3.3). This signal contains only a small fraction of the total data, but the data is available within only $t_d = 5\,\mu s$ after the event, and, if biased carefully, contains the complete track information for all particles.
- $t = 100\,\mu s$ are sufficient for 300,000 clock cycles on a $3\,GHz$ standard CPU, or 10,000 clock cycles on a specialized parallel computing machine running at $100\,MHz$. For the time being, this seems to leave enough room for doing the computing necessary for particle identification. This issue will be addressed again later (cf. chapter 6).

The resulting timing for the DAQ of a complete event is shown in fig. 3.2 for the current situation, and for the scenario with a trigger in comparison. An additional latency of $100\,\mu\,s$ is introduced by the trigger after the drift time t_d in the CDC and the signal from the level-1 trigger. However, for events, that are rejected by the kaon trigger, the complete time period for the time consuming CDC readout and the data compression is bypassed, thus saving roughly $95\,\%$ of the total event time.

3.2.3 Trigger Concept

Since requirements and constraints are now known, it is possible to schematically outline the trigger concept. This issue covers data input, employed hardware, and algorithm partitioning.

- Though the CDC signals are easily and quickly available at the FADCs' comparator outputs, connecting them to the trigger is not trivial. Due to the fact that up to 960 signals have to be transmitted, there are in principle only two possibilities of managing this task: One could either collect a certain number of signals locally using a multiplexer, and connect the multiplexers to the trigger. Yet this requires these modules (and maybe also additional de-multiplexing in the trigger) to be developed and installed, while on the other hand it only reduces the amount of cabling directly at the trigger, and the total number of cables stays at least the same. The other possibility is to directly connect each FADC front electronics output to the trigger via a separate cable, thus saving the multiplexers but complicating matters at the trigger module in terms of the available space.

- Whichever alternative is finally chosen might in addition require that only a certain fraction of the available signals is used. At the same time, the high input bandwidth (number of signals times time resolution) requires specialized hardware, at least for the preprocessing of the data. The input bandwidth from the CDC calculates as follows:

  $$D_{CDC} = n f = n \frac{v_d}{s}, \tag{3.9}$$

  where n is the number of signals, f the sampling frequency for the comparator signals, v_d the drift velocity in the CDC and s the spatial resolution of the hit recognition, that can be achieved with the sampling frequency. Using $n=960$, $v_d=45\,\mu m/ns$, $s=2.25\,mm$, and a maximum drift time of $t_d=5\,\mu s$, one finds $f=20\,MHz$, $D_{CDC}=2.4\,GB/s$, and finally $M_{CDC}=D_{CDC}\,t_d=12\,kB$ as the amount of data to be read from the CDC. Considering only signals from every second sense wire still delivers $D_{CDC}=1.2\,GB/s$ and $M_{CDC}=6\,kB$. Even Gigabit Ethernet links are by far not capable of data rates in this region. Turning the multiplexers mentioned above into readout buffers could on the one hand reduce this data rate. Yet on the other hand t_d were increased by the same factor, so that the available computing time t were possibly reduced.

- The detector image has a size of $6\,kB$ representing 96,000 different pixels, because mirror hits have to be included. Approx. 3,600 of them are active in an average, i.e. semi-central, $Au+Au@1.5\,AGeV$ event. In the first part of the particle identification, the track recognition, the image has to be compared to a set of e.g. 100,000 patterns (when using a global approach, see chapter 2.4). Even if detector symmetries are exploited, and a sophisticated tracking method is applied, perhaps 120 comparisons per hit plus one per pattern may still remain. But this would already result in more than 500,000 operations for all active hits. From the above results it is clear that this can only be done either by a massively parallelized algorithm on an appropriate machine, or by many machines working in parallel.

  The next part of the particle identification matches a few track candidates (maybe around 80) to a few TOF records. However, these calculations involve floating point operations and are therefore well suited for multi-purpose hardware like CPUs or DSPs.

To put everything together, FOPI's kaon trigger will be composed of a specialized parallel computing machine for data acquisition and track recognition, and a machine carrying a CPU or DSP for the floating point rich matching between track and TOF data.

3.3 Tracking Algorithm

The particle recognition mainly consists of two parts: a track recognition and a matching between track and TOF parameters (chapter 2.4). The latter can be done by a comparatively simple analysis: the extrapolation of tracks and search for TOF records within a given region around the extrapolated tracks. The first part, however, is a by far more sophisticated task: track parameters have to be extracted from an image, which possibly contains incomplete data, and in addition noise at the same time.

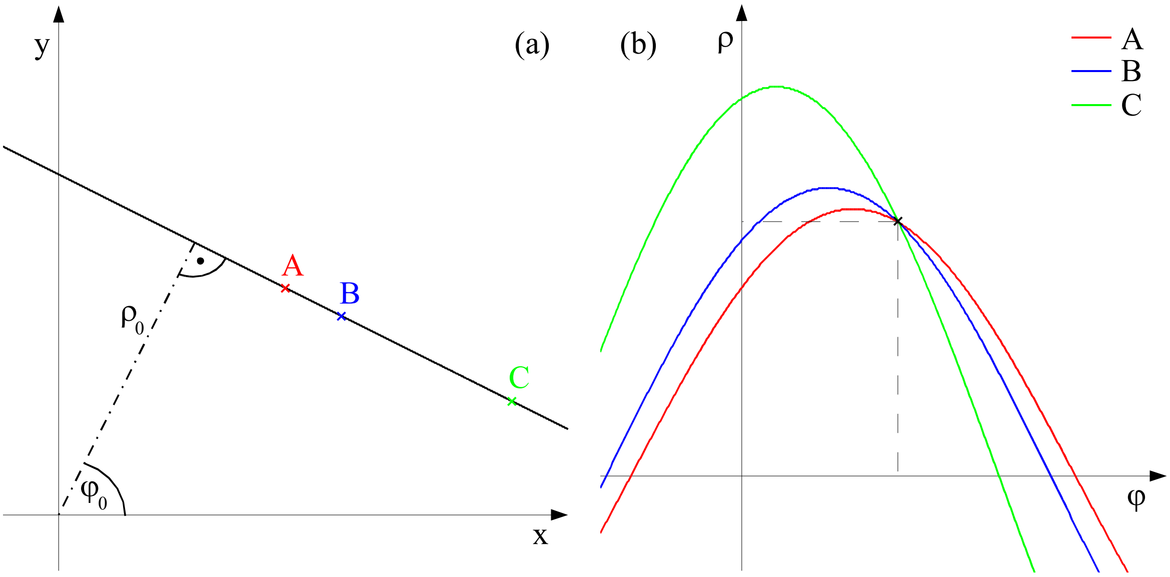


Fig. 3.3: Hough transform for a straight line in (ρ, φ) parametrization. Single points of the object (a) result in curves in Hough space (b). Their point of intersection represents the parameters (ρ0, φ0) of the original object.

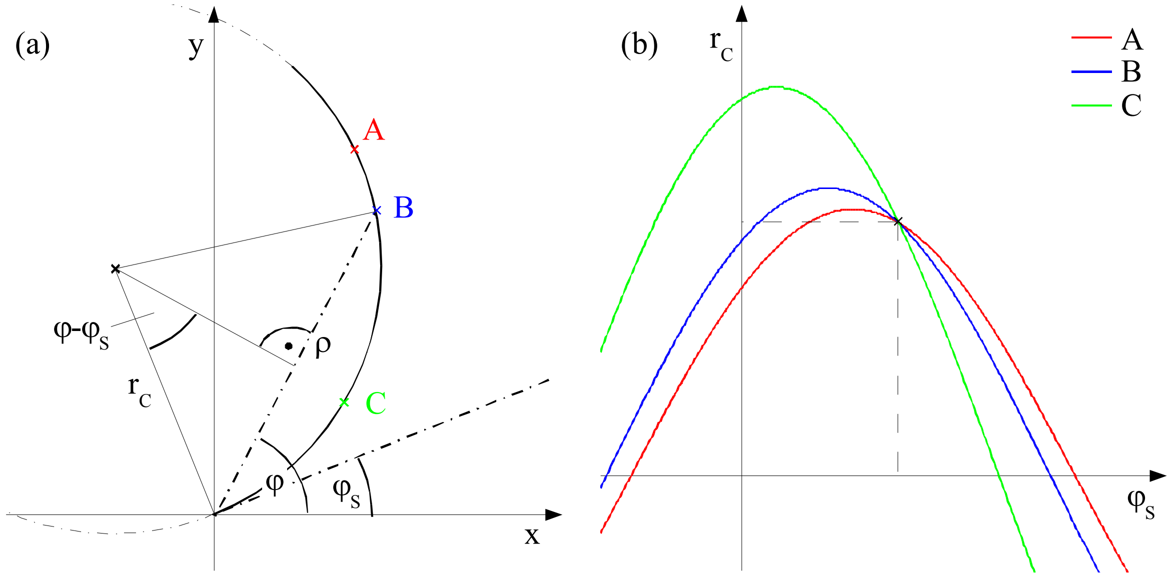


Fig. 3.4: A circle segment (a) in (rC, φS) parametrization and its Hough transform (b).

3.3.1 Hough Transform

The Hough transform [Houg62] was developed by P.V.C. Hough and granted a patent to IBM Corp. in 1962. An early application to a high energy physics tracking task is e.g. described in [Perl86]. The Hough transform finds the parametrization of an object by transforming the pixels of an image into the parameter space for that object. To each pixel of the image is assigned a curve in the parameter space. This curve represents all parameters of objects that the pixel could possibly belong to. The point of intersection of all these curves then gives the parameters of the sought object in the image. This, of course, only works for objects that are known and have a common parametrization. Other objects can not be found with the Hough transform.

The original and simplest application of the Hough transform is the detection of straight lines. In an image with coordinates x and y a straight line can be represented by its slope a, and b, the value at $x=0$, via the equation $y=ax+b$. However, this does not work for lines parallel to the y axis, so for the Hough transform usually the alternative representation

$$x\cos\varphi \;+\; y\sin\varphi \;=\; \rho \tag{3.10}$$

is used instead, where φ is the angle between the line's normal through the origin and the x axis, and ρ is the line's distance from the origin along this normal. In this representation, each image point in (x, y) space yields a curve in (ρ, φ) space. The coordinates of the point of intersection of curves originating from points on the same straight line then give the parametrization of this line (see fig. 3.3 for an illustration).

In the case of particle tracks in a magnetic field, circle segments (i.e., projections of helices) rather than straight lines must be searched for. As the particle tracks originate from the center-of-collision, the circles to be detected intersect the origin. Such a circle can be represented by φ_S, the starting angle of the track, which is the angle between the positive x axis and a tangent to the circle at the origin, and r_C, the track's radius, via

$$\frac{\rho}{2r_C} \;=\; \sin\left(\varphi - \varphi_S\right), \tag{3.11}$$

with $\rho=(x^2+y^2)^{-1/2}$ and $\varphi=\tan^{-1}(y/x)$ being the pixels' coordinates in polar representation. The Hough transform for circle segments is illustrated in fig. 3.4. The direction of the curvature (left / right) can be expressed by different signs of ρ.

3.3.2 Hough Map

To practically process the Hough transform, the parameter space can be quantized, and each point in this space is then assigned to a counter. Every time the parameter curve of a transformed point passes a point in Hough space, the corresponding counter is incremented. The pattern histogram that arises from this transformation is often referred to as the Hough map. Rather than calculating the points of intersection of curves the second step of the Hough transform turns into the detection of local maxima of the emerging peaks in the Hough map. An example of a Hough map that contains the peak of a transformed object is shown in fig. 3.5.

If the original image contains many objects, parameter curves do not only intersect at points representing the parametrization of objects in the image. In addition, curves belonging to points from different objects intersect at numerous and unpredictable locations. To exclude these accidental matches from being identified as objects, only peaks in the Hough map that exceed a certain threshold are collected. Thus this minimum peak height defines a minimum number of points objects in the original image must consist of. An object may, but needs not have to, be composed of more points, and will be detected in all these cases. Objects consisting of less points are discarded.

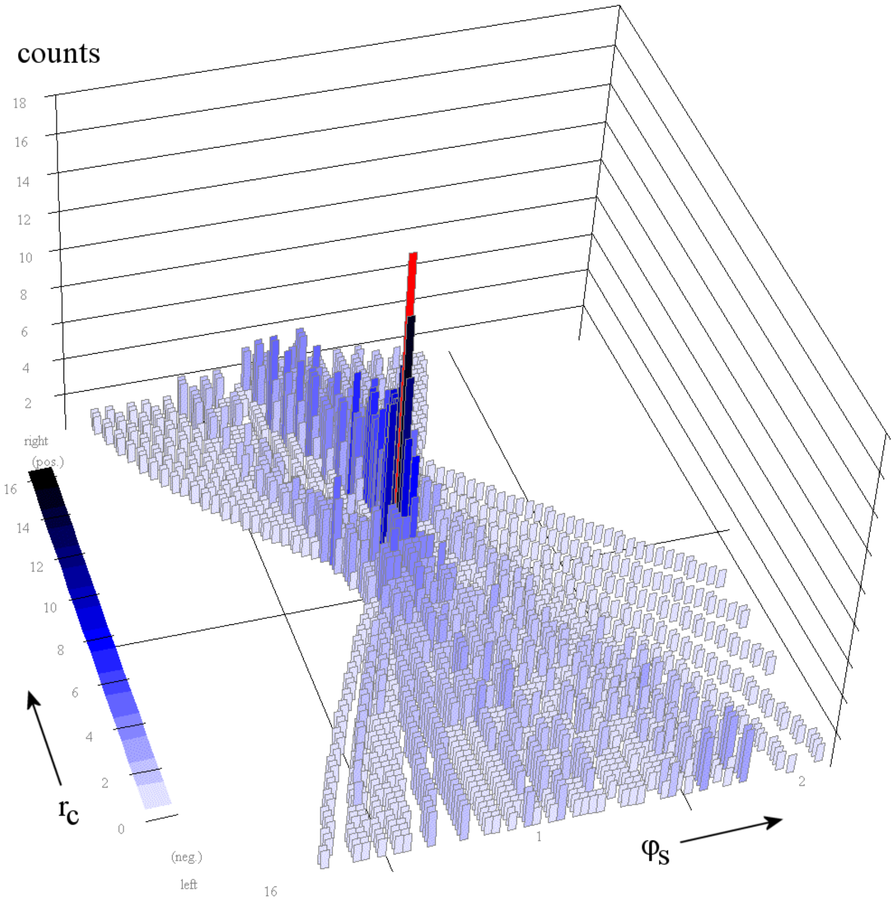


Fig. 3.5: A peak in the Hough map that arises from a single track in the CDC. Histogram counters are ordered by φ_S value (left to right) and r_C (front to back). Their heights and color shadings correspond to the number of accumulated hit transformations. The maximum (red) can clearly be identified; the enhancement of the background is caused by contributions from the mirror track.

The granularity of the quantization must be chosen such that different objects lead to well separable peaks. Further accuracy of the detected object parameterizations can be achieved by either using finer quantizations, or by fitting the parameters to the points in the original image that contributed to them. The granularity is often chosen such that the size of the Hough map is equal to the size of the images [Jähn01].

3.3.3 Lookup Table

If, as is the case in a trigger algorithm, the Hough transform is to be applied to many images, it is useful to store the transformations from all points into the Hough map in a large table. This saves most of the computing during runtime, especially if the Hough transform itself becomes more complicated. The lookup table, however, requires a lot of memory, because in general it will have to store one value for each pixel of the image and for each possible value of one of the parameters, e.g. for r_C, thus resulting in

$$N_{LUT} = N_{\rho} \cdot N_{\varphi} \cdot N_{r_c} \quad (3.12)$$

entries. In the above example, assuming $N_{r_c}=100$, this already leads to $N_{LUT}=4.8\cdot 10^6$. If it is further assumed, that a full byte is necessary to encode the values to be stored in the lookup table, namely $129\leq N_{\varphi_s}\leq 256$, the size of the lookup table results to $4.8\,MB$. In times, where PCs' main memories easily exceed $1\,GB$, this may not sound like a problem. However, if a fast random access must be provided, standard SD-RAM is due to the required wait cycles quickly found beyond its limits. The alternatives are expensive SRAM modules and other cache memories, for which $4.8\,MB$ are still a challenge.

The lookup table defines, which patterns (represented by the counters in the Hough map) a pixel in the image possibly belongs to. If, however, the transformation is reversed, which basically means that the lookup table is sorted by histogram counter rather than by image pixel, it contains a description of the patterns with a list of their corresponding pixels instead. Thus the Hough transform can be viewed as a reverse pattern matching algorithm.

3.3.4 Algorithm Discussion

The Hough transform is a powerful tool, that has been used for image analysis and pattern recognition purposes for many years. Reasons, why a tracking approach based on the Hough transform was also chosen for this work, are briefly discussed in the following:

- Due to the expected complexity of the track finding and the narrow timing constraints estimated above, it is necessary to have an algorithm that can very well be parallelized. This is true for both steps of the Hough transform [Sess00]. The parallelization issues are discussed in detail in chapter 6.1.
- The algorithm must be robust against noise and missing data. The Hough transform is known to satisfy these requirements better than most other approaches. The allowed level of noise and the acceptable fraction of missing features can effectively be balanced by the height of the histogram threshold.
- The Hough transform can be implemented in such a way, that it only invokes basic operations, e.g. read and write accesses to memory, counters, or comparisons. All more advanced computations can be shifted towards the calculation of the lookup table (which is only done once) or are part of the last step, where only a few track candidates remain. This property is important for a parallel hardware implementation, which thus turns particularly simple and effective on special hardware platforms. See section 3.4.2 for an example.
- The FOPI collaboration successfully uses the Hough transform for event analysis [Best92], and the Mannheim FPGA group has experiences with Hough transform based trigger algorithms [Sess00] and their hardware implementation [Hink03]. This assures, that a trigger based on a Hough transform can in principle work.

3.4 Trigger Hardware

Due to the timing constraints and the high input bandwidth a proper selection of the hardware platform is of great importance. In this section the requirements are discussed, and an appropriate solution is presented.

3.4.1 Requirements

Basically, the trigger algorithm requires three modules. They may not necessarily be implemented on the same machine, but already clearly limit the choice among existing technologies.

- The CDC data must be collected by a module which can supply several hundred input channels. Conventional inputs are only capable of managing a few (e.g. 8, 16, 32,...) parallel connections. When thinking instead of a fast serial interface, additional multiplexing is required in order to to convert the data, and in addition, even the speed of Gigabit Ethernet is not sufficient. Moreover, the data input module must either do the preprocessing of the acquired signals including zero suppression, or it must have a fast private connection to the Hough module, because the expected data rates are far beyond the capabilities of standard solutions like PCI (up to $133\,MB/s$) or even PCI $64\,bit/66\,MHz$ (up to approx. $0.5\,GB/s$). Thus, a custom hardware platform is necessary for the data input.
- The Hough module has to transform the CDC hits into the Hough map and discover the positions of local maxima. Therefore, it has to store the histogram counters as well as the lookup table. In order to be able to process the transformation within the available time, either a special memory access with a very high data bit width or an ultra-fast storage medium is required for the lookup table (confer also section 3.3.3) and the histogram. In addition, the memory must provide a fast random access, because the addresses of the data and the counters are determined by the CDC hits and can therefore not be read sequentially. Standard memory solutions provide a data width of e.g. $64\,bit$ (SD-RAM DIMM), so multiple modules could be combined and accessed in parallel, or the high speed of e.g. DDR-SD-RAM could be exploited. However, random accesses to SD-RAM suffer from wait states, which slow down these accesses substantially. SRAM technology, which is used for fast cache memories, may be a better choice, because it offers a reasonable data rate of $133\,MB/s$ that is not slowed down by random accesses. The high throughput might also require to treat lookup table and the Hough map independently, so that also two independent RAM buses need to be supplied.
- As was already mentioned, particle identification requires the combination of tracking and TOF data. In principle, this can be done by using a second lookup table. However, only a few track candidates need to be extrapolated, and the TOF data is relatively small. The calculation invokes a few floating point operations and comparisons, which can very quickly be executed on a standard processor featuring an ALU, so a CPU or a DSP is a reasonable alternative for this module.

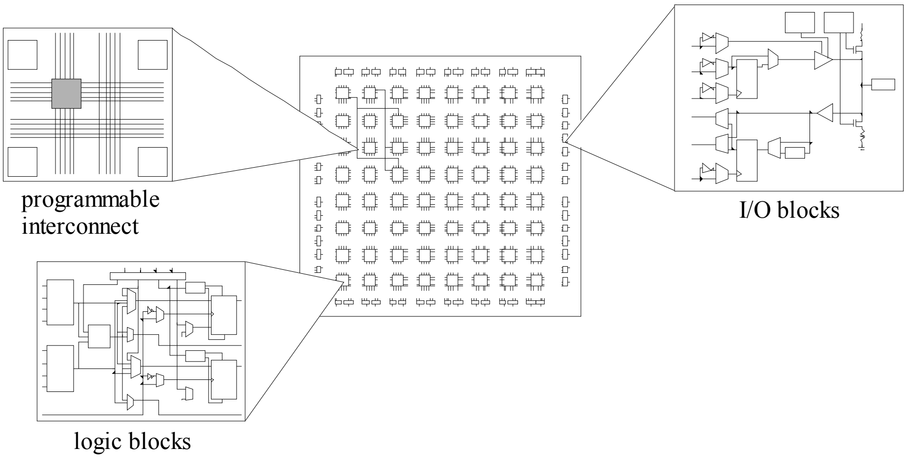


Fig. 3.6: Schematic layout of a typical FPGA device [Xili97]. Configurable logic blocks are connected via configurable routing resources. I/O blocks provide the connection to the outside world.

3.4.2 FPGA Processors

A Field Programmable Gate Array (FPGA) is a configurable pool of elementary logic resources. Some of its basic cells can represent an arbitrary function of a few binary inputs. In addition, simple arithmetic units and storage cells are available. These basic resources are combined to Configurable Logic Blocks (CLB). The CLBs as well as the connections between them are freely configurable by the user. The block diagram of a typical FPGA is shown in fig. 3.6. It is possible to implement any digital circuit, ranging from gates and counters via FIFOs and memory to complex devices, e.g. controllers or even CPUs, on an FPGA. Limiting factors are only the amount of available logic and routing resources, and I/O pins. Because of their flexibility, FPGAs can carry out different operations simultaneously and independently, so they are an excellent platform for parallel computing.

Many of currently available FPGAs are equipped with additional features. In particular, the embedded memory blocks of modern Xilinx® Virtex™ II devices [Xili04] are useful for the presented trigger application. They provide many fast (also in random accesses) and independent memory blocks in direct neighborhood of the processing resources. They can thus store Hough maps or lookup tables without absorbing logic resources or I/O paths.

Therefore, FPGA processors are a promising platform for the implementation of a Hough transform based pattern recognition. They perfectly meet the requirements for the computing and I/O intensive processing modules outlined above.

3.4.3 MPRACE

FPGA based computing machines have been developed for many years at Mannheim University. The Hough Board [Klef93] and the Enable Machine [Klef92] were large arrays of FPGAs, which multiplied the capabilities of a single FPGA. The next generation, Enable++ [Högl95], introduced a modular concept with different boards optimized for either computing intensive or I/O oriented tasks. ATLANTIS [Sing00, Bros00] continued this concept, but with much fewer and larger FPGAs, which led to a much better usability. A different approach was

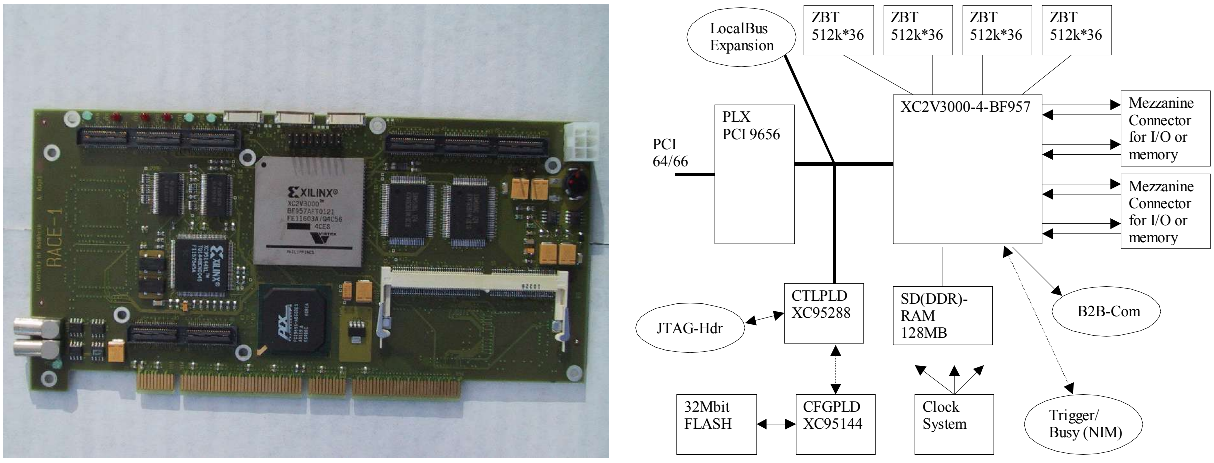


Fig. 3.7: Photograph (left) and block diagram (right) [Kuge01] of the MPRACE FPGA coprocessor. The most important functional units are explained in the text.

introduced with microEnable [Högl97], which became a commercially available product of a spin-off company, Silicon Software GmbH. microEnable is a simple and easy-to-use coprocessor, that features a single FPGA, several I/O connectors and additional memory on a PCI card.

The latest development is called MPRACE (Multi-Purpose Reconfigurable Architecture / Computing Engine) [Kuge01] and combines the flexibility and power of ATLANTIS with the simplicity of microEnable. Fig. 3.7 shows a photograph and the block diagram of MPRACE. Features of this coprocessor are:

- PCI interface for fast communication with the host system with 64 *bit* at 66 *MHz*. The board is also compatible with PCI (32 *bit* at 33 *MHz*). A PLX® PCI 9656 bridge chip connects the PCI bus to the local bus (32 *bit* at 66 *MHz*). The resulting data bandwidth is thus 266 MB/s.
- Xilinx® Virtex™ II FPGA. The basic configuration uses a Virtex II-3000 FPGA in a BF957 package (XC2V3000-4BF957C). This FPGA features:

  14,336 slices. Each slice contains two logical units, so more than 28,000 basic operations can be executed in parallel. Designs (i.e. circuits represented by the configured FPGA) are known to run at 125 *MHz* and more [Müll04b], even over 200 *MHz* depending on the configuration.

  1.7 *Mbit* of internal RAM, organized in 96 blocks. Each of these blocks can be configured e.g. to store 2,048 values of 9 *bit* and to be used simultaneously for read and write accesses (“dual-ported”).

  684 configurable I/O pins.

  96 hard-wired floating point multipliers.
- Four banks of ZBT-SRAM, each 36 *bit* wide (32 plus 4 extra bits) and 512 *k* deep, resulting in a total of 8 *MByte* (ignoring the extra bits). ZBT-SRAM (Zero Bus Turnaround) needs no wait cycles when switching from read to write access and vice versa. Only three cycles at

133 *MHz* are needed to access ZBT-SRAM, and subsequent random accesses can be done at each clock cycle.

- One SD-RAM connector in S0-DIMM (notebook memory) format.
- Four expansion connectors, each 36 *bit* wide, providing 144 independent channels for data input. The connectors may also be used to attach a mezzanine board with additional ZBT-SRAM, that can be addressed as a single bank (512 *k* times 144 *bit*). The data width of the resulting total memory outside the FPGA is then 288 *bit*.
- Two high-speed serial board-to-board interfaces (one per direction), that can run at typically 266 *MHz*.

In addition to the MPRACE hardware, a software environment for the communication between host applications and the coprocessor, and for the configuration of the hardware is available. It consists of a device driver for the PCI bridge, pciDriver [Müll04b], the uelib [Müll04, Hink04], and the local bus interface. The uelib is a C++ library that is responsible for the recognition, initialization and configuration of FPGA based coprocessors. Moreover, it allows the access to the interfaces of the designs loaded into the FPGA via single read / write cycles or DMA. The local bus interface is the matching counterpart inside the FPGA.

3.4.4 Hardware Discussion

MPRACE offers everything required for data input and tracking. It perfectly exploits and supplements the FPGA features to provide a suitable platform for the trigger algorithm. Up to 144 channels per board are available for the input of the detector data, so only a few boards are necessary to fulfill this task. MPRACE features fast on-board and pluggable SRAM memory with a data width of up to 288 *bit*, that can be used for the lookup table, and the FPGA internal block memory, where the track histogram can be stored. The FPGA's logic resources can be used for the implementation of the LUT based transformation, and for the maximum search. MPRACE is a PCI based coprocessor; the host PC on the other side of the PCI bus contains a CPU and thus represents the ideal completion of the hardware required for the kaon trigger (see section 3.4.1).

If a different hardware platform was chosen instead, e.g. based on PCs connected by a fast network technology, at least the problem of the input of the CDC's sense wire data would remain unsolved. Thus, a custom solution, possibly comprising multi-channel single bit ADCs and multiplexers, would have to be developed anyway. Moreover, the memory speed problem (cf. section 3.3.3) and latencies introduced by the network technology might increase the number of required computing nodes in order to keep their load sufficiently low. MPRACE avoids or solves all these shortcomings immediately, and is thus chosen as the hardware platform for FOPI's kaon trigger. The details about the hardware implementation of the trigger algorithm on MPRACE are discussed in chapter 6.

Chapter 4: Algorithm Implementation

Having learned how a trigger algorithm for FOPI can in principle work and what requirements with respect to efficiency and timing it must meet, the implementation of this algorithm is the next step. In addition, several variations and improvements have been introduced in order to investigate their behavior and to improve performance. The software implementation for evaluation, simulation and performance measurements has been done in C++ using the MFC class library (see appendix B for details) for the user interface. The central functions in the tracking algorithm have been designed in such a way, that a porting to an FPGA based implementation and thus parallelization of time-critical operations (see chapter 6.1) can easily be done. This chapter describes all relevant details of the different modules and their implementation.

4.1 Lookup Tables

As it was suggested in chapter 3.3.3, the implementation is based on a Hough transform using the lookup table approach to minimize computing during runtime. In this case, the design of the lookup tables is the most important part of the implementation, because all relevant details of the Hough transform, including detector geometry, pattern granularity and all optimizations, are shifted to the calculation the lookup tables.

4.1.1 Track Symmetries

Though the Hough transform in principle works as it was described earlier (see chapters 3.3.1 through 3.3.3), nothing has been said about pattern distribution nor granularity. Regarding the pattern distribution, two symmetries come into play: the symmetries of the tracks and the symmetries of the drift chamber. Thus, the quantization of one of the histogram dimensions, the starting angle φ_S of the track, is straightforward, because this parameter has point symmetry with respect to the origin, the center-of-collision (see fig. 4.1). So the different entries of the Hough map will naturally be chosen to refer to equally spaced starting angles. Moreover, the 16-fold point symmetry of the CDC's sense wire layout can be used to reduce the size of the lookup table that transforms points in real space into the Hough map: a complete lookup table would repeat its contents (only shifted by a fixed angle) 16 times.

The second parameter, the track's radius r_C, is only symmetric with respect to the direction of its curvature, and all tracks with the same direction but different r_C are mutually different in shape. Moreover, the track radius has no correspondence in the drift chamber wiring, so no symmetries can be used here. Therefore, for the implementation of the tracking algorithm, three different quantization approaches were developed and investigated, that each aim at a

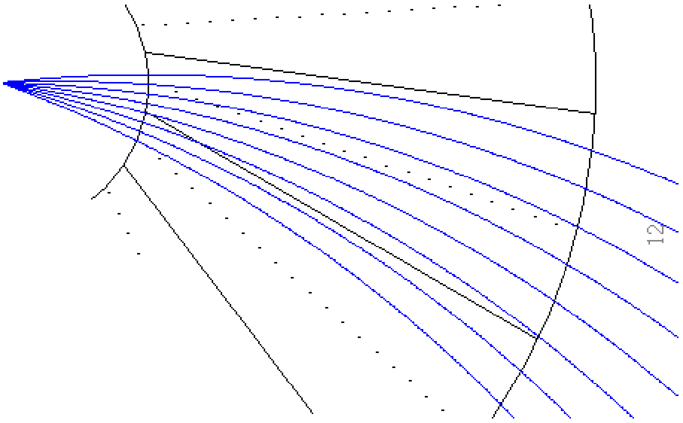


Fig. 4.1: Some tracks with equal radius, but different starting angles. Their shapes are identical; the tracks are rotated around the center by multiples of a fixed angle.

different aspect of the particle reconstruction. Each of them can be combined with any histogram granularity.

4.1.2 Peak Shape Optimization

An important issue regarding pattern distribution is the distribution of starting angles covered by the tracks that a given hit possibly belongs to. For a certain hit in the CDC, one can construct a track with any radius r_C, so that the hit is part of the track. On the other hand, the resulting starting angles φ_S can cover only a part of the full angular range. However, this part changes significantly with the hit's distance from the origin, ρ, and in particular it becomes considerably large for large distances. Moreover, all Hough curves cross the Hough map in the same diagonal direction (see fig. 4.2 for the illustration of both effects). The resulting peak from all curves of all hits of one track is not round, but diagonally distorted (fig. 4.3). The solution to both problems is to replace the starting angle parameter φ_S by the track's angle φ_a at an intermediate distance R_a from the origin, i.e. $R_{CDC}^{in} < R_a < R_{CDC}^{out}$, e.g. $R_a = 60\,cm$. The Hough curves from hits closer than R_a to the origin then cross the histogram in the opposite direction, the resulting peaks become more symmetric, and the maximum φ_a coverage of a single hit and thus of the complete peak's background is much smaller than the maximum φ_S coverage.

The Hough transform for circle segments (see chapter 3.3.1),

$$\varphi_S = \varphi - \sin^{-1}\left(\frac{\rho}{2r_C}\right), \tag{4.1}$$

also changes, if φ_S is replaced by the parameter φ_a. The new transformation rule is constructed as follows (see fig. 4.4 for illustration): Let α be the angle between the lines from the origin to where circles with radii ρ and R_a around the origin intersect the track. Furthermore, ψ is the angle between the first line and a line to the center of the circle that completes the track. Then from the two relations

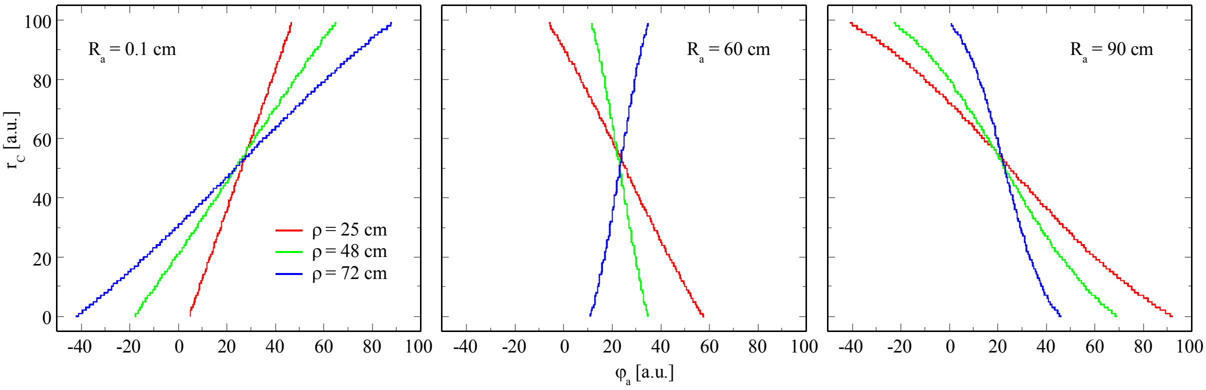


Fig. 4.2: Hough transform for three hits of a particle track with different ρ, using different values of R_a. If R_a is shifted towards the center (left) or beyond the outer boundary of the CDC (right), the transformed lines become asymmetric and spread over a wide range of $φ_a$ values.

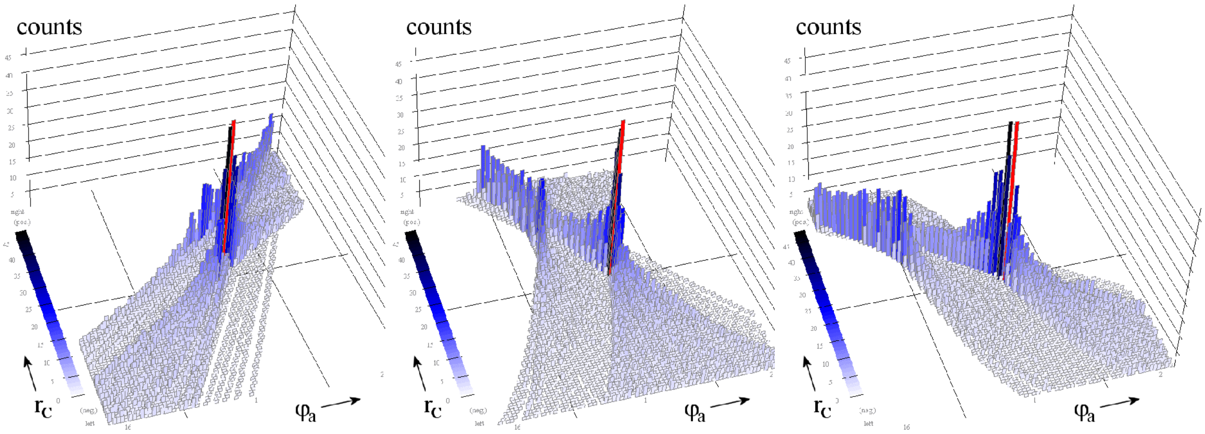


Fig. 4.3: Hough map (see fig. 3.5) with the entries of a complete particle track (including mirror hits), using the same values for R_a as in Fig. 4.2.

$$\cos\psi \;=\; \frac{\rho}{2r_C} \quad \text{and} \quad \cos(\psi+\alpha) \;=\; \frac{R_a}{2r_C} \tag{4.2}$$

follows that

$$\alpha \;=\; \cos^{-1}\left(\frac{R_a}{2r_C}\right) - \cos^{-1}\left(\frac{\rho}{2r_C}\right), \tag{4.3}$$

and with $\varphi_a = \varphi + \alpha$ the new transformation reads

$$\varphi_a \;=\; \varphi + \cos^{-1}\left(\frac{R_a}{2r_C}\right) - \cos^{-1}\left(\frac{\rho}{2r_C}\right). \tag{4.4}$$

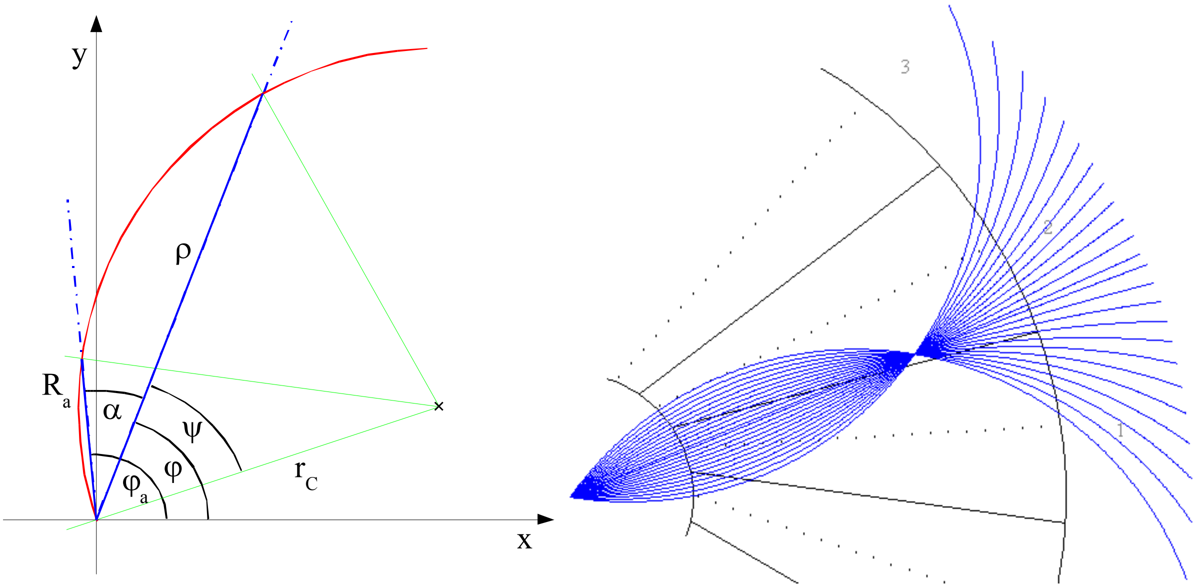


Fig. 4.4: Left panel: Construction of the Hough transform for a circle segment, using the intermediate angle φ_a instead of the starting angle φ_s. Right panel: Some tracks with equal φ_a and different r_C, distributed according to the inverse transverse momentum approach; high momentum tracks exhibit a slightly denser CDC coverage.

4.1.3 Inverse Transverse Momentum Approach

It is obvious, that choosing an equidistant distribution of radius values r_C is not reasonable, because one would then end up with many almost straight tracks (with different radii but almost equal shape) on the one side of the spectrum, and only a few strongly curved tracks on the other. It is rather straightforward to choose the tracks' curvatures as the distinguishing feature instead, i.e. to define search patterns equally spaced in curvature. Since the curvature is the reciprocal of the radius, which in turn is proportional to the particle's transverse momentum (component of the momentum perpendicular to the beam axis, see chapter 2.4), this approach can be labeled the inverse transverse momentum approach. The reason for why this approach is discussed first is that it has already been used before [Sess00], and that it is simple to compute.

In order to determine the different curvatures C_n of the search patterns, tracks with positive and negative curvatures are treated separately. For each half a minimum value $\pm C_{min}$ represents an almost straight track (to avoid infinite radii), and $\pm C_{max}$ is the maximum reasonable value, i.e. that represents the smallest radius which still allows a particle to hit the TOF detector ($1/C_{max} > R_{TOF}/2$). Now both intervals,

$$\left[-C_{max}, -C_{min}\right] \quad \text{and} \quad \left[C_{min}, C_{max}\right], \tag{4.5}$$

are each split into $N_C/2-1$ equal parts, and are thus limited by N_C equidistant values altogether. The distance between two neighboring values within one interval is always

$$\Delta = \frac{2}{N_C - 2} \left(C_{max} - C_{min} \right), \tag{4.6}$$

and the spacing between the two intervals is $2C_{min}$. Hence,

$$C_{\pm n} = \pm C_{min} \pm n\Delta. \tag{4.7}$$

The corresponding track radii (see fig. 4.4) can later be recovered using the relation

$$r_n = \frac{1}{C_n}. \tag{4.8}$$

4.1.4 TOF Matching Approach

The identification of tracks is only the first part of the tracking algorithm, and the tracking is completed by the matching of track candidates to the available TOF data. It is thus straightforward to adjust the possible accuracy of the pattern recognition output to the desired tolerance of the matching. This means, that all track patterns, whose parameters are defined in the Hough map, have to be equally spaced with respect to their points of intersection with the TOF detector shell. For subsequent patterns in φ_a direction this is already the case, because the neighboring φ_a values are always equally spaced. Yet for the tracks' radii r_C, additional effort must be made.

The parameter C now represents the path length along the TOF detector between the projection of a track and a straight track with the same angle φ_a. It is equal to

$$C = \alpha R_{TOF}, \tag{4.9}$$

where α is the angle introduced in section 4.1.2, but now with $\rho = R_{TOF}$. So it is immediately obvious, that C_{min} and C_{max} can be derived from the equation

$$C = R_{TOF} \left[\cos^{-1}\left(\frac{R_a}{2r_C} \right) - \cos^{-1}\left(\frac{R_{TOF}}{2r_C} \right) \right]. \tag{4.10}$$

Again, the different C_n are obtained by adding multiples of Δ to C_{min} or $-C_{max}$.

To get the tracks' radii back from the C_n, a different approach can be used (see fig. 4.5). Let a be the length of the connection between the ends of the lines, that start at the origin and intersect the track at $\rho = R_a$ and at $\rho = R_{TOF}$. From the law of cosine follows that

$$a = \sqrt{R_{TOF}^2 + R_a^2 - 2R_{TOF}R_a\cos\alpha}. \tag{4.11}$$

Now the particle track represents a part of this triangle's circumcircle, and r_n is its circumradius, for which the relation

$$2r_n = \frac{a}{\sin\alpha} \tag{4.12}$$

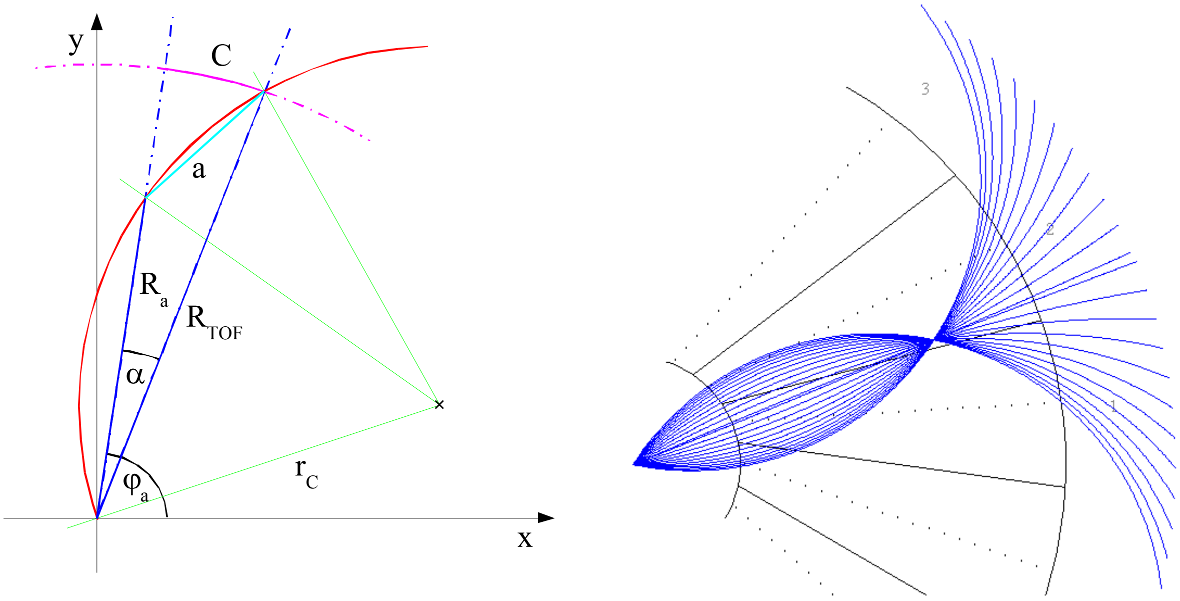


Fig. 4.5: Left panel: Construction for a quantization using the TOF matching approach. Right panel: Some tracks with equal φ φ_a *and different* r_C*, distributed according to the TOF matching approach; low momentum tracks exhibit a significantly denser CDC coverage.*

is valid. Putting everything together, one obtains

$$r_n = \frac{\sqrt{R_{TOF}^2 + R_a^2 - 2\,R_{TOF}\,R_a \cos\left(\dfrac{C_n}{R_{TOF}}\right)}}{2\sin\left(\dfrac{C_n}{R_{TOF}}\right)}. \tag{4.13}$$

4.1.5 CDC Coverage Approach

Each of the two preceding approaches has its advantages. However, in the inverse transverse momentum approach, the achievable resolution in detector space for high momentum particles is enhanced. In the TOF matching approach, the resolution for low momentum particles is even strongly enhanced, both in momentum and in space. Therefore, a third approach is introduced as a compromise, which delivers an equal distribution of the particle tracks in space.

Parameter C denotes the largest distance between the track and a straight track with the same angle φ_a at any $\rho < R_a$. This means that the line, along which the distance is measured, crosses the straight track at $\rho = R_a/2$ (see fig. 4.6). The angle between this line and the line from the center of the track's circle to the origin is α, and β are the angles between either of these two lines and the line from the origin to the intersection of the first line with the track. Then

$$\tan\beta = \frac{R_a}{2\,C} \quad \text{and} \quad \sin\alpha = \frac{R_a}{2\,r_C}. \tag{4.14}$$

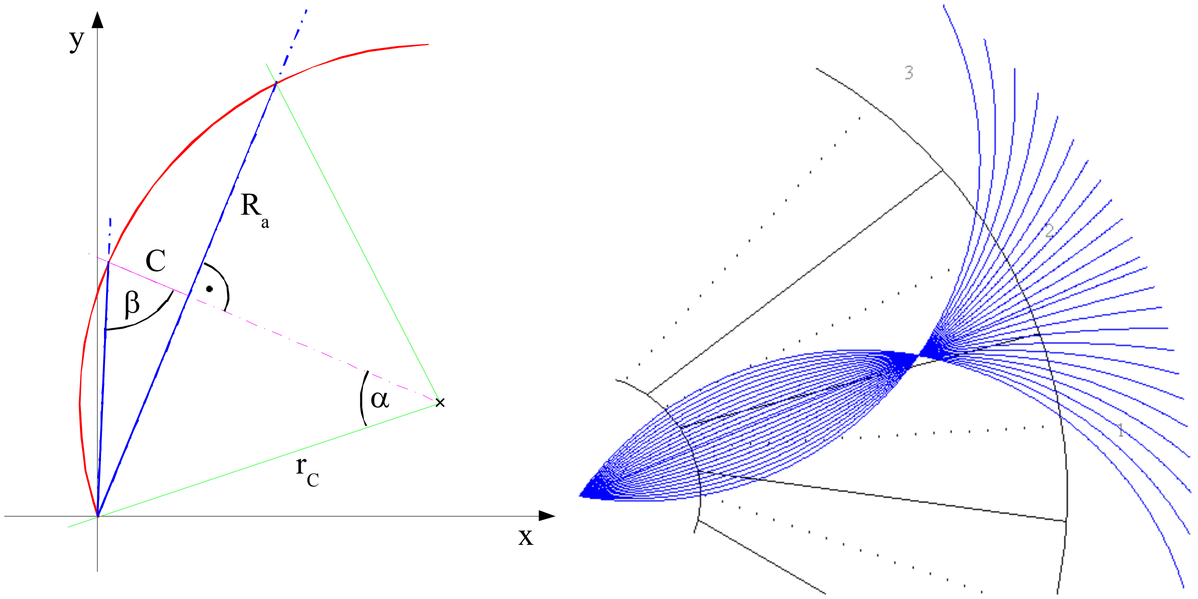


Fig. 4.6: Left panel: Construction for a quantization using the CDC coverage approach. Right panel: Some tracks with equal φ_a *and different* r_C*, distributed according to the CDC coverage approach; the track spacing is equal for all momenta.*

Now α and β are connected via the relation $\beta=(\pi-\alpha)/2$, so solving for C yields

$$C = \frac{R_a}{2\tan\left[\frac{\pi}{2} - \frac{1}{2}\sin^{-1}\left(\frac{R_a}{2r_C}\right)\right]}, \tag{4.15}$$

and the discrete values for the C_n are obtained in analogy to the preceding sections.

Solving the above equations for r_C instead recovers the tracks' radii

$$r_n = \frac{R_a}{2\sin\left[\pi - 2\tan^{-1}\left(\frac{R_a}{2C_n}\right)\right]}. \tag{4.16}$$

4.1.6 Track Width

When tracks are defined and their composition is to be written into the lookup table, a problem arises, if every possible CDC hit is assigned to its best fitting tracks or vice versa: the first case might skip some of the tracks between neighboring hits, while in the latter case not all hits might be referred to by a track.

To avoid both shortcomings, a track width can be introduced. It assures, that no gaps in terms of possible hits on the same sense wire can occur between neighboring tracks in φ_a or r_C direction, and that the width of each track with respect to every sense wire is at least one hit. In addition, the track width can be extended to include the definition of a track overlap. The overlap region extends a track beyond halfway to its neighbor. First of all, this increases the peak heights in the Hough Map. The effects of this feature on the tracking efficiency is part of this work's investigations.

However, with an individually calculated track width or overlap the generation of the histogram becomes more complex, because the number of histogram bins to be incremented for each active hit and individual value of the parameter C is no longer constant. Yet this problem can easily be bypassed by choosing as the track width an appropriate constant in terms of deviation in φ_a.

The described possibilities can be selected as an option for the lookup table calculation:

- A track width of “zero”, which means that exactly one (and no neighboring) histogram bin is assigned to each hit and each r_C value. By this, all hits are covered; but depending on the hit and φ_a granularity, neighboring hits might skip individual tracks, especially in the inner part of the CDC and the vicinity of R_a, where neighboring tracks are close together.
- A track width of “one”, which includes the best matching and the two neighboring histogram bins in φ_a direction. This already reduces the probability of skipped tracks, but still keeps the number of histogram bins to be incremented constant.
- The calculated exact track width, which assigns to a hit for each r_C value the best matching track in φ_a direction and in addition tracks, whose overlap region the hit possibly belongs to. The overlap region can be chosen to vary linearly in size from the inner to the outer boundary of the CDC. If it is chosen to be at least half of the distance (in φ direction) between neighboring hits, no tracks will be skipped between them.

4.1.7 LUT Structure

In the preceding chapters the transformation of hit coordinates into histogram coordinates was described. However, the lookup tables have to supply also the transformation of the detector data, which is available in terms of sector and wire numbers, into hit coordinates ρ and φ. In addition, every detector hit is connected to two transformations, because every hit has a corresponding mirror hit (see chapter 2.2.1). So the lookup table turns into a transformation of array indices

$$\left(wire\,,drift\;time\,,drift\;direction\right)\;\rightarrow\;\left(C\,,\varphi_a\,,\Delta\,\varphi_a\right) \tag{4.17}$$

or, to write it as a program instruction,

$$\left\langle\varphi_a\,,\Delta\,\varphi_a\right\rangle\;=\;\text{lookuptable}\left[wire\,,drift\;time\,,drift\;direction\,,C\right]. \tag{4.18}$$

As sector numbers are not contained in the lookup table, the correct φ_a values are obtained by adding the sector number of the hit times $1/16$ of the total number of φ_a values to the lookup table entries. The transformation must be done twice, once for each possible drift direction. Finally, $\Delta\,\varphi_a$ takes care of the track width (see chapter 4.1.6) by assigning multiple neighboring φ_a values to the transformed hit.

4.1.8 Reverse LUT

It is useful to be able to compare the output of the Hough transform to the original input, e.g. in order to estimate the quality of a recognized track by means of certain criteria, that go beyond the capabilities of the Hough transform (see e.g. section 4.2.7). To increase accuracy, also a fit of the track candidates to the original hit data has been implemented (see section 4.2.8).

For such processing steps a back transformation of the reconstructed trajectories into real space is necessary. This requires a reverse lookup table, which is easily calculated by sorting the lookup table by pattern numbers rather than by hit coordinates. Unlike the original lookup table, which contains φ_a numbers for all combinations of hit coordinates and r_C values, the reverse LUT contains for each histogram bin a list of hit coordinates.

4.2 Processing Modules

While section 4.1 covered the one-time calculations, that can be done before an experiment (or at least after the CDC has been calibrated according to v_d, B and α_L, see chapter 2.2.1), this section describes the individual modules of the tracking algorithm, that have to be executed for every event during the running experiment, from the preparation of the detectors' data to the trigger decision.

4.2.1 Hit Recognition

From the drift chamber, the hit information arrives as a logic signal, which is high whenever the rising edge of a pulse of sufficient height is detected (see chapter 2.3.3). As this signal is available in real-time, the time information of all hits, which are connected to the location of the hits via the drift velocity v_d and the Lorentz angle α_L, can be extracted from it. This is simply done by implementing a counter and incrementing it at a fixed frequency. Whenever the CDC signal switches from low to high, the current counter value together with the corresponding sector and wire numbers is stored in a hit table, or the corresponding bit in a hit lookup table is set to “true”. The hit recognition is illustrated in fig. 4.7.

The counter frequency represents the sampling frequency of the CDC data. Its value influences the size of the lookup table, because the Hough transform for every possible hit is stored there. It is also an adjustable parameter in the trigger algorithm in order to investigate possible effects on the trigger performance.

4.2.2 Hit Filter

Due to external effects like fluctuations of the high voltage, shocks or vibrations, pulses may be induced on the CDC's sense wires, that do not originate from particle ionization tracks. They appear at about the same time on many drift wires, which thanks to the tilted sector boundaries is not possible for a real particle track. Their effect is to introduce a locally enhanced background in the Hough map, which later complicates the search for real local maxima. Fig. 4.8 illustrates this situation.

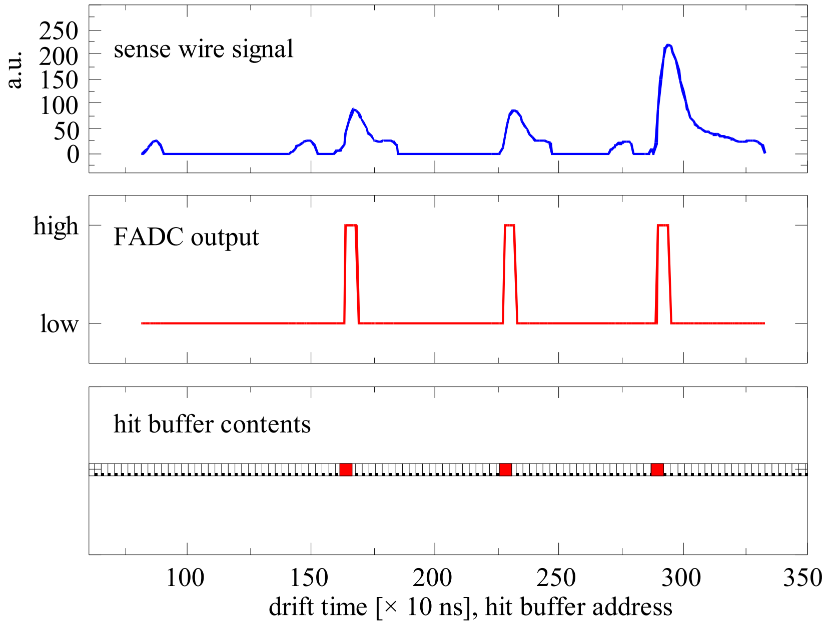


Fig. 4.7: Upper and middle panel: Signal of a sense wire and output signal of the FADC front-end electronics vs. time. Lower panel: Extracted hit information vs. bin number inside the hit buffer. Active entries are shown in red.

These pulses can easily be identified and eliminated. The hit filter deletes pulses, if more than an adjustable number appears at the same time on different wires in the same sector of the CDC.

4.2.3 Histogrammer

The histogrammer is the core of the runtime part of the trigger algorithm. For every active hit, both drift directions and each possible value of C, it increments the counters in the Hough map as suggested by the φ_a values contained in the lookup table.

4.2.4 Histogram Filter

Due to the histogram's background, the search for local maxima requires a sophisticated algorithm, e.g. because some peaks may not reach the height of the background in the neighborhood of others. Alternatively, the histogram can be prepared for simpler algorithms to produce reasonable results. This preprocessing identifies and reduces the background, so in its output the height of a peak becomes a measure for the quality of the underlying track alone (i.e. the number of hits the track is composed of), and the influence of the track's neighborhood (e.g. local detector occupancy) is suppressed.

The histogram filter assumes, that the histogram values change only slightly within the background, but change rapidly in the vicinity of a peak that belongs to a real track. By subtracting now the constant part and leaving the dynamic part (i.e. by applying a high pass filter), the desired behavior can be achieved. Fig. 4.9 shows examples for the results of this filter.

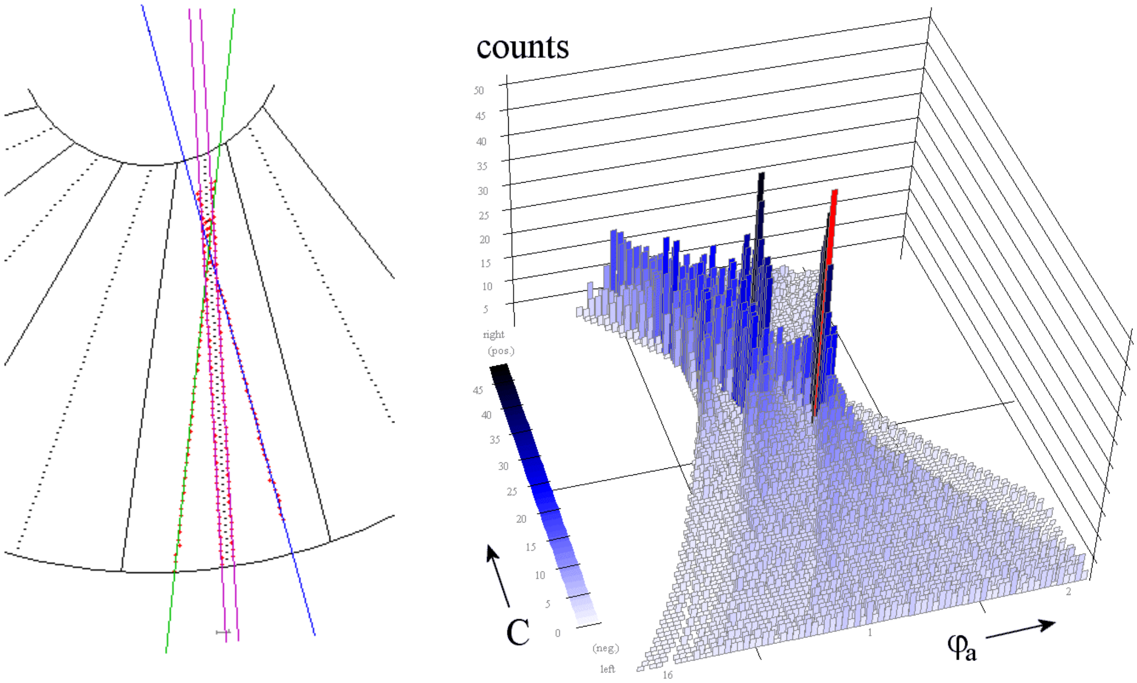


Fig. 4.8: Left panel: Hits caused by a real track (blue) and its mirror track (green), together with a CDC artifact (indicated by violet lines). Only the real track crosses the center. Right panel: Histogram for these tracks. Mirror track and artifact sum up to a second peak of comparable height, which has a different shape and is significantly broader. After elimination of the artifacts, the histogram looks like in fig. 4.3 (middle panel)

Different filters have been implemented and tested, that all work according to the same principle: from each histogram value, the minimum or a weighed average of selected neighboring values is subtracted. At least eight and up to 48 neighboring counters are included. A detailed description of the different filter masks is given in appendix A.3.2.2.

4.2.5 Maximum Finder

In the new filtered histogram, everything below an adjustable global threshold is considered as background. This is equivalent to the application of a dynamic threshold to the unfiltered histogram, if the local value of the threshold is then determined in analogy to the histogram filter described above. After thresholding, the remaining areas are searched for local maxima, whose corresponding parameters represent candidates for detected particles. The definition of what is accepted as a local maximum is shifted to the next section.

The histogram filter (see preceding section) subtracts from each histogram counter a value that is individually calculated for that counter. It is thus likely, that the maxima of some of the peaks are shifted to slightly altered positions. In order to account for this fact, the algorithm offers the possibility to perform the maximum search on the original histogram values instead, while the threshold is still applied to the filtered histogram.

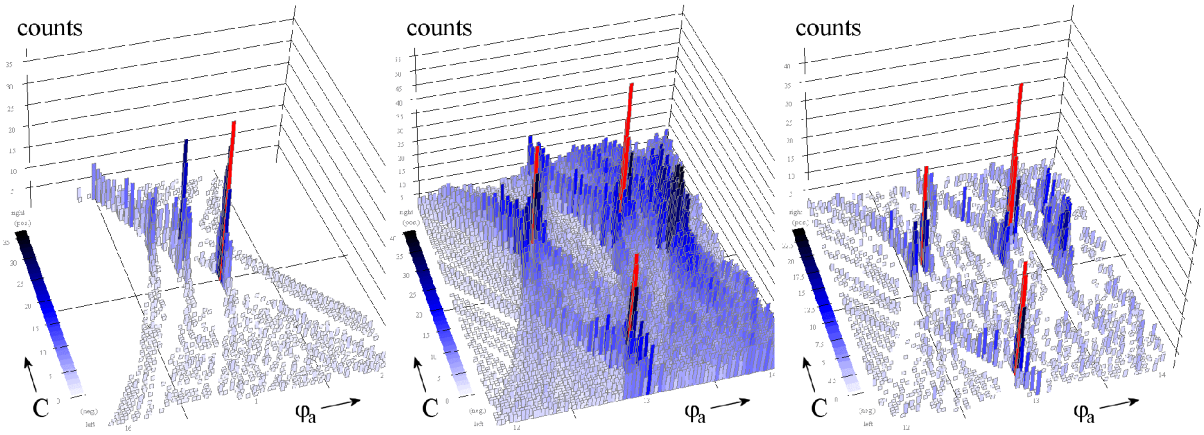


Fig. 4.9: Left panel: Filtered version of the histogram shown in fig. 4.8. Middle and right panel: Unfiltered and filtered versions of the histogram for the tracks shown in fig. 2.5.

4.2.6 Maximum Filter

In general, one considers a value as a local maximum, if all neighboring counter values are less than or in some cases equal to the value at the current position. However, it often occurs, that despite the efforts described in chapter 4.1.2 the shape of a peak is not perfect, so that next-but-one values or values at the at the edge of a peak become local maxima as well.

To avoid every single of these local maxima to be detected, and take only what is likely to represent real track parameters, a maximum filter is applied. The maximum filter defines an area around each counter. Then the corresponding track parameters are only accepted as a track candidate, if the counter value is an absolute maximum within that area.

Again, several filters have been tested, ranging from areas consisting of eight up to 28 neighboring histogram counters. The environments also differ in shape, allowing local maxima to lie closer together in selected directions. The motivation for this is, that the hits of tracks, that differ from each other in both φ_a and C in opposite directions, are (depending on R_a) closer together in the CDC than those from tracks, that differ by the same amount in only one parameter or even in both in the same direction (see fig. 4.10). So the maximum filter can be tuned to include in the neighborhood of local maxima counters that belong to similarly shaped tracks, instead of only similar parameters. The different filter masks are described in appendix A.3.2.4.

4.2.7 Gap Filter

It is possible, that not only hits from a single track, but also from parts of two or more different tracks produce a peak in the Hough map, that is able to pass through all filters. In general there are, however, at least two obvious differences between an erroneously reconstructed and a real track:

- The hits in the CDC oscillate statistically and at small distances around the reconstructed real track. There may be an additional component in the deviation from the track, as the maximum finder might have failed to extract the correct center of the peak, but this component is continuous over the track. This is not the case for accidental matches, because their

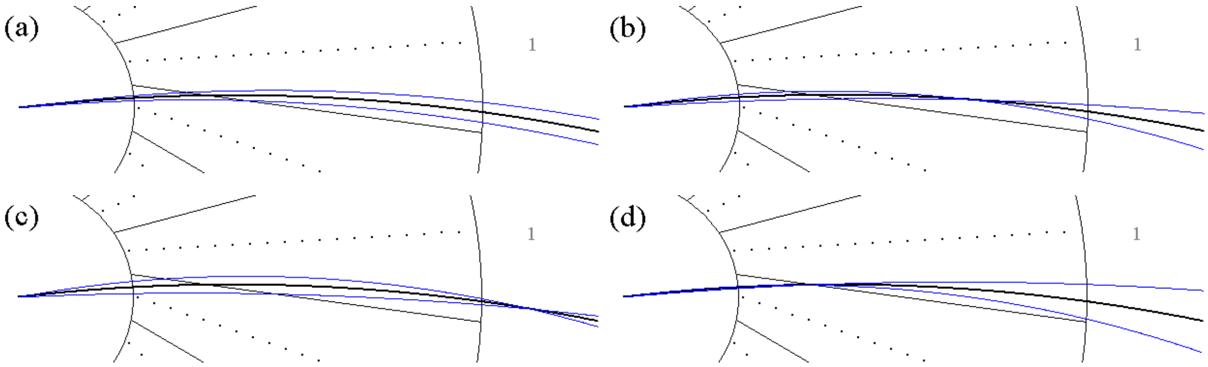


Fig. 4.10: Some tracks with neighboring parameters (a) in φ$_a$ and (b) in C direction, and(c) in both parameters in the same direction and (d) in opposite directions. R$_a$ of 60 cm was chosen for all examples.

hits do not form an exact circle, or at least jumps and kinks can be observed in the deviations of the hit positions from the reconstructed track (see fig. 4.11).

- Accidental matches are often composed of two or more fractions of different real tracks. In this case it it most likely, that an extended region without any hits is situated amidst the track.

If a reconstructed track is a member of the second group, it can easily be identified and deleted: the gap filter removes a track from the list of candidates, if no hit has been assigned to it from more than an adjustable number of successive drift wires. In the ATLAS TRT algorithm [Sess00] a similar gap criterion is used to split erroneously combined tracks.

The gap filter requires the use of the reverse lookup table (see section 4.1.8). As this is rather computing intensive, and as most of the false track candidates do not pass the following matching and particle identification routines anyway, it might turn out to be reasonable to omit this filter in favor of a higher processing speed.

4.2.8 Track Fit

The accuracy with which the parameters of the track candidates can be reconstructed is limited by the granularity of the Hough map. If the maximum finder delivers a neighboring histogram position, the precision of the obtained tracks is consequently even lower.

The solution to this problem is the application of a fitting algorithm, that calculates the ideal track parameters directly from the hits that contributed to the corresponding histogram counter. This again invokes the reverse lookup table to identify those hits.

The employed fitting algorithm has been implemented for demonstration purposes only, and is thus very simple. However, it has proven to converge reliably, albeit slowly. It iteratively calculates a new φ_S from the average of tracks resulting from the old r_C value and the positions (ρ_i, φ_i) of the hits, and afterwards a new r_C that corresponds to the average curvature r_C^{-1} resulting from the current φ_S and the hit positions:

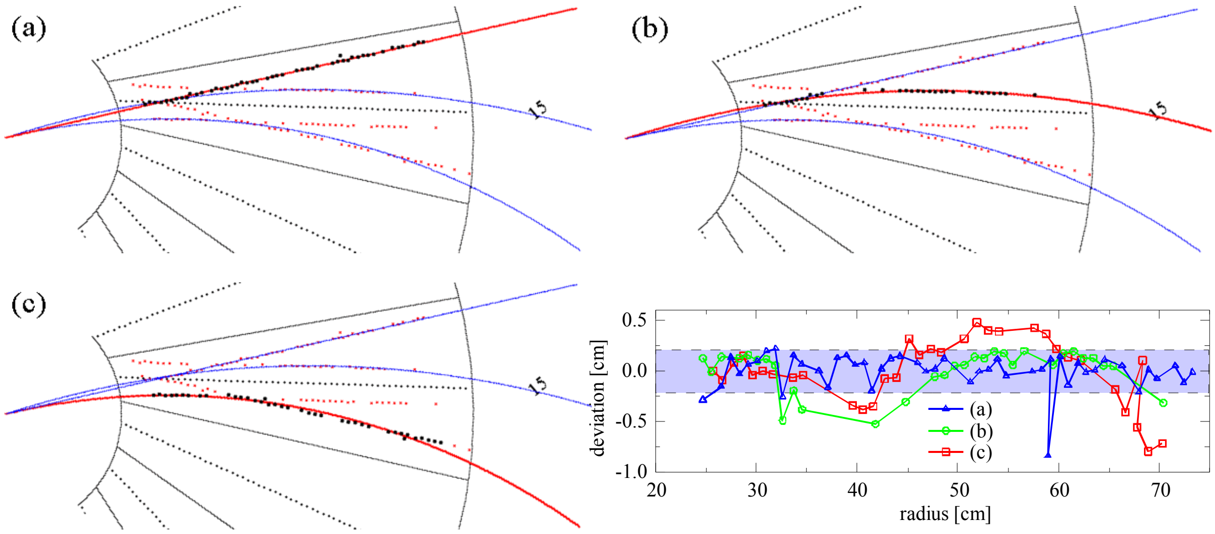


Fig. 4.11: Reconstructed tracks with their corresponding hits: a real track (a) and accidental matches of parts of an artifact with a real track (b) and with a mirror track (c). The lower right panel shows the deviations of the hits in φ direction from the reconstructed tracks: the deviations are evenly low only for the real track (shaded area).

$$
\begin{aligned}
\varphi_S^{(n+1)} &= \frac{1}{N}\sum_{i=1}^{N}\left[\varphi_i + \sin^{-1}\left(\frac{\rho_i}{2\,r_C^{(n)}}\right)\right] \\
\frac{1}{r_C^{(n+1)}} &= \frac{1}{N}\sum_{i=1}^{N}\left[\frac{2}{\rho_i}\sin\left(\varphi_S^{(n+1)} - \varphi_i\right)\right].
\end{aligned}
\tag{4.19}
$$

4.2.9 TOF Matching

The TOF matching, the matching of track and TOF data, assigns to each remaining and improved track candidate the best fitting hit on the TOF detector, if any is considered appropriate. The other tracks are discarded, the remaining TOF hits are discarded as well. This step completes the information that is required to identify particles (see next section).

Track candidates are extrapolated to the location (in the $(\rho\,,\varphi)$ plane only) of the TOF detector using

$$
\varphi_{TOF} = \varphi_S - \sin^{-1}\left(\frac{R_{TOF}}{2\,r_C}\right).
\tag{4.20}
$$

The closest TOF hit within an adjustable region around this point is assigned to it, hence adding a z coordinate (along the beam axis) and the time-of-flight t to the track's properties.

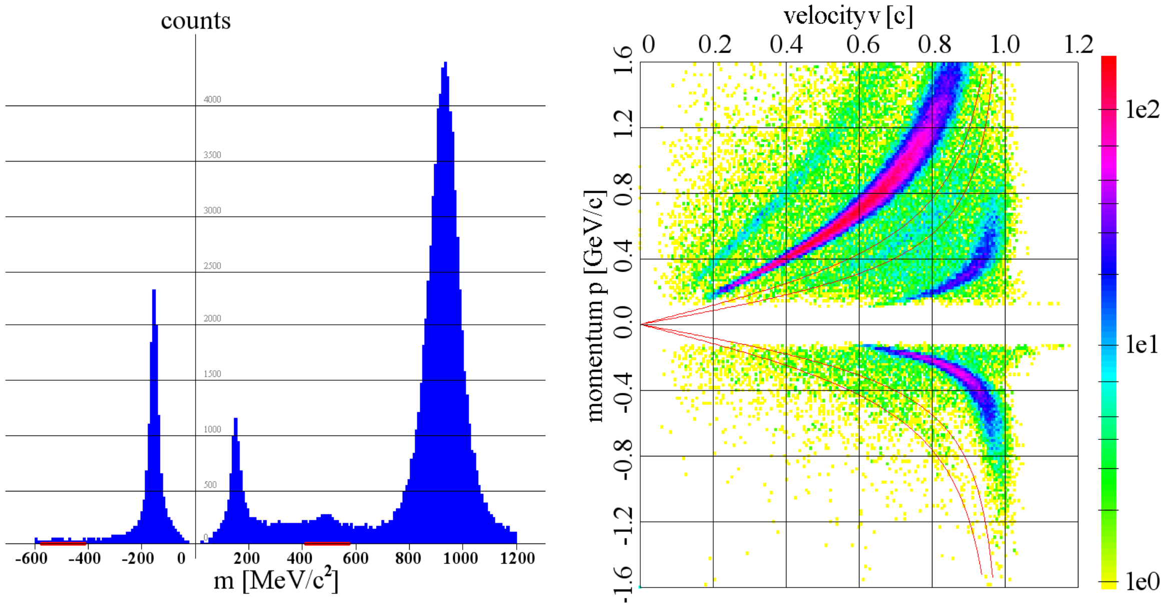


Fig. 4.12: Particle identification using a mass histogram (left panel) and a momentum vs. velocity histogram (right panel). In both examples the area, in which a particle may be identified as a kaon, is marked in red. Negatively charged particles are shown with negative masses and momenta. Both histograms show entries collected in many events. The three largest peaks in the left plot can be associated with the counts for (from left to right) anti-pions, pions, and protons, respectively. The same particles are represented by the three darkest elongated clusters in the right image (from bottom to top). Background entries are mainly caused by false matches between track and TOF data. The corresponding ghost particles may exceed the speed of light (right panel).

4.2.10 Particle Identification

Once all relevant details about the kaon candidates are known, particle identification can be done. From each particle parameter set the particle's rest mass is calculated according to chapter 2.4, until a kaon is detected. A particle is accepted as a kaon, if its rest mass is found within a tunable interval around the kaon mass, 493.677 MeV/c^2. The left panel in fig. 4.12 shows a particle histogram, in which candidates are entered according to their masses. Negatively charged particles are shown with a negative mass for distinction. The windows for the acceptance of kaons and anti-kaons are highlighted by red bars. The peaks belong to particle species produced more frequently: pions and protons. The particle identification suffers from a high background, i.e. many pions, protons and accidental matches are often mistaken for kaons, while on the other hand kaons fail to meet the acceptance interval. A careful selection of this interval's size is thus important.

During the rest mass calculation, also a particle's velocity and momentum can be extracted. This is useful, because at higher momenta, the CDC's momentum resolution decreases, since the resulting tracks lie closer together. Moreover, the velocities of different particles with the same momentum become similar at higher momenta. Therefore, a clear separation of positively charged particle species gets difficult, if not impossible. An upper threshold neglecting higher particle momenta can thus increase the overall trigger performance. In addition, an upper threshold for the particle velocity improves the separation between kaons and pions at intermediate momenta. The right panel in fig. 4.12 illustrates these difficulties with a particle histogram. This time, particle counts are indicated by different colors. In the p vs. v plane, different particles are found as curved and elongated clusters. For clarity, negatively charged particles are shown with negative momenta. The windows, in which kaons and anti-kaons are expected, are framed by red lines.

Chapter 5: Trigger Simulation

In addition to the high-level trigger algorithm introduced in chapters 3 and 4, a simulation environment for the entire FOPI trigger process has been created within the scope of this thesis. This environment is meant for the evaluation of the trigger algorithm and the determination of its optimal parameters, along with corresponding efficiency estimations. It also features a built-in event display for algorithm and detector debugging purposes.

5.1 Simulation Framework

In order to evaluate the trigger algorithm and to estimate its performance, the processing modules have been embedded into a framework. This framework sets the parameters for the modules, supplies the algorithm with data, controls the processing chain, collects the results, and calculates a trigger efficiency on their basis. Important topics covered by this section are the source of event data, the operation modes of the framework, and the efficiency calculations.

5.1.1 External Events

It is natural, that one wants to use original data from FOPI events for the evaluation of the kaon trigger algorithm. However, the use of real event data introduces a major disadvantage: the true composition of these events is unknown, since they contain the detector data, but no information about the constituting particles. Hence, an evaluation of the algorithm based on real events is impossible. Yet the environment can still be used as an event viewer (see section 5.2), e.g. for debugging purposes.

Anyway, the data can only be processed off-line, i.e. not during an experiment, because the simulation environment is a software-only implementation that is not hardware-accelerated and thus too slow to process event data in real-time. So the event data has to be supplied in files, one for each event (see appendix A.7.1.1 for details about the file format convention). The files are created elsewhere and can be read by the framework.

To avoid these shortcomings, the data supplied to the simulation framework must be generated from known particles and their tracks. This is usually done by creating events based on the IQMD model (see chapter 1.2.3) and simulating the detector response with the GEANT detector simulation software [GEAN93]. This simulated event data is again made available in files (see appendix A.2.3.1 for the file format conventions), but this time includes the necessary information about the underlying particles and their parameters.

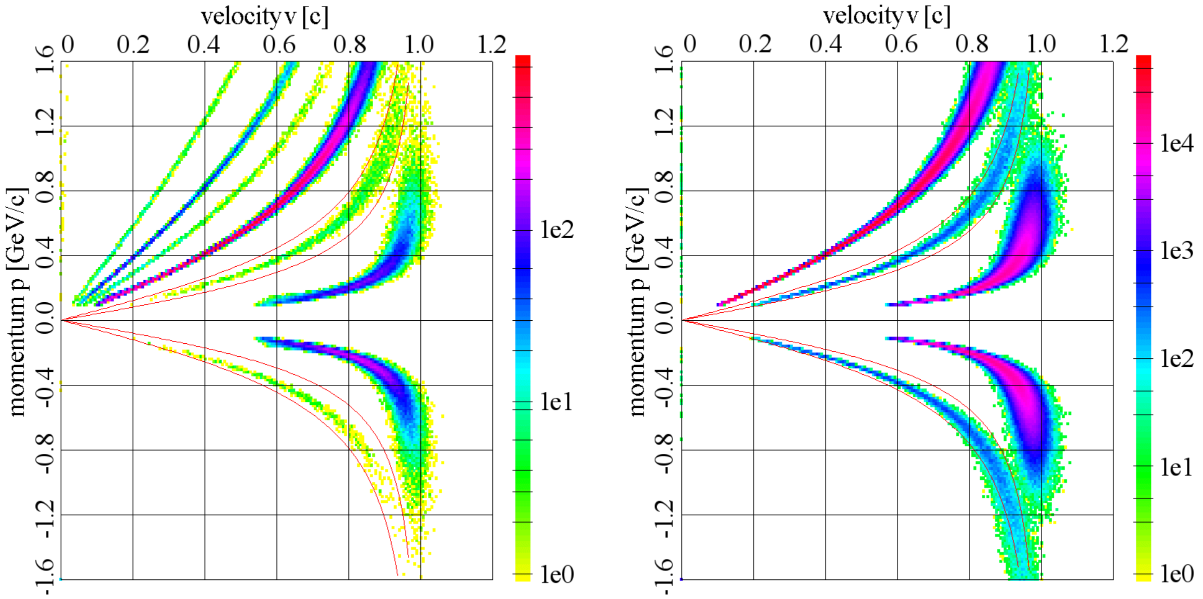


Fig. 5.1: Momentum vs. velocity particle histograms (left panel) for 8,908 GEANT events and (right panel) for 1 M “look-alike” events. While the first includes all observable fragments that are visible as clustered curves (from bottom to top: K^-, π^-, π^+, K^+, p, ^{3}He nuclei, d and t; α are included in the d cluster), the latter incorporates only protons, kaons, and pions. Momenta are calculated assuming $q = \pm e$, thus the results for ^{3}He and α are too low by a factor of 2. Negatively charged particles are reflected towards negative momenta. Kaons are enclosed between the red lines. For both plots, a TOF detector resolution of $\Delta t \leq 80$ ps and $\Delta z \leq 3$ cm has been assumed.

Once in- and output of the trigger algorithm are known, the framework can extract (and save) the following details:

- the numbers of correctly and erroneously detected, and of missed particles,
- the average deviation between extrapolated tracks and assigned TOF hits and
- the kaon status of the event. The kaon status is a number, that indicates the type (positively or negatively charged) of the contained kaon, if any exists, whether or not it has been detected, and if other tracks have been mistaken for a kaon (for further details refer to the explanation of “kaon” in the table in appendix A.1.3.1). This kaon status is also the basis for the efficiency calculation (see chapters 3.2.1 and 5.1.4).

5.1.2 Event Creation

In order to gain reasonable results regarding the trigger efficiency, the algorithm input (i.e. the detector data) must be realistic (e.g. signal quality, phase space distribution of the particle tracks, detector occupancy and hit multiplicity) as must be the expected output (i.e. the different particles with a natural distribution of species, velocity, momentum, and emission angle, confer chapter 5.2.4). However, the underlying physical model, that has been used to generate the data, is negligible for the estimation of the trigger efficiency. Secondly, only a limited number of event data can be made available on a hard disk, and moreover, the parsing of event description files turns out to be considerably time consuming.

The easiest and fastest way to obtain event data that meets the described requirements is to construct them on-the-fly, using only the mentioned assumptions. Events constructed this way look like real events during all processing steps, which can be verified by the event viewer (see section 5.2). So there is no particular reason, why this should not be true as well for the estimated trigger efficiencies. Fig. 5.1 shows a comparison of particle histograms (i.e., properties of particles from many events) from realistic GEANT and from simple look-alike events: the distributions of kaons and the adjacent pions and protons are nearly identical. Other particles, like deuterons (*d*), tritons (*t*) and Helium-3 nuclei (^3He), are missing in the look-alike events. However, their occurrences are rare and their ratios of momenta and velocities do not allow a confusion with kaons, so the simulation results are not changed by this circumstance.

This is why a method for creating such look-alike events has been integrated into the framework. Because of its superior speed and nevertheless comparable quality, it is used for algorithm debugging and the search for the best processing parameters. However, for the calculation of trigger efficiencies, GEANT events are used as far as possible.

5.1.3 Batch Processing

The implemented framework allows the batch-processing of many events, e.g. of all GEANT events contained in the same directory. This is not only necessary for the efficiency calculation (see next section), but also important for the event viewer presented in chapter 5.2, as some of the results can only be extracted from many events:

- the kaon detection rate, i.e. the percentage of correctly detected kaon events,
- the ghost rate, i.e. the percentage of events with erroneously detected kaons, and
- the overall trigger efficiency, which is expressed in terms of the kaon enhancement E.

However, the batch processing has not been enabled for real events, because most of the relevant details can not be calculated on their basis anyway. This is due to the fact, that no information about involved particles is included in their data (confer chapter 5.1.1).

A higher-level batch mode also exists: it includes the batch-processing of either self-created look-alike events or GEANT events, as described above. In addition, it summarizes many batch runs, but can utilize different algorithm parameters for each run. The parameters to be used are supplied in a file (see appendix A.1.3.2 for a detailed description of the format convention). This operation mode is useful for a systematic search for optimal algorithm parameters.

5.1.4 Efficiency Calculation

The estimated trigger efficiency is equal to the kaon enhancement factor introduced in chapter 3.2.1, because this factor expresses exactly the gain in terms of kaons collectible in an experiment, that uses the trigger system, compared to an experiment, that does not.

However, not only the achievement of a high trigger efficiency, but also the knowledge about the variation of kaon detection and ghost rates across different track multiplicities (i.e. the number of particle tracks that fall into the acceptance of the CDC) and kaon momenta, is necessary. In order to draw reliable conclusions about kaon production rates (see chapter 1.3.1), and especially about their momentum distribution and dependence on impact centrality, the effects of the trigger selection must be minimized or at least be known. For example, a high multiplicity leads to an augmented emergence of ghost particles, thus reducing the trigger efficiency. Moreover, due to the vanishing boundaries between protons, kaons and pions in the particle histogram, high momentum kaons are less likely to be detected by the tracking algorithm than others.

Therefore, the framework calculates the efficiencies in more detail: All events are classified with respect to their track multiplicities, and all kaons (only within the CDC and GRPC acceptance) are classified according to their momenta. The ranges of track multiplicities and momenta are then divided into five bins each. Now efficiency results are calculated separately for all events that meet the different bins, and in addition for integrated bins. Integrated bins include the current bin and all bins with lower multiplicities or momenta. By this, 81 different results are obtained, one set for every combination of multiplicity or momentum bins. The combination of the integrations up to the highest values constitutes the overall trigger efficiency.

Ghost rates and the derived trigger efficiencies can only be obtained from the integration over all momentum bins, because no momentum of a real kaon can be assigned to a ghost particle. Therefore, only nine efficiency results remain, but kaon detection rates are provided for all 81 combinations. All results are calculated separately for positively and negatively charged kaons.

5.2 Event Viewer

The second important functionality of the simulation environment is the built-in event viewer. It is very useful for the debugging of the tracking algorithm, the trigger simulation, and also the available event data (and thus the detector itself, although other event viewers exist for that purpose, like the one included in the GEANT package [GEAN93]). In single event operation, the viewer visualizes the event data after each processing step, from the detector data via the track histogram to a display of recognized particles. In batch processing mode, it displays particle histograms and the different efficiency results. The individual items shown in the different displays are described in greater detail in appendix A.8.4.

5.2.1 Detector Data

Detector data is visualized in two different ways, at two different stages in the processing chain, and for two different purposes.

The first detector display shows the digitized and zero-suppressed CDC data versus time for one CDC sector. This is the unmodified data that is contained in GEANT event data files, real event data files, or that has been generated on-the-fly, but this data is not available from the CDC's front-end electronics output. The hit information extracted after the hit recognition (chapter 4.2.1) and filtering (chapter 4.2.2) modules, are also plotted against time. The left panel in fig. 5.2 shows an example. Hence, this display allows their comparison, and the mon-

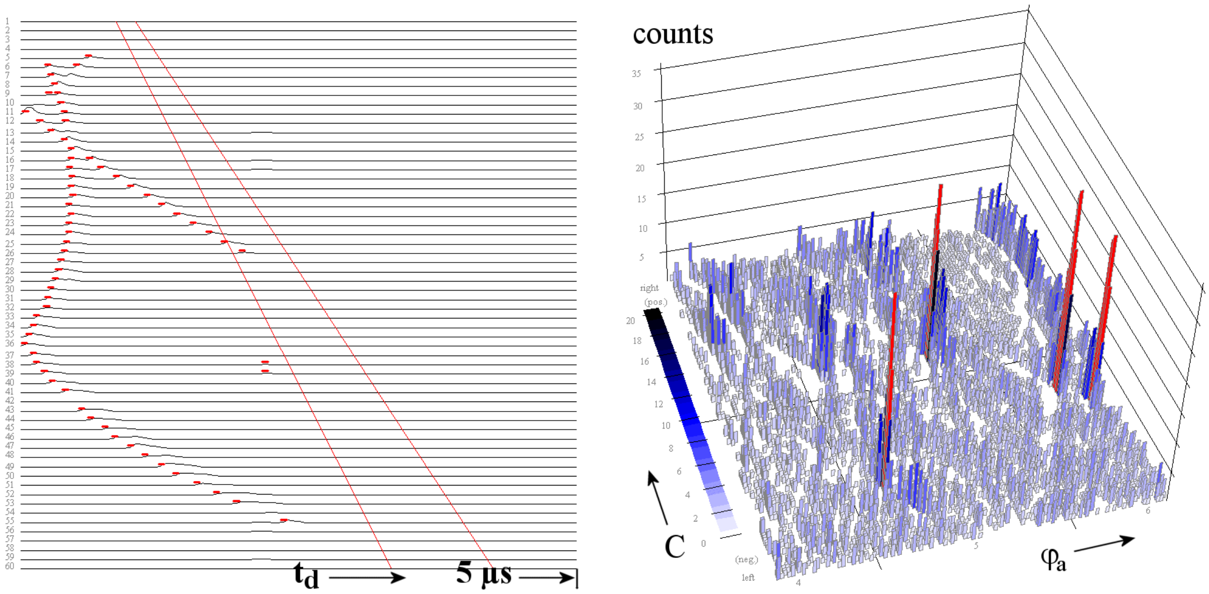


Fig. 5.2: Left panel: Event display of a CDC sector's raw data (drift wire signals, black) and the corresponding FADC discriminator output (red). Lines from top to bottom belong to inner to outer sense wires. The sector boundaries are indicated by red lines. Right panel: Section of a track histogram after filtering. Peak positions result in parameters of recognized particle tracks. Counters with values above the threshold are marked in red.

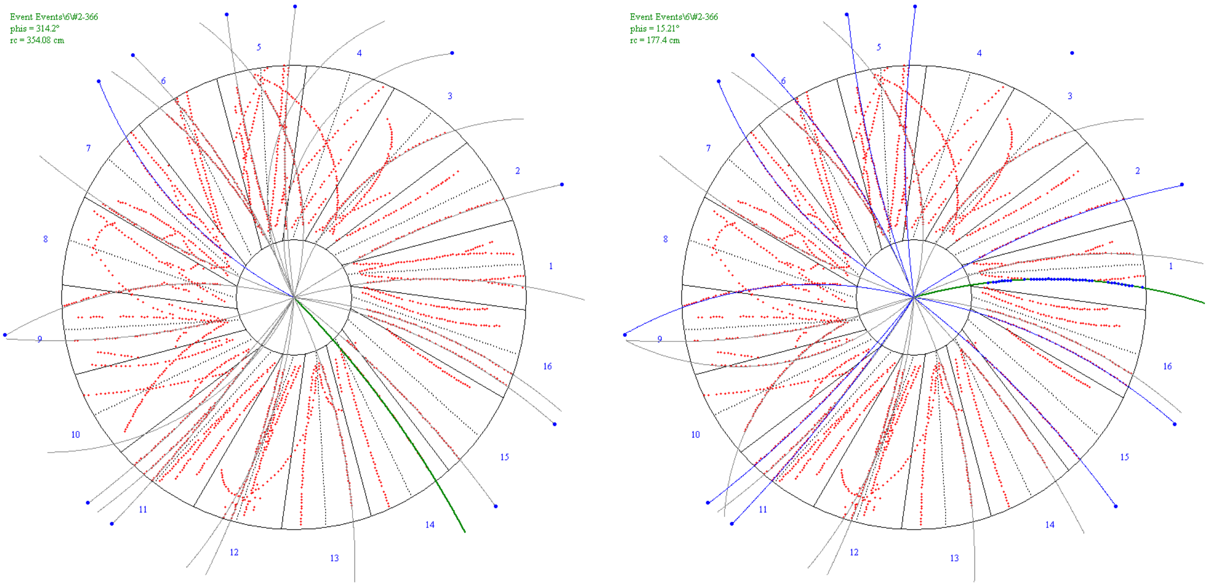


Fig. 5.3: Left panel: Event display showing the CDC with recognized CDC and TOF hits from a Ni + Ni @ 1.93 AGeV GEANT event. Input particle tracks (blue: kaon; green: track with given parameters) are shown for clarity; some of them do not match the CDC hits, see chapter 7.1.1 for details. Right panel: Reconstructed tracks (blue: successfully matched with a TOF hit) for the same event.

itoring and debugging of the involved processing modules. It is essential for the determination of a suitable hit detection threshold in simulations. As soon as real front-end electronics data become available, this display can be used also for their monitoring and thus finding of the correct settings for the discriminator thresholds.

A cross section of the FOPI detector, including the CDC and TOF detector geometries, is shown in the second detector display (see left panel in fig. 5.3). Like the first display, it features all recognized and filtered hits, and in addition all TOF hits. However, they are plotted at their respective positions in the (x, y) plane, which are determined by the detector geometry, the drift time and direction, together with the tracks they are based upon. Kaons are marked in a different color. The main benefit of this event display arises from the comparison between algorithm in- and output (see also chapter 5.2.3). It also enables the valuation of the CDC signal quality, because it puts the detected hits in direct relation to the respective particle tracks.

5.2.2 Track Histogram

The track histogram display reveals the internals of the tracking algorithm, the Hough transform and the subsequent feature extraction. It shows a region of the Hough map before filtering, after background filtering (chapter 4.2.4), and the remains after the application of the maximum filter (chapter 4.2.6) at the same time, by using bars with different heights and color shadings (see right panel in fig. 5.2 for an example). The display makes the effects of the applied filters directly visible, and helps selecting a proper histogram threshold. At least for peaks originating from tracks in less occupied regions of the CDC, it demonstrates the peak shapes that arise from the settings used to calculate the lookup table (chapter 4.1.2).

5.2.3 Track Reconstruction

While the track histogram visualizes the internal event processing, the track reconstruction display shows its result. It is most important for algorithm debugging in single event operation, and shows the output of the tracking (chapter 4.2.5), and in addition, whether the TOF matching (chapter 4.2.9) and the kaon identification (chapter 4.2.10) had positive results. It has the same layout as the second detector data display (see section 5.2.1), thus allowing direct comparisons of tracks and kaons between the algorithm in- and output. An example of such a comparison is shown in fig. 5.3.

If the back-mapping functions, that require the use of the reverse LUT, are enabled, the track display shows also the CDC hits that have been associated with a selected particle track.

5.2.4 Particle Histograms

In multi-event operation, particle histograms provide the most powerful visualization of algorithm data. For both in- and output, two different versions have been implemented: a mass and a momentum vs. velocity histogram.

In the mass histograms, different particle species are visible as individual peaks (see left panel of fig. 5.4). Their heights are a measure for a particle species' frequency of occurrence, and their widths play an important role in particle recognition, because they determine, whether different particle species can well be separated. The mass histogram thus helps finding the ap-

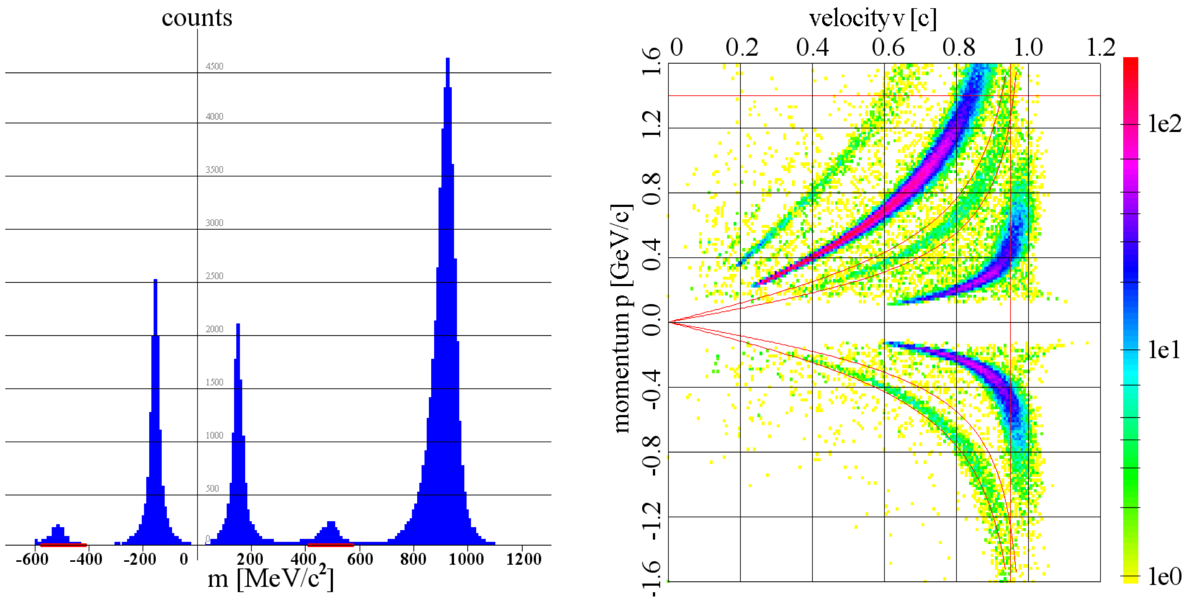


Fig. 5.4: Left panel: Mass particle histogram for the simulation output of 9,848 Ni + Ni @ 1.93 AGeV GEANT events, that have been enriched with kaons. Right panel: Momentum vs. velocity particle histogram for the same events. Negatively charged particles are shown with negative masses and momenta. The peaks and clusters belong to (from left to right and bottom to top, respectively) K^-, π^-, π^+, K^+ and p. The kaon recognition windows are marked in red. Background entries are caused by false TOF matches and lead to the emergence of ghost particles.

propriate acceptance window for the kaon recognition. The comparison between the in- and output mass histograms delivers an excellent demonstration of the algorithm's particle reconstruction capabilities.

The second type of particle histogram contains an additional dimension of information. Particle momenta, velocities, and the number of particles found with the respective parameters are displayed. Particle species are now clearly visible curved clusters in the (p, v) plane. A particle spectrum of this type is shown in the right panel of fig. 5.4. These displays give a more detailed insight into the particle reconstruction capabilities, because now also the dependence of the separability upon momentum and velocity can be assessed. Hence, they allow a proper setting of the momentum and velocity thresholds discussed in chapter 4.2.10.

5.2.5 Efficiency Diagrams

For the trigger simulation with many events, all efficiency results that were presented in chapter 5.1.4 are displayed.

Three items are shown for different track multiplicity bins and combinations using bar graphs: the measured kaon reconstruction efficiency, the ghost kaon event rate, and the resulting trigger efficiency, which is represented by the kaon enhancement factor introduced in chapter 3.2.1. The multiplicity bins are equal in size and range from zero to the largest number that

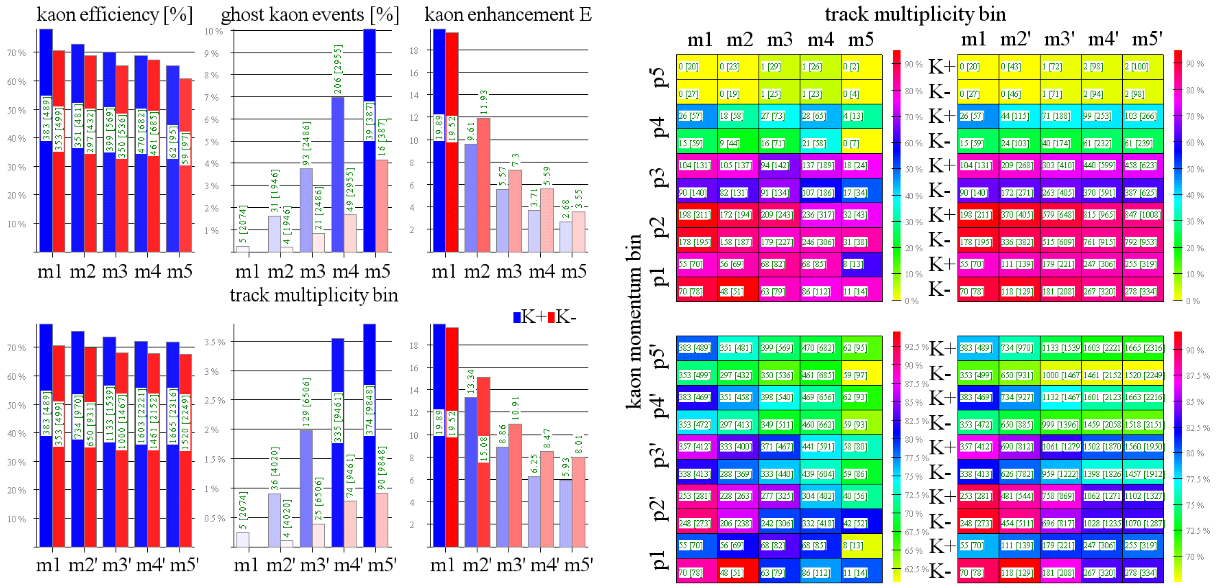


Fig. 5.5: Left panel: Kaon efficiency, ghost kaon event rate, and kaon enhancement for different track multiplicity bins (m1 to m5). Red bars visualize the results for K⁺, blue bars for K⁻. The green numbers are absolute values. In the lower row multiplicity bins m2' to m5' stand for combinations of m1 through m2 to m1 through m5 from the upper row. Right panel: Kaon efficiencies for different track multiplicity (m1 to m5) and kaon momentum bins (p1 to p5). Primed labels again indicate combinations of bins. Subsequent rows represent values for K⁺ and K⁻ alternately. Reconstruction efficiencies are shown as different colors, absolute values are again given by green numbers. The lowest efficiencies appear as yellow, while the highest are red. A momentum cut of p < 1.4 GeV/c has been applied.

occurred. The results are shown without a distinction between different kaon momenta, but separately for K^+ and K^- and for different track multiplicities within the CDC acceptance. The left panel of fig. 5.5 gives an example for this display.

In addition, the kaon reconstruction efficiency is shown in greater detail. Results for all combinations of track multiplicity and kaon momentum bins and also combinations of the integrations of these bins are given, again separately for K^+ and K^-. This time, different colors are used for the visualization of the measured efficiencies (see right panel in fig. 5.5). The momentum bins have an upper boundary of $2.2\,GeV/c$.

Chapter 6: Hardware Implementation

The full structure of the trigger algorithm, including the necessary exploitation of all parallelization benefits, can not be imitated in the software simulation within reasonable efforts. However, the porting of the trigger algorithm to the chosen hardware platform MPRACE (see chapter 3.4.3) is straightforward, because the algorithm has been designed according to its special characteristics: the algorithm collects CDC hits from many input channels, transforms them into a track histogram and extracts positions of local maxima (see chapter 4.2). The hit coordinates are used to address a lookup table; values read from the lookup table serve as addresses for the incrementation of counters in the track histogram.

MPRACE enables the parallel execution of this algorithm (cf. chapter 3.4.4): a single board offers up to 144 inputs to receive the CDC data. The lookup table can be stored in the fast onboard SRAM, that allows very large data words (up to 288 *bit* per coprocessor). The histogram is kept inside the FPGA, which provides up to 96 independent memory blocks for that purpose. The FPGA also supplies configurable logic resources, that implement the necessary address logic and data paths. All functional units of MPRACE can work in parallel. If a sufficient number of MPRACE accelerator boards is employed, one CDC hit can be processed per clock cycle. How this can be achieved, is discussed in the following.

6.1 Parallelization Issues

In order to make the FOPI trigger algorithm meet the timing requirements (chapter 3.2.1), a high degree of parallelism is necessary (see chapter 3.2.3). It can be introduced by two different techniques: while the first implements identical processing modules many times in order to compute independent datasets in parallel data paths, the second, pipelining, sends the individual datasets in succession through a line of different processing modules that work in parallel. While it is obvious, that some of the trigger's processing modules can be pipelined (e.g. the particle identification can start as soon as the TOF matching releases its first data, and the hit filter does not need to wait until the hit recognition is completely finished) or executed in parallel (e.g. the hit recognition is independent for different wires, as is the particle identification for different kaon candidates), two additional features promise a great enhancement for the level of parallelism: analogies in the designs of lookup table and histogram, and the independence of the different filters applied to the histogram contents.

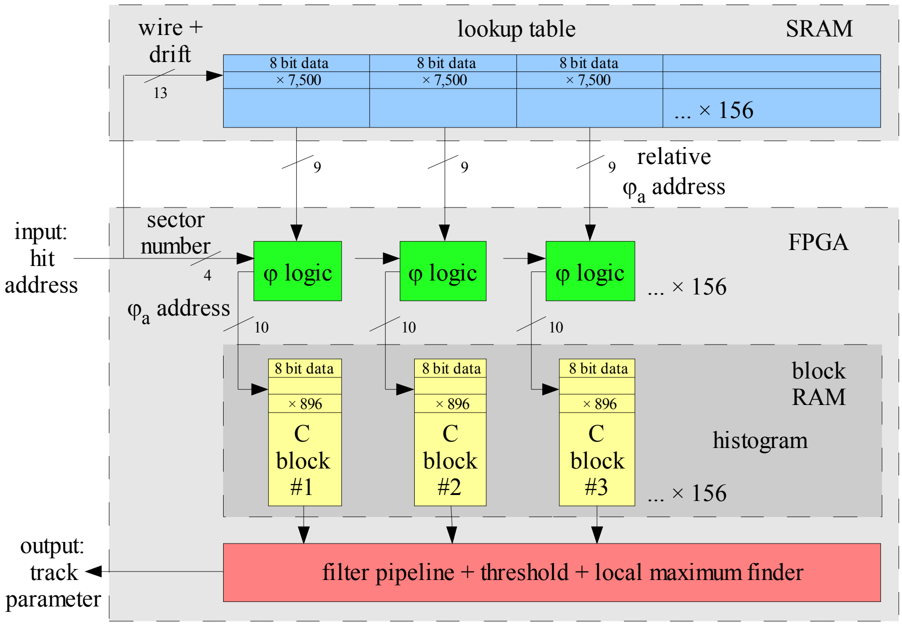


Fig. 6.1: Sketch of the tracking implementation after [Sess00]. Hit coordinates with removed sector numbers are used to address the lookup table. The retrieved data word contains relative φ_a values for all possible values of the momentum parameter C. Together with the original sector number they represent the addresses of the respective histogram counters within the corresponding C blocks to be incremented. After the histogramming, the filtering, the setting of a threshold and the extraction of local maxima is done by feeding the histogram contents into a large pipeline. The given numbers are based on the following dimensions: 896 φ_a values, 156 C values, 30 wires used per CDC sector and 25 MHz sampling frequency of the FADC discriminator signals. The lookup table in this example has a size of 1.12 MBytes and a width of 1,248 bits, the histogram measures 136.5 kBytes.

6.1.1 LUT Access

The lookup table is organized in Hough space patterns: if a hit coordinate is supplied, a complete pattern consisting of (C, φ_a) tuples can be retrieved. This is usually done by reading the corresponding φ_a (and possibly $\Delta\varphi_a$) entries for single $(wire, drift\ time, C, drift\ direction)$ combinations.

However, while nothing else is said about the φ_a at this point, exactly one of these values exists for every possible *C*. So one can as well read the complete pattern, with the entries for all *C* together, by supplying the hit coordinates alone. This must then be done only once for every hit. The bit width of the values retrieved from the lookup table then multiplies by the number of possible *C*, and consequently the time needed for the entire Hough transform reduces by the same factor, provided that the histogram access (see next section) is fast enough as well. To put it in other words: the access to the lookup table is parallelized and thus accelerated by increasing the width of the retrieved data words.

6.1.2 Histogram Access

Access to the histogram data can be parallelized at both sides of the algorithm, i.e. at the compiling of the Hough map as well as at the analysis of its final contents.

- As seen in the previous section, the output of the lookup table is a large data word, that replaces single φ_a values for each possible C. They can be recovered by simply splitting the data word. On the other hand, the histogram is also organized in C blocks, because all counters with the same C coordinate represent a single column in the Hough map. If now an individual counter incrementation logic is implemented for every single column, all columns can be processed in parallel, thus handling the complete data word in a single step. Fig. 6.1 gives an overview of the implementation with parallel C blocks: between the addressing of the lookup table and the application of the histogram background filter, all C blocks are completely independent and can thus be processed in parallel.
- The post-processing of the histogram data is more complicated, because in the different filter implementations, the individual counter values are no longer independent. However, if an own post-processing pipeline of sufficient length is implemented for every C block, data from the neighboring (as suggested by the filter layouts) pipelines may be taken into account. Hence again, all C blocks can be treated in parallel. Moreover, the histogram filter, the maximum finder, and the maximum filter can be pipelined one after another, because values depend only on their neighbors, and their order is preserved during the complete processing. Hence, a new result is available at the end of the pipeline for every C block with every clock cycle. Thus, the execution of the FOPI trigger algorithm is further accelerated.

6.2 Implementation Layout

This chapter presents a sketch of the logic modules and the data flow through them, as needed for the hardware part of the trigger implementation. It is mainly based on preparatory work for the ATLAS TRT algorithm suggested by M. Sessler [Sess00]. It utilizes the considerations on parallelism mentioned above, and the detector symmetries (chapter 4.1.1) as far as possible.

6.2.1 LUT Addressing

Due to the symmetries of the CDC wiring and the search patterns, the lookup table contains only transformation rules for CDC hits from a single sector. The addressing of the LUT and the track histogram can thus be done as follows (cf. fig. 6.2): the Hough map is thought as divided into 16 sectors, in analogy to the 16 CDC sectors, and the φ_a values read from the lookup table are interpreted as relative to the current histogram sector, that is identical to the sector of the hit in the CDC. So results for the other sectors are easily obtainable by adding multiples of the number of φ_a values per sector to the data retrieved from the LUT.

Since wire and drift time numbers are the coordinates in the lookup table, the LUT addresses can easily be generated from the hit coordinates. The sector number must simply be stripped from the complete address for LUT access. The φ_a output of the LUT access is modified accordingly. The relative sector offset is incremented by the hit's sector number, that has previ-

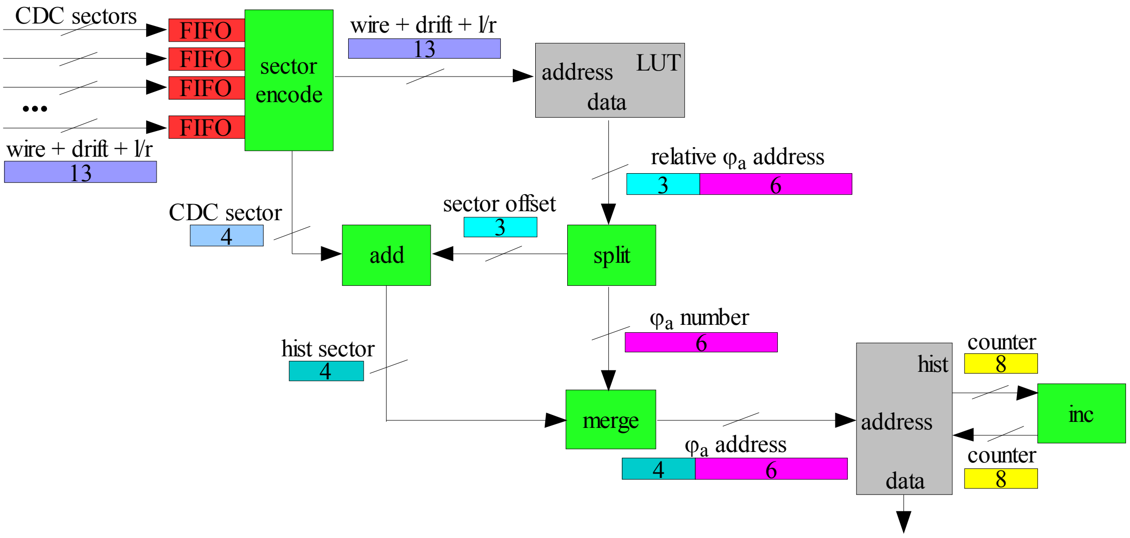


Fig. 6.2: Sketch of the addressing of the lookup table and the track histogram for a single C block. Following the CDC symmetry, sector numbers are treated separately. Wire number, drift time index and drift direction of identified CDC hits are collected in FIFOs and interpreted as LUT addresses, and the sector number is generated. The φ$_a$ address read from the LUT for a C block is divided into a sector offset and a pure φ$_a$ number. Sector number plus offset point to the virtual histogram sector, within which the φ$_a$ number is the address. The corresponding counter is then incremented.

ously been separated from the hit coordinates. The total sector plus the remaining relative φ_a value then gives the address of the memory cell in the Hough map to be incremented.

6.2.2 Histogram Performance

The LUT access delivers for every histogram *C* block one φ_a address per cycle. Thus, subsequent φ_a values are available at a rate of up to 133 *MHz*, with a latency of only two cycles after the addressing (SRAM). In the FPGA, the histogram counter values stored at the respective addresses must be read, incremented, and written back within the same cycle. In principle, there are two possibilities to nevertheless maintain the data rate provided by the address pipeline:

- The histogram is stored in the internal block RAM of the FPGA, access to which can be done considerably fast as well. If the histogram incrementation module can operate at a higher (e.g. two- or four-fold) clock frequency than the LUT access and the sector logic, it is ready to receive the next address at the next cycle of this external pipeline clock. While the external ZBT-SRAM accepts up to 133 *MHz*, the logic inside modern Xilinx® Virtex™ II FPGAs, including the RAM blocks, can be operated at more than 200 *MHz*, depending on the configuration.
- The *C* blocks of the Hough map, that contain the counters for the different φ_a addresses, can further be divided into real sectors (similar to the partition of the CDC into sectors, but not to be confused with the virtual histogram sectors introduced earlier). If they are stored

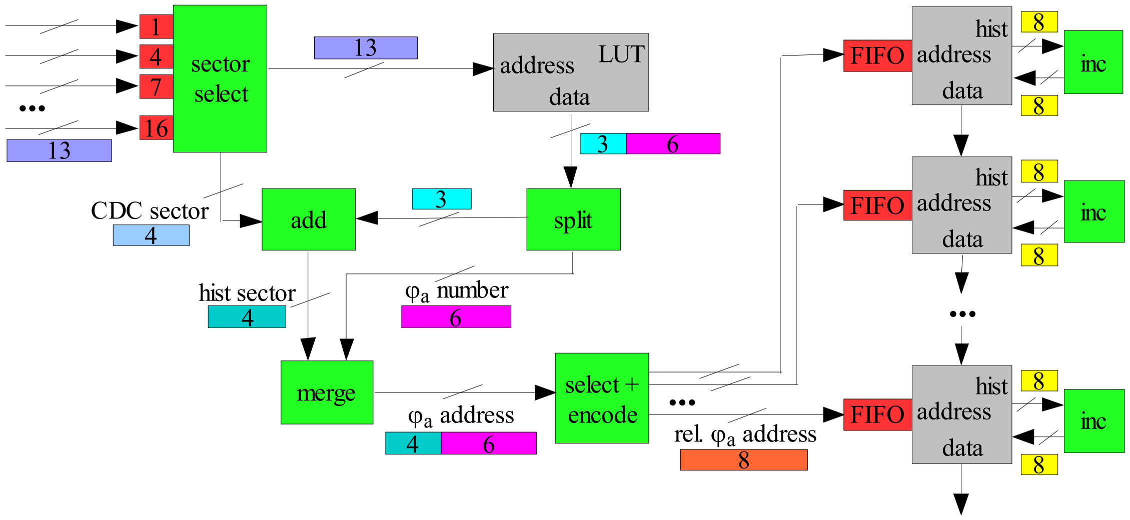


Fig. 6.3: Modified layout of the lookup table and histogram addressing using divided histogram C blocks. Hit data is read from alternating CDC sectors in such a fashion, that the same and neighboring sectors are not requested in successive cycles. The histogram C blocks are split into sectors, which will then not be addressed by successive hits. Thus, the incrementation logic for this C block sector has enough time before the next counter address arrives. Additional FIFOs ensure an equal distribution of counter addresses over time, if hits in the CDC are concentrated within a few sectors.

in separate RAM blocks, they can be addressed alternately to leave each of them enough clock cycles to complete the incrementation operation, while simultaneously maintaining the full speed of one hit per cycle for the total histogram (see also fig. 6.3 for an illustration). This is possible, because a hit, that belongs to a certain CDC sector, can only belong to tracks, that have their φ_a angle in the same or one of the neighboring sectors. Therefore, only the corresponding histogram sectors can accordingly be affected by the incrementation. In order to serve the sectors of a *C* block alternately, the hits from different CDC sectors must be fed into the histogramming pipeline successively. The histogram address logic must thus be extended in order to select the correct *C* block sector and supply it with the right φ_a number.

6.2.3 Filter Pipeline

The post-processing of the histogram data can be implemented in one parallel pipeline for each *C* block, and comprises the background filter, histogram thresholding, and the maximum finder with its filter (see chapters 4.2.4 through 4.2.6).

Counter values from each histogram block are fed into the corresponding pipeline one after another, a new value with each clock cycle. Depending on the applied filter mask and logic for both of the two filters, histogram values are propagated through the pipelines and compared or added during the following cycles. The result is subtracted from (in case of the background filter) or compared to (for the maximum filter) the original counter value. Fig. 6.4 shows a

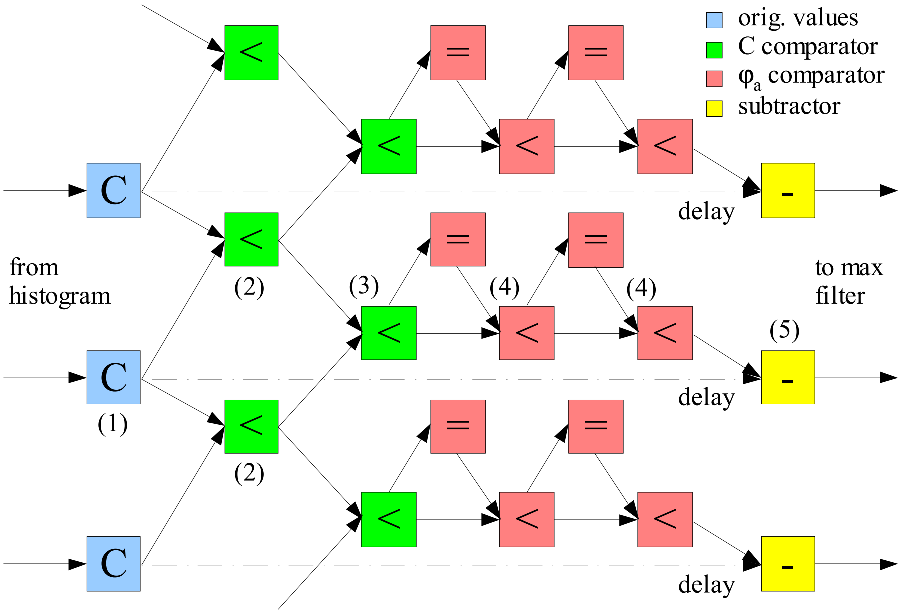


Fig. 6.4: Part of an implementation of the "3×3 minimum" histogram filter. Each C block has its own pipeline. A counter value (1) is first compared to the values of the neighboring pipelines (2), the smallest value survives (3). Afterwards, succeeding results are propagated (4) and compared to the first value. The final result (i.e., the minimum value in this case) is subtracted from the original counter value (5).

sketch of the implementation of this pipeline for the "3×3 minimum" histogram filter (see appendix A.3.2.2 for its layout). The size of the respective filter mask determines the number of incorporated neighboring pipelines and the lengths of the corresponding pipeline segments. At the end of the pipelines, only a stream of binary values remains, that indicate, whether each of the corresponding histogram counters has been identified as a valid maximum.

The histogram coordinates of the extracted maxima are then passed to the last processing stages, the matching with the TOF data and the particle identification. They can more conveniently be done on a CPU, due to the involved floating point operations, which are extraordinary resource demanding on FPGAs [Kube98]. The trigger decision can be generated, as soon as the particle identification has finished, or earlier, if a particle has been identified within the mass window for the acceptance of kaons.

6.2.4 Execution Speed

From the discussed layout of the algorithm implementation and the possibilities of the MPRACE hardware, the time needed for the execution of central parts of the algorithm (i.e., histogramming through maximum extraction) can now be estimated. The histogramming part consumes twice as many clock cycles, as hits (derived from the CDC track multiplicity M_T) are detected on those CDC's sense wires (N_W), that are used as input for the trigger. The factor 2 is owed to the mirror hits, that must be histogrammed as well. The additional latency of the SRAM and the incrementation logic of few clock cycles can be neglected. A contribution

CDC wires per sector N_W	avg. track multiplicity M_T	histogram φ_a values	design frequency $f\ [MHz]$	execution time $t_{TRK}\ [\mu s]$
60 (all)	60 $(Au+Au)$	800	133	60
			66	120
		400	133	57
			66	114
	25 $(Ni+Ni)$	800	133	29
			66	57
		400	133	26
			66	51
20 $(1/3)$	60 $(Au+Au)$	800	133	24
			66	48
		400	133	21
			66	42
	25 $(Ni+Ni)$	800	133	14
			66	27
		400	133	11
			66	21

Table 6.1: Execution times for the histogramming and the maximum extraction on the MPRACE coprocessor. The number of used CDC wires per sector is approximately equal to the number of hits per track. SRAM and pipeline latencies are neglected. See also eq. (6.1).

of the hit recognition is also very small, because it is pipelined before the histogrammer, so that the latter can start as soon as the first hit coordinates are available. The time needed for filtering, thresholding and maximum extraction is determined by the length of this pipeline (which is again neglected as a first approximation) and the number of values to be propagated through it. This number is identical to one of the dimensions of the histogram, the number of different starting angles N_{φ_a}. If f is the FPGA's design frequency that is also used to access the lookup table, the execution time t_{TRK} for the trigger modules responsible for the tracking becomes

$$t_{TRK} = \frac{1}{f}\left(2 M_T N_W + N_{\varphi_a}\right). \tag{6.1}$$

The results for different configurations of the CDC readout and the track histogram, using different types of events and FPGA clock settings, are summarized in table 6.1: most of the values are far below $t = 100\,\mu s$, which was the first assumption for the total trigger processing time t (including particle identification) and thus the basis for an estimate of the kaon enhancement E achievable with the trigger system of above 5 to 8 (chapter 3.2.1).

6.3 Resource Consumption

The amount of resource consumption of the trigger algorithm in the FPGA device depends on the granularity of the Hough map and the CDC data. This is an overview of the relevant parameters and their effects on the required logic and block RAM resources. Examples for an MPRACE configuration are given.

6.3.1 Parameter Dependence

Several of the parameters that can be adjusted in the trigger algorithm have their influence on different hardware resources.

- The search patterns' φ_a resolution determines one dimension of the track histogram. Speaking of hardware, this affects the necessary size of the RAM blocks, in which the C columns of the histogram (see chapter 6.1.2) are stored. It can also influence their number, if the required number of RAM cells exceeds the maximum block size available in the used FPGA devices. Also the bit width needed to address them is concerned. Since these addresses are stored in the lookup table, the length of a data word to be retrieved from it also changes, and thus changes the required bit width of the access to the external lookup table memory.
- The C resolution of the predefined tracks is responsible for the number of required RAM blocks, and thus likewise for the width of lookup table data words and the necessary width of the data access to the external memory.
- The sampling frequency for the CDC data, which is directly connected to the number of different possible hits per sense wire, accounts for the number of lookup table entries, and thus the size of the LUT and also the bit width of the lookup table addressing.
- Finally, the fraction of used CDC sense wires is likewise responsible for the size of the lookup table and its address bit width. Moreover, it influences the fraction of the events' hit multiplicity to be processed, so the computing time for the histogrammer (chapter 4.2.3) and the required resources for the hit detection (chapter 4.2.2) change as well.

6.3.2 Example Configurations

As mentioned in chapter 3.4.3, the MPRACE FPGA coprocessor can be equipped with additional memory on a mezzanine board. Hence, the bit width for the access to the lookup table can be increased, allowing more C columns of the histogram to be addressed. Table 6.2 provides an overview of the resources available on a single MPRACE board for different configurations.

The required lookup table bits per C block result from the necessary bits for coding a single sector (1/16) of the φ_a values plus 3 extra bits for a sector offset, because the lookup table contains only the transformation for one sector of the CDC. In addition, due to the tracks' cur-

mezzanine equipment	resulting LUT RAM width	histogram φ_a values	addressable C blocks	max. sectors per C block	remaining interconnect
–	$144\,bit$	$257-512$	20	4	$144\,bit$
		$513-1024$	18	5	
		$1025-2048$	16	6	
1 connector ($36\,bit$)	$180\,bit$	$257-512$	25	3	$108\,bit$
		$513-1024$	22	4	
		$1025-2048$	20	4	
2 connectors ($72\,bit$)	$216\,bit$	$257-512$	30	3	$72\,bit$
		$513-1024$	27	3	
		$1025-2048$	24	4	
3 connectors ($108\,bit$)	$252\,bit$	$257-512$	36	2	$36\,bit$
		$513-1024$	31	3	
		$1025-2048$	28	3	
4 connectors ($144\,bit$)	$288\,bit$	$257-512$	41	2	–
		$513-1024$	36	2	
		$1025-2048$	32	3	

Table 6.2: Example configurations for the MPRACE coprocessor board. The available bit width of the SRAM memory can be increased with mezzanine boards, leaving different numbers of connectors for I/O tasks. The number of addressable C blocks results from the bits needed to encode the φ_a addresses. This in turn sets a limit to the number of RAM blocks, that can be used to divide the C blocks into subsectors.

vatures, Hough patterns from up to approximately half of the histogram can refer to the hits in that sector.

Additional interconnect between MPRACE boards can be implemented via the dedicated high-speed board-to-board connectors. These connections are well suited to distribute the CDC hit information among different histogrammer boards, or to collect track candidates for the matching and particle recognition procedures.

Depending on the mezzanine configuration, up to 144 CDC sense wires can be read out and preprocessed on a single MPRACE board. Alternatively, up to 41 C blocks can be addressed on each board by the lookup table. Hence, the available logic and routing resources decide, whether specialized boards in individual configurations must be used for data input and processing, or if both tasks can be executed on the same boards. This question is addressed in greater detail in chapter 7.3.2.

Chapter 7: Performance Results

Simulations of the kaon trigger system with realistic GEANT events as well as with events generated on-the-fly (see chapter 5.1.2) have been done in order to estimate its performance. The results cover four different aspects:

- estimation of the maximum possible performance,
- determination of suitable algorithm parameters,
- performance stability with respect to resource optimization,
- performance variation with event properties.

A detailed knowledge of the performance, the corresponding conditions and constituting parameters is crucial for the final system design. Moreover, it is substantial for the planning of experiments to be run with the trigger system enabled, in order to be able to draw valid conclusions from the measured data.

7.1 Simulation Conditions

The simulation of the trigger system requires a pool of realistic event data as well as a profound estimate of all involved quantities. These include the general conditions derived in chapter 3.2 and the implementation strategy outlined in chapter 6, but also physical quantities like kaon spectra and production rates, or achievable clock frequencies and data bandwidths in connection with the target hardware.

7.1.1 Employed Event Data

The event datasets used for the performance measurements have been prepared by Yvonne Leifels according to the description in chapter 5.1.1 [Leif02]. They represent simulated experiments of two different systems, $Ni+Ni@1.93\,AGeV$ and $Au+Au@1.5\,AGeV$. Table 7.1 and fig. 7.1 show some further details about these event data, like kaon contents or multiplicity distributions for tracks (track information is available in the PID description file, see appendix A.2.3.1) and complete particles (for which both track and TOF information are available).

Decays (see chapter 1.1.1) are not accounted for in the simulated TOF data; this means, that a TOF record is included for every particle that has been emitted into the direction and within the acceptance of the GRPCs. The description file also contains the track parameters for those particles, even if due to an early decay no track can actually be found in the CDC data. This

experiment	$Au+Au@1.5\,AGeV$	$Ni+Ni@1.93\,AGeV$
projectile / target	gold / gold	nickel / nickel
impact energy	$1.5\,GeV/u$	$1.93\,GeV/u$
number of events	8,908	11,825
number of K^+	4,047(1,402)	2,166(694)
number of K^-	724(178)	6,779(2,386)

Table 7.1: Some important properties of the GEANT event data that were used in the simulations. Numbers in brackets denote particles, whose reconstruction is not possible, because either the time-of-flight or the track information is missing, due to the geometrical acceptance of the respective detector.

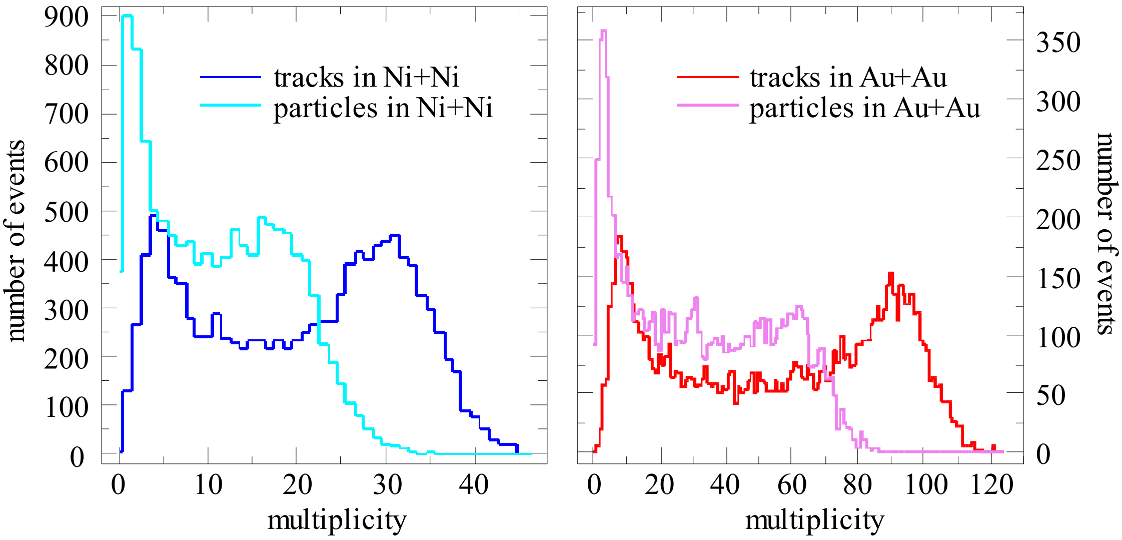


Fig. 7.1: Multiplicity distribution of particle tracks and complete particles (that include the TOF information) in Ni + Ni (left panel) and Au + Au (right panel) GEANT events. The data is enhanced with central events; the resulting average number of tracks per event is about 22 for Ni + Ni and 59 for Au + Au.

circumstance is expected to reduce the measured tracking efficiencies. An example has already been shown in fig. 5.3, where all tracks of an event could apparently be reconstructed. However, the tracking efficiency only results to 88 %, and in addition, three ghost tracks have to be counted.

Moreover, the events have been strongly enriched with kaons (almost one per event) to allow for a better statistics within fewer events. According to the geometrical acceptances of the CDC and the GRPCs, particle data may be incomplete, if the information of either detector is missing. So events with only partial kaon data can nevertheless be consulted for the investigation of the ghost kaon suppression. This is possible and fortunately does not affect the simulations, because viewing the event data alone, there is no difference between an event with incomplete kaon data and an event without a kaon; also in the off-line analysis, kaons cannot be identified with either the track or the TOF information missing (particle identification using instead the energy loss dE/dx along the path of flight leads to significantly less accurate results, rendering them virtually useless above kaon momenta of about $p=0.5\,GeV/c$).

The TOF data contained in the event files is exact in time as well as in space. Thus, prior to input of these data into the particle recognition algorithm, they must be blurred with statistical errors according to the achievable detector resolutions.

Events generated on-the-fly have been used only for a faster identification of the best algorithm parameters, but not for measurements of the actual performance.

7.1.2 Standard Parameters

The performance measurements are based on a standard set of parameters. Some, like the histogram threshold and the different filter masks, are pure algorithm options and can in principle be set arbitrarily. The results of the efficiency simulations have been used to find appropriate combinations for the optimization of the kaon enhancement E in eq. (3.5). A detailed discussion follows later in this chapter, so only fig. 7.6 and table 7.2 are referenced for a summary at this point. However, two other groups of parameters require an investigation as well, because they are given by the experiment (physics, detector and DAQ), and by the selected hardware platform for the trigger.

The occurrence of kaon signatures in the event data is determined by the geometric acceptances of the CDC and the GRPC barrel, and by the system under investigation, which can be characterized in terms of the size of the reaction partners, the incident beam energy, and the distribution of impact centralities. For the available event data, $Ni+Ni@1.93\,AGeV$ and $Au+Au@1.5\,AGeV$, the following rates can be assumed as realistic for a real experiment: $N_{K^+}^{Ni+Ni}=100$, $N_{K^-}^{Ni+Ni}=5{,}000$, $N_{K^+}^{Au+Au}=20$, and $N_{K^-}^{Au+Au}=1{,}400$ [Herr04]. In the case of the $Au+Au$ data, more pessimistic values of 16 and 800, respectively, have been used for the calculations instead. These values are still within a reasonable uncertainty in these experiments; moreover, for the examples given in chapter 3.2.1, the calculated kaon enhancements change only from 2,98 to 2,90 for K^+ (assuming $b=20\%$) and from 5,45 to 5,43 for K^-, so the introduced error is negligible.

The DAQ time for an event depends on the reaction partners and the geometric acceptances as well, but in this case expressed in terms of the hit multiplicity in the CDC. The time the SSMs need to collect the data from the FADC modules is fixed to $T_{SSM}=0.8\,ms$. For the data compression that is executed in the SAM3 units, the very optimistic values of $T_{SAM}^{Au+Au}=1\,ms$ and $T_{SAM}^{Ni+Ni}=0.4\,ms$ were assumed. Moreover, the total DAQ latency, which currently is at least the sum of both times (cf. chapter 2.3.2), was estimated to be as low as the largest of the latencies of the involved DAQ submodules, thus resulting in $T^{Au+Au}=1\,ms$ and $T^{Ni+Ni}=0.8\,ms$.

Finally, the design frequency of the FPGA implementation was assumed to be $f=67\,MHz$. This is safely below the possible clock settings of at least around $f_{FPGA}=125\,MHz$ and $f_{SRAM}=133\,MHz$, but already includes the assumption, that the incrementation of a histogram counter can be done within a single of these cycles (cf. chapter 6.2.2). The total number of clock cycles needed for histogramming, filtering and maximum finding is then given by the number of involved CDC hits, which depends on the hit multiplicity and the fraction of readout sense wires, plus the number of φ_a values in the histogram (cf. chapter 6.2.4). In addition, other parts of the processing (TOF matching, particle identification) were estimated to take

$5\,\mu s$ in total. Since this is the only assumption, which is optimistic in favor of a large kaon enhancement, the simulation results obtained according to eq. (3.5) will still be rather pessimistic.

7.2 Trigger Efficiency Results

As explained in chapter 3.2.1, the kaon enhancement factor E in eq. (3.5) is a good measure for the performance of the trigger system. Yet this number is not the only result of interest, because the trigger algorithm may produce results of different quality for different types of events. Hence, the quantities that characterize aspects of its performance and their connection to the events' properties are presented in larger detail.

7.2.1 Tracking Efficiency

The core of the trigger algorithm, the track reconstruction, is the basis for all other computations. Hence, also its performance is of general interest. Results for three aspects of the tracking efficiency can be obtained: the percentage of correctly reconstructed tracks, the number of produced ghost tracks, and the accuracy of the reconstructed tracks. While the accuracy is directly connected to the granularity of the Hough map, the other results can be influenced by tuning the parameters of the processing chain. Tracking accuracy in this context is defined as the deviation of a reconstructed from the original track at the location of the TOF detector.

The first two plots in fig. 7.2 show the tracking efficiencies and ghost rates for $Ni+Ni$ and $Au+Au$ events as a function of the histogram threshold. Both naturally drop with rising thresholds: in $Ni+Ni$ ($Au+Au$) events, the reconstruction efficiency starts at $60\,\%$ (above $60\,\%$) for small threshold values, and drops via $50\,\%$ ($40\,\%$) for intermediate to $15\,\%$ ($7\,\%$) for high thresholds. The number of produced ghost tracks per event changes accordingly from 70 (115) via 5 (8) to slightly above 1. However, while the tracking efficiencies decrease only slowly from lower to medium thresholds, the ghost rates are already reduced by more than an order of magnitude. This is a first hint, that an effective trigger algorithm will have to avoid the use of the lowest thresholds.

The efficiencies for both types of events differ visibly especially at higher thresholds. A possible interpretation is, that the background filter reduces peak heights more efficiently (instead of only separating them from surrounding background), if the peaks lose sharpness due to the higher and thus noisier background in events that contain more particle tracks. The absolute values of the tracking efficiency are further reduced by the fact, that particles with an early decay also count as missing tracks: since the difference is not visible in the event description, it cannot be revealed by the performance measurement either.

On the other hand, the ghost rate is higher by almost a factor of 2 in $Au+Au$ events. This behavior can be explained by the higher occupancy in the Hough map, that makes accidentally compiled peaks reach beyond the threshold by far more easily. This is also reflected by the situation in the CDC: a high hit occupancy increases the probability for false combinations of track segments (compare fig. 4.11).

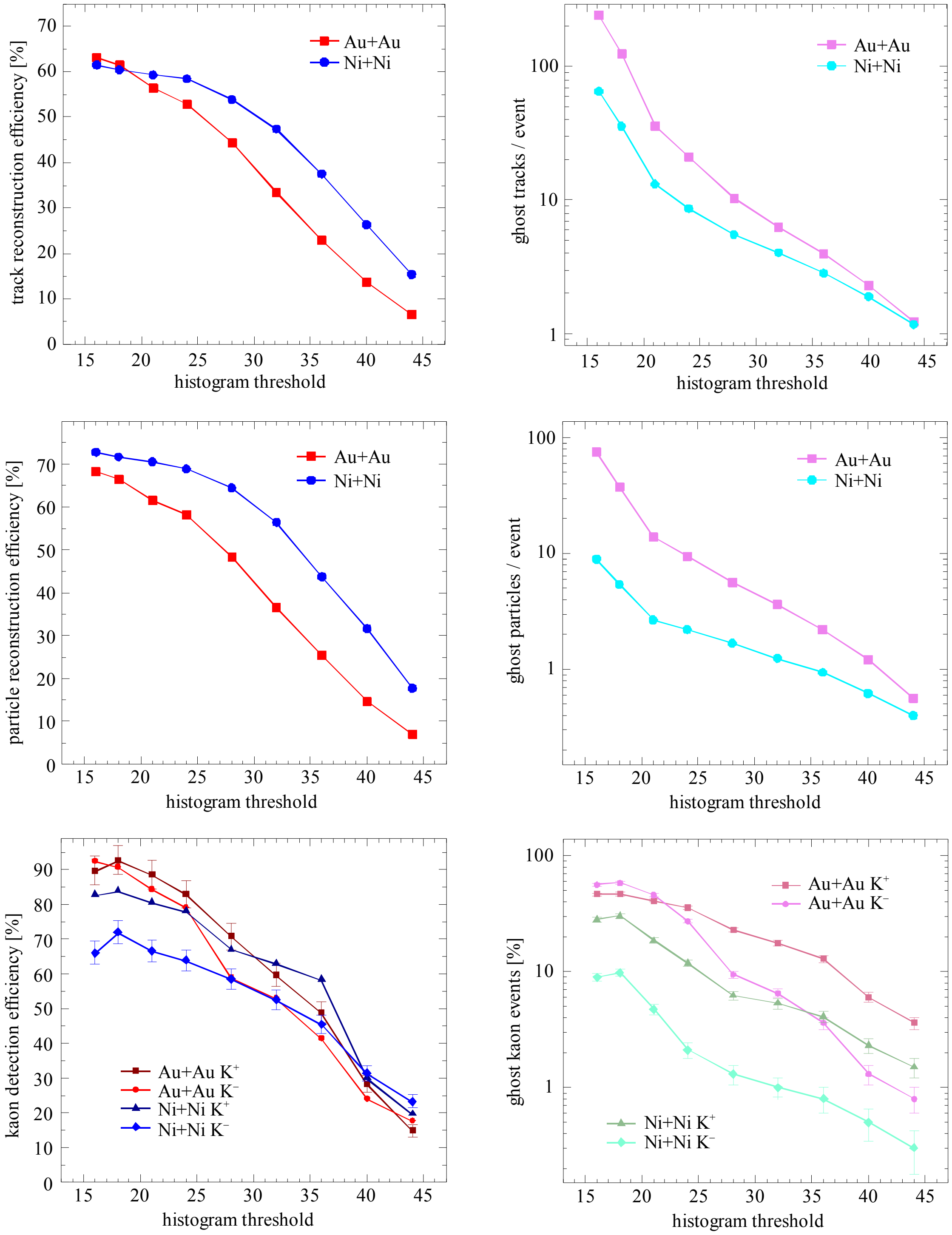


Fig. 7.2: Performance of the trigger algorithm for different histogram thresholds. First row: Track reconstruction efficiency (left) and erroneously reconstructed ghost tracks per event (right) in Ni + Ni and Au + Au events. Second row: Reconstruction efficiency for particles (after TOF matching, left) and emerging ghost particles per event (right). Third row: Kaon detection efficiency (left) and ghost kaon event rate (right) for K^+ and K^-. Note that tracks, particles, and kaons are only accepted within narrow boundaries, other objects are counted as ghosts. See text for further details. Given errors are purely statistical.

7.2.2 Particle Recognition

Besides the average DAQ time T needed for every event, which cannot be influenced by the trigger, and the trigger processing time t, which has been discussed in chapter 6.2.4, the trigger performance mainly depends on the particle reconstruction capabilities and the ghost particle suppression. As one might have already guessed, one quantity cannot effectively be maximized without negatively affecting the other, which was already the case for the tracking results.

It is important to keep in mind, that with the chosen particle reconstruction approach (extraction of track properties and matching with TOF data) only complete particles can be reconstructed. Fragments that have crossed the CDC, but have failed to produce a hit in the GRPC barrel, are automatically lost. For example, kaons that did not reach the GRPC barrel due to a prior decay cannot be reconstructed. Therefore, particle efficiencies are measured only in comparison to complete particles at the algorithm input. Even worse, the remaining track signatures represent an ideal source for ghost particles, because they are nevertheless available for the matching with other TOF datasets.

The particle reconstruction results are shown in the second row of fig. 7.2. Qualitatively, they do not differ much from the results of the tracking: in $Ni+Ni$ ($Au+Au$) events, the reconstruction efficiency starts at slightly above (below) $70\,\%$ for small values of the histogram threshold, and drops via $60\,\%$ ($45\,\%$) for intermediate to $18\,\%$ ($7\,\%$) for high thresholds. The number of produced ghost particles per event drops from 9 (75) via 1.5 (4.5) to 0.4 (0.55). The particle efficiencies are higher than the tracking efficiencies, because complete particles are more likely to be actually seen in the CDC, since an early decay is excluded for complete particles. The particle efficiencies were even higher, if the reaction products did not suffer a loss in kinetic energy along their paths, which is not accounted for in the search patterns of the Hough transform, nor in the positions contained in the TOF data included in the event description files.

The particle ghost rates are much lower than the rates of ghost tracks, because the matching with the TOF data eliminates most of the tracks that originate from background peaks in the Hough map. However, as events contain more and more tracks, the probability for accidental pile-ups in the Hough map increases. In addition, also more TOF hits become available as partners for the matching routine. As a consequence, the ghost suppression by TOF matching becomes less powerful in high-multiplicity events. Therefore, the ghost rates in $Ni+Ni$ and $Au+Au$ events differ by up to almost one order of magnitude.

Figs. 7.3 and 7.4 show the particle reconstruction performance using two alternative representations: particles, characterized by their masses or by their momenta and velocities, are compared at the input and the output of the tracking algorithm for different histogram thresholds. An ideal algorithm would be able to reproduce exactly the particles from the input channel. However, when proceeding from higher to lower thresholds, the number of reconstructed particles, both real and ghosts, increases quickly, and the peaks in the mass histograms as well as the lines in the p/v diagrams broaden significantly. This effect is in particular pronounced for the $Au+Au$ event data: a high detector occupancy offers a growing number of occasions for arbitrary matches. They turn out to be numerous especially with small momenta: since the produced ghost tracks cross a larger number of drift paths towards the sense wire planes, the constituting peaks in the Hough map are found beyond the threshold with a higher probability.

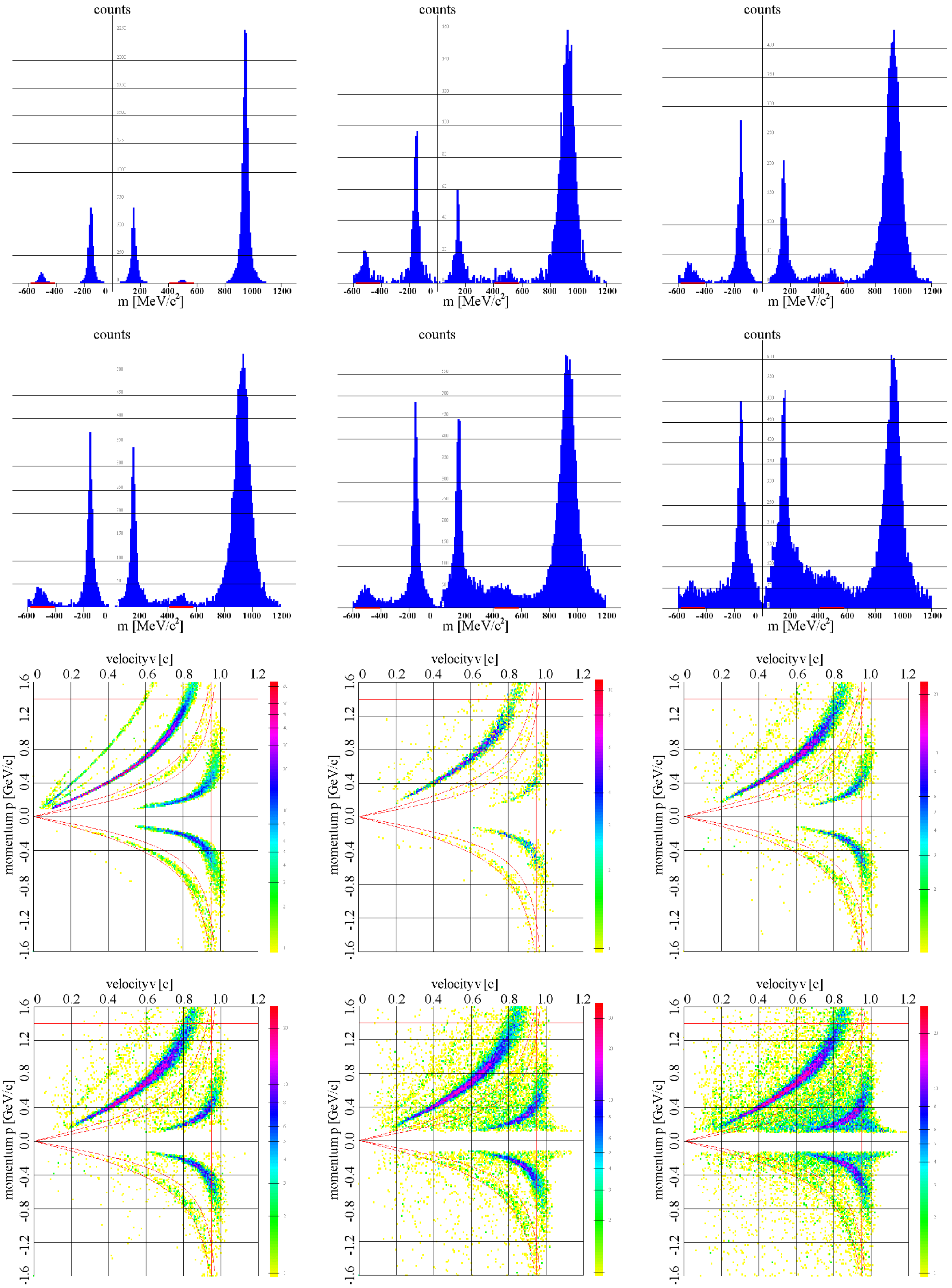


Fig. 7.3: Particle histograms for 2,000 Ni + Ni GEANT events. The upper two rows show reconstructed masses, the lower diagrams momentum vs. velocity for the same particles. The first plot of each group visualizes the algorithm input, the others are the output using thresholds of 44, 36, 28, 21, and 16, respectively, in the Hough map. Confer fig. 4.12 for an explanation of these diagrams.

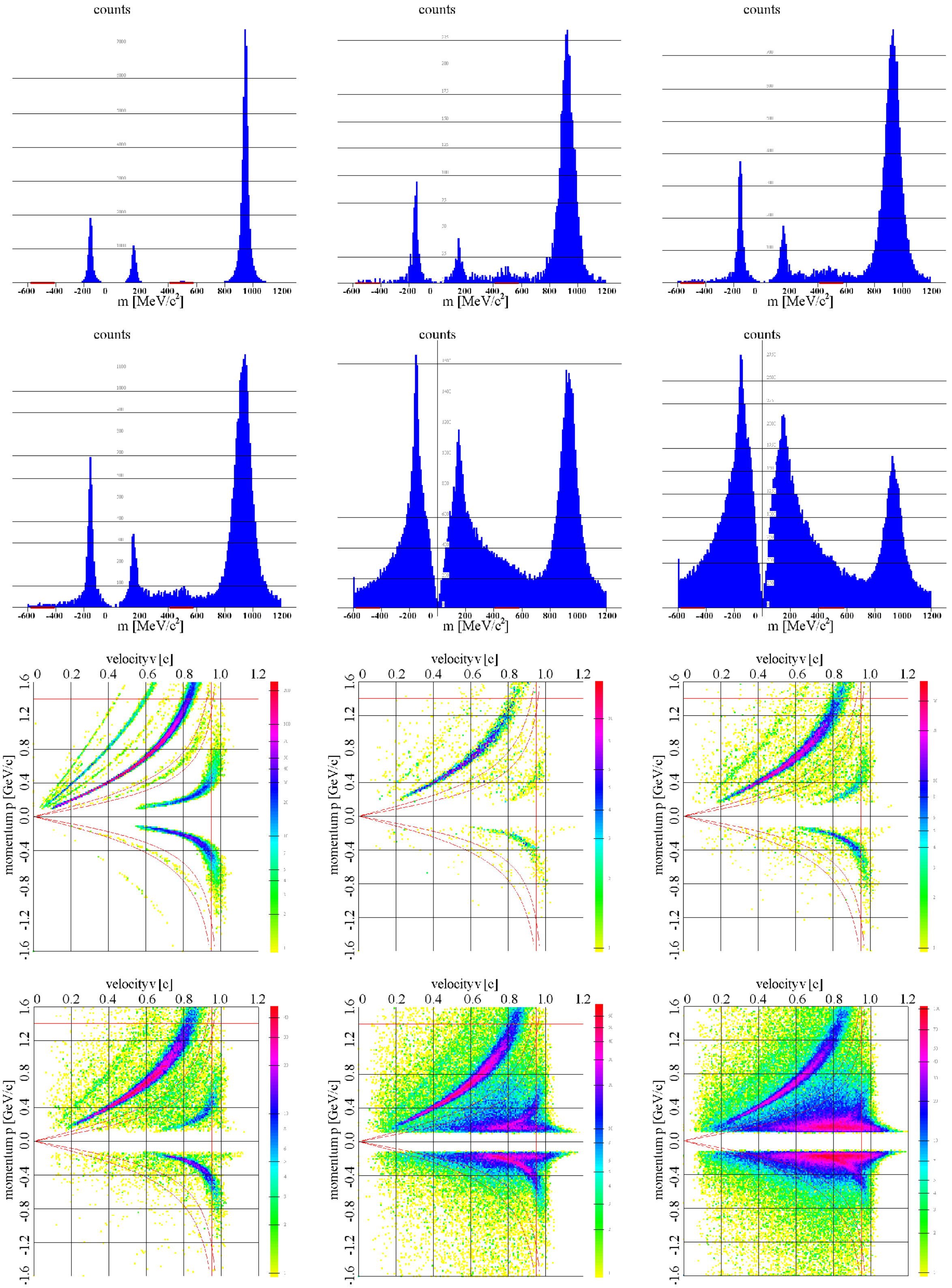


Fig. 7.4: Particle histograms for 2,000 Au + Au GEANT events. For further details see descriptions of figs. 7.3 and 4.12. For lower thresholds, the background begins to increase dramatically, and especially contains many low momentum ghost particles, that are caused by false combinatorial matches of CDC hits (confer fig. 4.11). See second row in fig. 7.2 for absolute values.

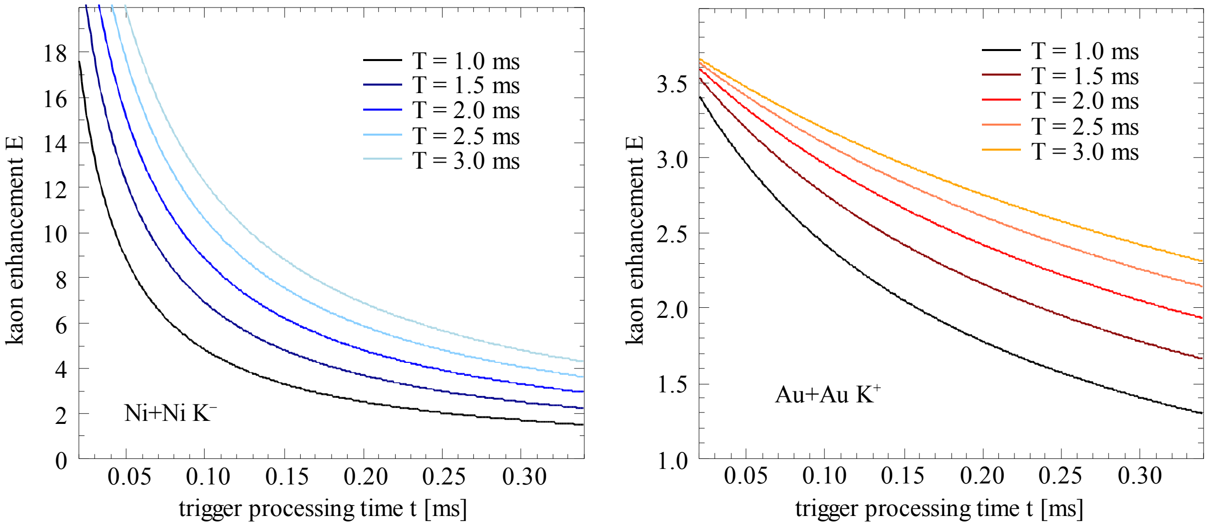


Fig. 7.5: Kaon enhancement E as calculated via eq. (3.5) for the efficiencies and ghost rates presented above (see fig. 7.2) using a histogram threshold of 32. The dependence of E on the trigger processing time t is shown for different DAQ times (readout + data compression) T. Especially for K^- in smaller systems the kaon enhancement depends strongly on the processing time. For Ni + Ni events, a DAQ time of T = 1.5 ms can be assumed, for Au + Au events T = 2 ms is more realistic [Leif04].

7.2.3 Trigger Performance

When the focus is switched to a certain particle species, like the rare kaons, a further level of abstraction is introduced, because in this case, events rather than tracks or particles are set into focus. Events that contain complete kaon datasets have to be identified only, while it stays irrelevant, whether the correct kaon properties have actually been found. Thus also errors of the second kind, i.e. when the kaon is lost and another match is mistaken for a kaon in the same event, positively contribute to the kaon efficiency (see chapter 3.2.1 for more details).

Kaon reconstruction efficiencies and events containing ghost kaons have been measured for K^+ and K^- separately. The results are shown in the lower two plots of fig. 7.2: starting between 82 % and 93 % (except around only 70 % for K^- in $Ni+Ni$ events) for small histogram thresholds, the kaon reconstruction efficiencies drop via between 55 % and 65 % for intermediate to around 20 % for high thresholds. The fraction of events erroneously associated with a kaon drops from 60 % (K^- in $Au+Au$), 45 % (K^+ in $Au+Au$), 30 % (K^+ in $Ni+Ni$) and 10 % (K^- in $Ni+Ni$) via between 20 % and 1 % down to between 3.5 % and 0.3 %. Higher efficiencies for K^+ reconstructions can be explained with the high ghost rate, because a ghost kaon in an event that contains a real kaon is also counted as a success. The ghost rates for the negatively charged K^- are significantly lower than those for the K^+, because thanks to the protons the majority of particles is positively charged, thus leaving much more track candidates for false positively charged matches than for negatively charged ones. Moreover, due to reconstruction inaccuracies, correct proton matches are often found within the kaon mass window as well. For low thresholds in $Au+Au$ events, ghost particles are frequently produced by arbitrary combinations of hits (cf. chapter 7.2.2 and fig. 7.4), which leads also to many negatively charged ghosts and thus increases the fraction of ghost K^- events.

resolution / TOF $[mm]$	pattern distribution	R_a $[mm]$	used CDC wires	CDC sampl. frequ. $[MHz]$	hist. filter	max. filter	hist. threshold	TOF matching window $\Delta\varphi$ $[mm]$	K mass window $[GeV/c^2]$	kaon enh. E
K^+ performance in $Au+Au$										
7.4	p_T	600	480	25	min3	sq5	36	20	80	2.38
7.4	p_T	600	480	25	4field	2diag	28	20	100	2.25
7.4	p_T	600	480	25	min3	sq5	36	16	80	2.20
7.4	p_T	200	320	25	min3	sq5	40	20	100	2.15
7.4	p_T	600	320	25	min3	sq5	36	20	100	2.15
K^- performance in $Au+Au$										
7.4	CDC	200	320	25	min3	sq5	40	20	100	5.39
7.4	p_T	200	320	25	min3	sq5	40	20	100	5.24
5.0	p_T	600	480	25	min3	sq5	28	20	100	4.58
7.4	p_T	200	480	25	min3	sq5	36	20	100	4.54
7.4	p_T	600	480	25	min3	sq5	32	16	80	4.50
K^+ performance in $Ni+Ni$										
7.4	p_T	600	480	25	min3	sq5	32	16	80	7.06
7.4	p_T	600	480	25	min3	sq5	32	16	90	6.77
7.4	p_T	600	480	25	min3	sq5	32	20	80	6.71
7.4	p_T	600	480	25	min3	sq5	36	18	80	6.48
7.4	CDC	600	320	25	min3	sq5	36	20	100	6.39
K^- performance in $Ni+Ni$										
7.4	TOF	200	320	25	min3	sq5	36	20	100	14.69
7.4	p_T	200	320	25	min3	sq5	32	20	100	13.41
7.4	CDC	200	320	25	min3	sq5	32	20	100	13.16
7.4	TOF	200	320	25	min3	sq5	28	20	100	13.08
7.4	TOF	200	320	25	min3	sq5	32	20	100	12.55

Table 7.2: List of the top performing out of around 110 parameter combinations for the enhancement of the K^+ and K^- contents in Au + Au and Ni + Ni experiments. The different pattern distributions are discussed in chapters 4.1.3 through 4.1.5, the filter layouts are explained in appendix A.3.2.2 and A.3.2.4. Some of these parameters influence the demands for hardware resources (see section 7.3.2) or the trigger execution time (see chapter 6.2.4). Note that most of the measurements were done using the parameter set 7.4 / p_T / 600 / 480 / 25 / min3 / sq5 / x / 20 / 100 only with different thresholds x, hence these parameters are found in the table most frequently.

According to eq. (3.5), the kaon enhancement E depends on the kaon efficiency and ghost rejection discussed above, on the trigger processing time t, and on the event DAQ time T (cf. chapter 2.3.2). Hence for a given T, the possible size of E is limited by the condition $t > 0$ and the efficiency and rejection results discussed above, which is expressed by relation (3.7). If also the performance is fixed to a desired value of $E \gg 1$, a narrow window of allowed values for t and possible values for E is obtained.

Fig. 7.5 shows possible values of E for K^- in $Ni+Ni$ (where after the preceding results the highest values are expected) and K^+ in $Au+Au$ events (representing the lowest values) as functions of T and t. According to this calculation, for which a standard set (see section 7.1.2) of algorithm parameters has been used, the achievable kaon enhancement is expected between 3 and over 13 in the $Ni+Ni$ case, and between 2 and 3 for $Au+Au$. Optimal adjustment of all parameters to the event characteristics (i.e. mainly track multiplicities) are certainly qualified to further improve the results obtained for the kaon enhancement E. Yet, the important result to be drawn from this investigation is that in addition to any algorithm's qualitative efficiency both T and t are quantities fundamental for the trigger performance.

7.2.4 Algorithm Parameters

An exhaustive search for the best performing LUT and processing parameters is virtually futile, because due to the number of parameters, that can be tuned individually, the search space comprises at least 13 dimensions (see the table in appendix A.1.4.1 for a complete list of parameters and options).

Table 7.2 shows the kaon enhancement E for some of the top performing parameter sets. They have been chosen out of simulations, in which only up to three parameters (including the histogram threshold) deviate from a standard set, so the parameter search is still far apart from being exhaustive. The results are grouped by kaon and event type and suggest, that a clear and safe maximum of the kaon enhancement can not be found in a defined region of the parameter space. Instead, the best performing parameter sets appear to be chosen arbitrarily, because they are scattered across a wide area in the parameter space. Moreover, statistical fluctuations constrain the accuracy of the measured kaon enhancements: the underlying kaon efficiency and ghost rejection performance is based on the simulation of only 2,000 events per parameter set (confer the statistical errors of the measured quantities in fig. 7.2).

Hence, the parameter search has been restricted to a search for suitable parameters, their comparison, and optimization of only the thresholds. The effects of a variation of the most important parameters on the trigger performance have been measured for each parameter individually by keeping all other parameters fixed. The results are plotted in fig. 7.6 and confirm the above result, that the choice of parameters does not predictably nor fundamentally influence the trigger performance. However, a few trends are nevertheless visible: E decreases

- if the signals of a larger number of CDC wires are used, which is obviously owed to the larger computing effort for the compilation of the Hough map (see chapter 6.2.4);
- if the sampling frequency of the CDC data drops down to only $12\,MHz$, because at this point, less sharply defined hits begin to form less sharply defined peaks in the Hough map;

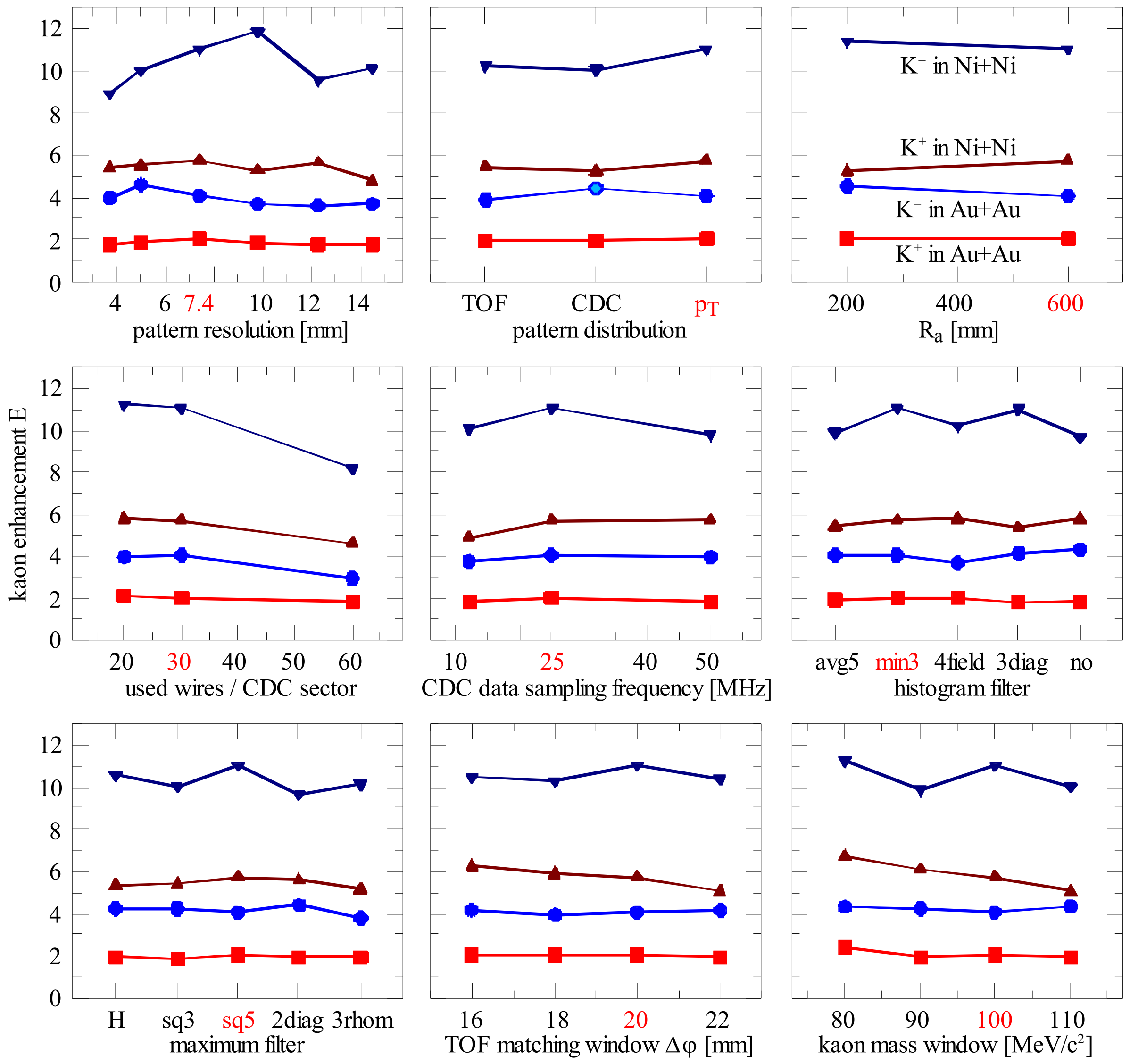


Fig. 7.6: Effects of the variation of a single parameter on the achievable kaon enhancement E in Ni + Ni and Au + Au experiments. The results are based on the same simulations as in table 7.2. The histogram threshold with the best result was selected for each combination. The standard parameters, that were used in all other cases, are marked in red. Confer table 7.2 for details about these parameters.

- and if the acceptance windows for the TOF matching or the kaon recognition get wide enough to significantly enhance also the acceptance of ghost particles and especially ghost kaons. However, this effect is only clearly visible for K^+ in $Ni + Ni$ events.

All other parameters show no significant direct influence on the overall trigger performance within the simulated range. However, better statistics (using maybe 20,000 instead of only 2,000 events per parameter set) and a more exhaustive search might still reveal parameter combinations with outstanding performance.

7.2.5 Performance Variations

The enrichment of kaons is not the only effect the trigger has on the data collected during an experiment. Very important for their analysis is also the knowledge, that the trigger performance is not constant across different event characteristics, the most obvious of them being the track multiplicity, which is connected to the system size, the beam energy and the impact centrality, and the momenta of the collected kaons.

The algorithm's kaon efficiency as a function of the kaon momentum is shown for different histogram filters in fig. 7.7. The same histogram threshold was applied for all measurements, and the statistics are again very low, especially for the kaons with the highest momenta. Yet independent of the filter applied, the trigger algorithm's kaon reconstruction efficiency reaches its maximum between 0.5 and $1.0\,GeV/c$ and decreases with growing kaon momentum. This effect is clearly more pronounced for K^- in $Ni+Ni$ than for K^+ in $Au+Au$ events. It is comparable in size for all histogram filters. It can be explained by the fact, that due to the increasing effect of the TOF resolution, the regions in a p/v diagram, where protons, kaons and pions can be found, broaden and begin to merge (see fig. 5.1 for an illustration). Thus more and more kaons miss the acceptance window, while at the same time other particles begin to meet it. As a consequence, the use of the trigger system will not only enhance an experiment's kaon yield, but will at the same time concentrate kaons of medium momenta and decrease the relative contents of high momentum kaons.

On the other hand, the number of ghost particles found in an event grows faster than linearly with the number of particles in that event. This originates from the fact, that, while the probability to be accepted as a kaon stays constant for a ghost particle coming from an arbitrary match, the opportunities for its creation multiply not only by the greater hit density in the CDC, but also by the enlarged matching probability with TOF datasets. Hence, the number of events, that contain a ghost kaon, grows rapidly with the track multiplicity. The consequences for the kaon enhancement factor E are shown in fig. 7.8 for the different histogram filters. When following the enhancement results to events with higher track multiplicities, its values drop dramatically at the beginning (by almost an order of magnitude up to multiplicities around 25); this effect slows down, so that E decreases by less than a factor of 2 from medium up to the highest multiplicities measured in $Au+Au$ events. This behavior can easily be derived from a connection of the quickly growing ghost rates with eq. (3.5), since assuming $N \gg 1$, $T \gg t$ and $gb \ll \{g, b\}$, it can be simplified to

$$E \approx 1 + \frac{g}{b} \tag{7.1}$$

(this is not a contradiction to eq. 3.8, because the latter was based on different assumptions). The comparatively small difference between the various layouts of the used histogram filters is due to the application of a uniform threshold; see the sixth plot in fig. 7.6, where the threshold was adapted to the respective filters, for comparison.

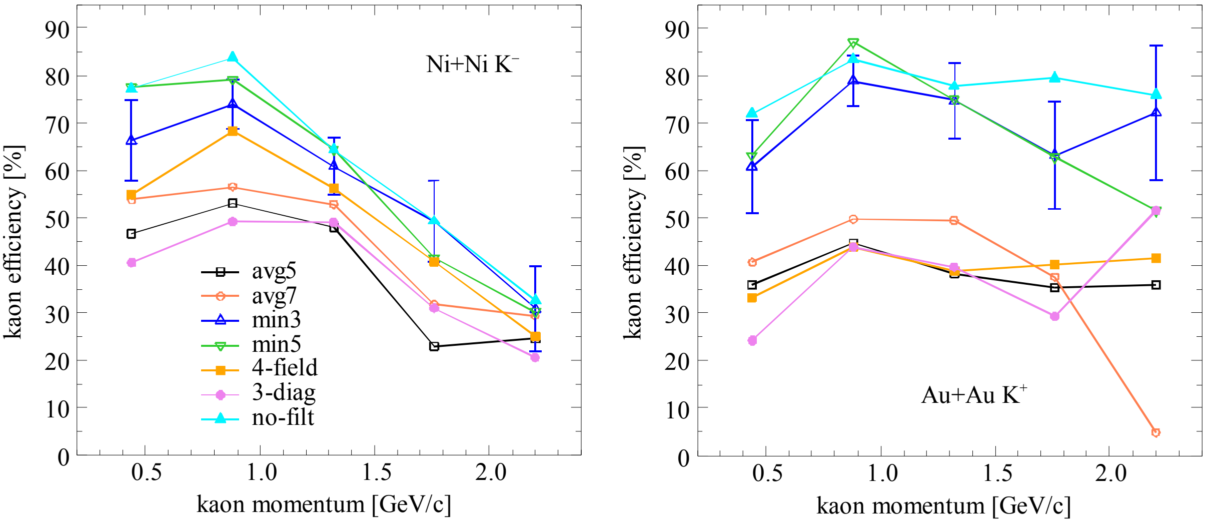


Fig. 7.7: Kaon reconstruction efficiency vs. kaon momentum for K^- in Ni + Ni (left) and K^+ in Au + Au (right) GEANT events. Results are shown for different histogram filters (see appendix A.3.2.2 for details of their layouts); data points are plotted at the upper boundaries of five kaon momentum bins. Parameters other than the filter have not been changed. Within statistical uncertainties (especially the rightmost data points represent only a small number of kaons; for clarity, statistical errors are shown representatively for a single graph only) the shape of the curves do not change significantly; absolute values are influenced by the maximum finder's threshold, because the results are optimized for the maximization of E (see fig. 7.8) instead.

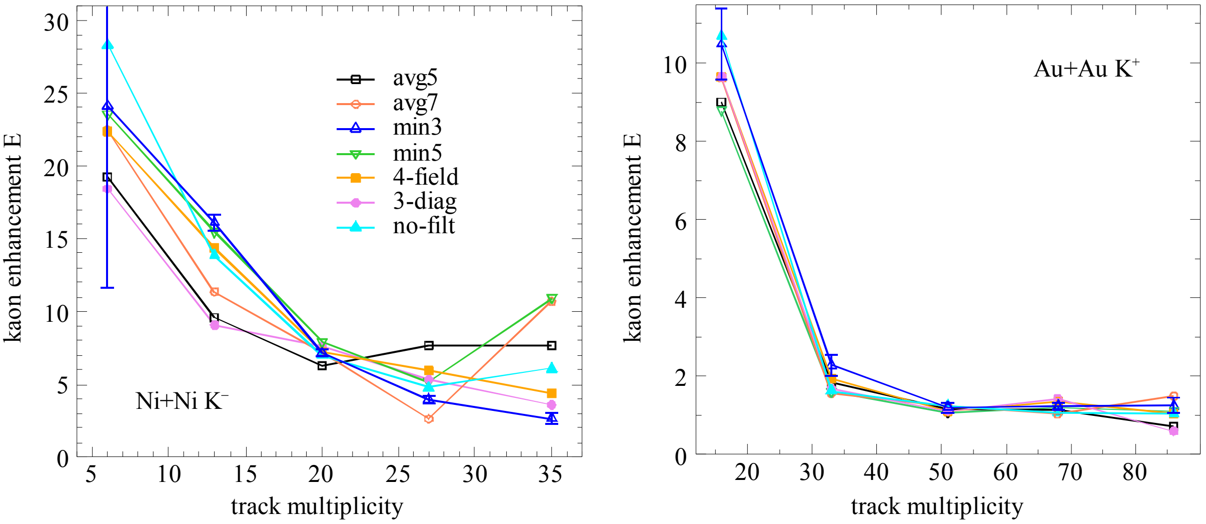


Fig. 7.8: Kaon enhancement E (see eq. 3.5) as a function of the events' track multiplicity for different histogram filter layouts (see appendix A.3.2.2 for details) for K^- in Ni + Ni (left) and K^+ in Au + Au (right) experiments. The positions of the data points are determined by the upper boundaries of five track multiplicity bins. Error bars are shown for one graph representatively. For remarks on parameters other than the used filter layout and the underlying statistics, see the description of fig. 7.7 above.

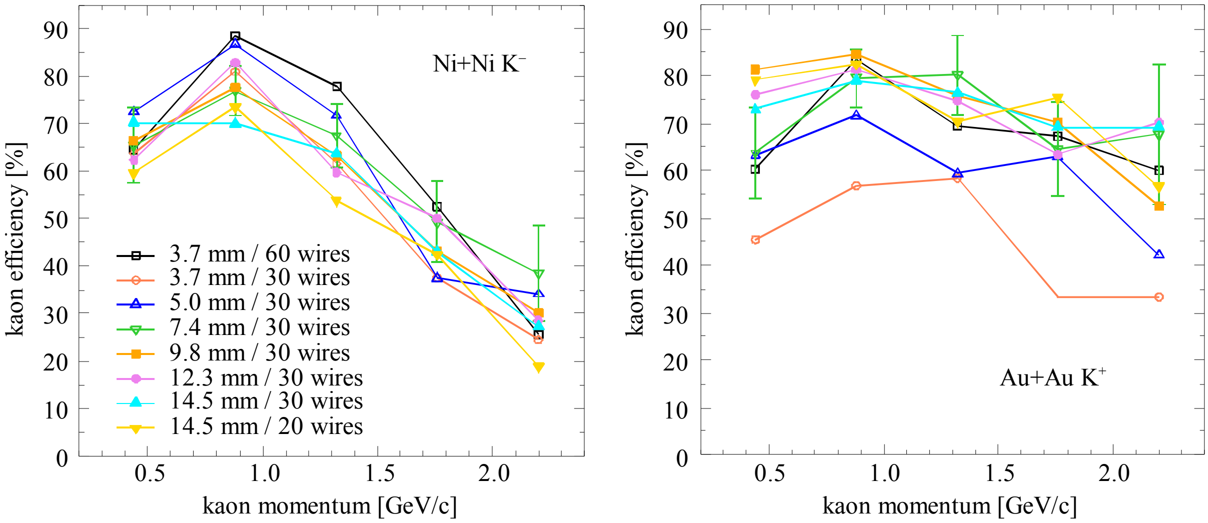


Fig. 7.9: Kaon reconstruction efficiency vs. kaon momentum for K^- in Ni + Ni (left) and K^+ in Au + Au (right) GEANT events. Efficiencies are shown for different histogram granularities (the distance between two neighboring tracks is measured in φ direction at the TOF detector position, see chapter 4.1.4 and appendix A.4.1.3 for details) and different numbers of used CDC wires per sector (60 stands for all wires, 30 for every second, and 20 for every third, respectively). For additional information regarding the positions of the data points, error bars, parameters other than pattern resolution and CDC wires and also the underlying statistics, see the description of fig. 7.7.

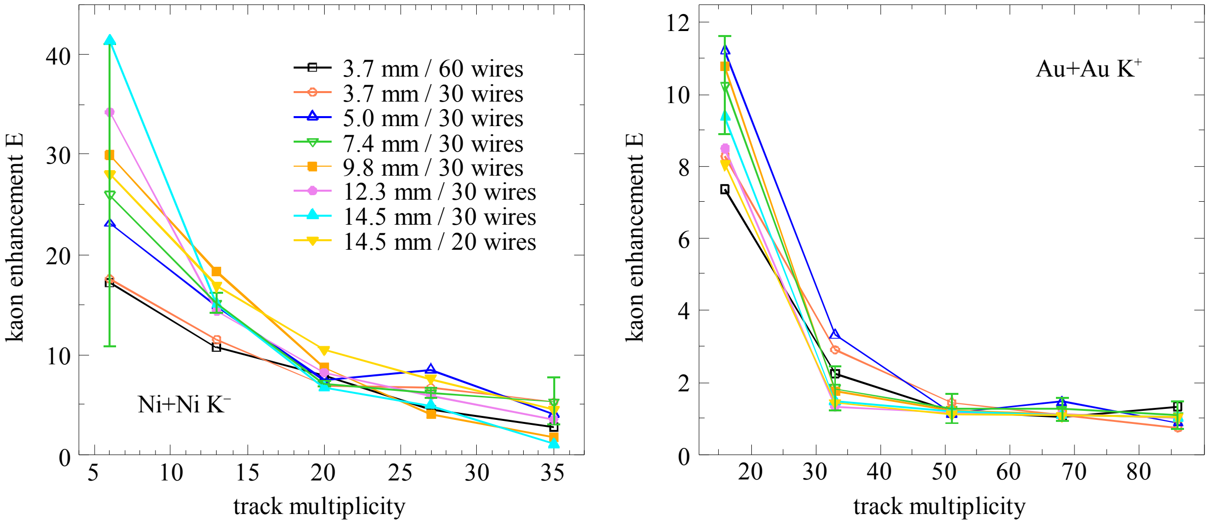


Fig. 7.10: Kaon enhancement E (see eq. 3.5) as a function of the events' track multiplicity for different histogram resolutions (measured at the TOF detector position) and different numbers of used CDC wires per sector. Results are given for K^- in Ni + Ni (left) and K^+ in Au + Au (right) experiments. The positions of the data points are determined by the upper boundaries of five track multiplicity bins. Error bars are shown for one graph representatively. For remarks on parameters other than pattern resolution and CDC wires and the underlying statistics, see the description of fig. 7.7.

7.3 Resource Optimization

The performance of the trigger algorithm has been investigated with different algorithm variants and lookup table constructions. However, hardware issues have not been addressed within this context. The topic of interest is the amount of hardware resources that a trigger algorithm with a given performance consumes, when it is implemented on the target architecture. This question is connected to a third collection of parameters, that has not been investigated in detail yet: the search pattern granularity and the resolution of the detector data. Both naturally influence the performance of the trigger algorithm, but also directly affect its hardware demands (confer also chapter 6.3 for details).

7.3.1 Performance Stability

The motivation for a reduction of the parameter or data resolution is the hope for a corresponding decrease of the necessary hardware resources or at least of the required cabling effort. The question, how far this reduction can be done, depends on its effects on the trigger performance and its variation over different kaon momenta and event properties.

Fig. 7.9 shows the dependence of the kaon recognition efficiency on the kaon momentum for different search pattern resolutions and detector image sizes. Within statistical uncertainties, the shape of the plots remains the same across all parameters, which can clearly be seen in particular in the $Ni+Ni$ case. The resolution of the search patterns or the number of used sense wires does apparently not influence the momentum dependence of the kaon reconstruction performance. The highest efficiencies are encountered at momenta between 0.5 and $1.0\,GeV/c$ (cf. fig. 7.7). The reason for the decrease of the efficiencies at higher momenta is the same as in the case of the dependence of the efficiency curves on the histogram filter layout (see section 7.2.5), i.e. the dominating influence of the time-of-flight resolution at high momenta.

The corresponding results for the kaon enhancement are displayed in fig. 7.10, classified according to the track multiplicities of the involved events. As has already been indicated in section 7.2.4, the kaon enhancement decreases slightly for higher histogram resolutions, which is due to the increasing computing effort for its compilation. This is again well visible in the $Ni+Ni$ plot, especially at lower track multiplicities. The fact, that this trend is not visible for $Au+Au$ events, can be interpreted as an additional benefit from the better accuracy provided by the higher histogram granularity (leading to a better peak separation for the maximum finder and yielding better track parameters for the TOF matching), which becomes important in particular for higher CDC and histogram occupancies. Anyway, the astonishing and important result of this investigation is worth being repeated: the trigger performance is, if at all, not negatively affected by a moderate reduction of the tracking accuracy or the fraction of tracking signals taken into account.

7.3.2 System Dimensions

On one hand, the system dimensions, i.e. the number of FPGA coprocessor boards needed to build the trigger system, depend on the details of the implementation (see chapter 6.2), in particular on the degree of parallelization (chapter 6.1). In modern FPGA devices, a large amount of logic and routing resources, that are necessary to connect the logic units, are available (see

search pattern resolution [*mm*]	histogram dimensions		required computing boards
	φ_a values	*C* blocks	
3.7	1,680	300	10
5.0	1,264	224	8
7.4	848	150	5
9.8	640	112	4
12.3	512	90	3
14.5	432	74	2

Table 7.3: Number of MPRACE coprocessor boards required for the processing of the tracking algorithm as a function of the search pattern resolution. The search pattern resolution is the distance along the TOF detector shell between neighboring tracks in φ_a and also between the two tracks with the highest momenta and the same charge in C direction. The number of C blocks is calculated using R_a = 60 cm and the TOF matching approach (see chapter 4.1.4). The number of computing boards depends on the number of available FPGA internal RAM blocks for the storage of these C blocks, and the resulting data width of the lookup table SRAM, that is necessary for their addressing (see chapter 6.3.2 for details).

wires per CDC sector	sampling frequency [*MHz*]	LUT address width	required input boards
60 (960 total)	50	15	7
	25	14	
	12	13	
30 (480 total)	50	14	4
	25	13	
	12	12	
20 (320 total)	50	14	3
	25	13	
	12	12	

Table 7.4: Number of MPRACE coprocessor boards required for the input of the discriminated CDC data as a function of the number of read-out sense wires. Both numbers are directly connected via the number of available input channels per MPRACE board (confer table 6.2 for possible configurations), which is assumed to be maximal in this case of dedicated data input processors. Together with the sampling frequency for the CDC data, the number of used wires also affects the depth of the lookup table and thus the width of the LUT addresses derived from the hit coordinates (see chapter 6.2.1).

chapter 3.4.3). Nevertheless, this question can definitely be answered only for the completed hardware version of the trigger algorithm, and gets less and less important, as newer FPGA devices become available.

On the other hand, the required system dimensions are influenced by the parameters, that have been investigated in the preceding section, in connection with the features of MPRACE. While the size of the lookup table stays relatively small in comparison to the on-board memory, restrictions do possibly apply to:

- the dimensions of the track histogram, because they must comply with the number and size of available FPGA-internal RAM blocks,
- the address width required for the incrementation of the histogram counters, since it is limited by the data width configuration of the on-board SRAM,
- the number of CDC channels to be used, because they must be compatible with the remaining input data width of the coprocessor boards.

The estimation of the required system dimension assumes, that data input from the CDC is done via separate boards, so the computing boards can be equipped with the maximum of 144 *bit* additional lookup table memory. The relevant details of this configuration concerning the resources that are responsible for the restrictions mentioned above have already been summarized in table 6.2. The resulting implications on the system dimensions are given in table 7.3 for the computing of the tracking algorithm and in table 7.4 for the input of the CDC data. The number of coprocessor boards needed for the execution of the trigger algorithm depends on the resolution of the search patterns, and the amount of boards required for the input of the CDC data is determined by the number of signal wires to be read out.

According to the results of the previous section, the trigger performance does not decrease when reducing the input bandwidth or the histogram granularity. Thus, an effective kaon trigger for FOPI can be built using the coarsest resolutions, which requires as few as only 5 or 6 coprocessor cards of the MPRACE type.

Chapter 8: Conclusion

In the preceding chapters, a kaon trigger for the FOPI experiment has been introduced in detail, and its performance has been evaluated by comprehensive simulations. This chapter assesses the results and points out their implications with respect to the integration of the system into the experiment. It closes with suggestions about further refinements and improvements of the algorithm and its implementation, in case they should turn out to be useful.

8.1 System Performance

The performance of the trigger system proposed in this thesis has been investigated under many respects, such as track and particle reconstruction efficiencies and their variation with event and particle properties. Special focus has been directed towards the global performance in terms of the strangeness enhancement factor *E*, based on which an assessment of the benefits of the kaon trigger can be made. It directly relates the strangeness yield that can be achieved in a triggered experiment to the yield that can be expected if the trigger is not enabled.

8.1.1 Benchmark Results

The trigger's track reconstruction efficiency (chapter 7.2.1) reaches up to around 60 % both for events of the types $Au+Au @ 1.5\,AGeV$ and $Ni+Ni @ 1.93\,AGeV$. Depending on the desired minimum quality of the reconstructed tracks, which is realized by the setting of a threshold in the track histogram, the reconstruction efficiency may drop down to 50 % in $Ni+Ni$, or 40 % in $Au+Au$ events. It is important to remark, that these values are not corrected for early decays (cf. chapter 7.1.1). This means that the reconstruction efficiencies for complete particle tracks starting from the center-of-collision and reaching the time-of-flight barrel are significantly higher.

Particles represent the second level of reconstruction, because both the tracking information and the time-of-flight data are needed. The particle reconstruction efficiencies (chapter 7.2.2) reach 60 % to beyond 70 % in $Ni+Ni$, and 40 % to slightly below 70 % in $Au+Au$ events. Although the second reconstruction stage has also only limited efficiency and thus diminishes the overall performance, particle efficiencies are nevertheless higher than the track efficiencies. This is due to the fact that early decays play no role in the determination of the particle efficiencies.

The third reconstruction level are kaon events. The associated efficiencies (chapter 7.2.3) are around 60 % for K^{-}, and 70 % for K^{+}, respectively, in both types of events, or up to 90 %, if the reconstruction of lower quality particle tracks is allowed. The highest values are achieved for kaons of momenta around $1\,GeV/c$, above this value efficiencies decrease with increasing momentum (chapter 7.2.5). The emergence of ghost kaons leads to a fraction of erroneously accepted events, which is as low as $1\,\%$ of all processed events, if K^- are triggered in $Ni+Ni$, but reaches $20\,\%$ for K^+ in $Au+Au$ events.

These results clearly indicate, that the proposed trigger algorithm can enhance the fraction of collected events, that contain strangeness. Yet nothing is said about the total strangeness enhancement, which can be expressed as a factor, that relates kaon yields in an experiment carried out with the kaon trigger to yields in experiments without trigger system. Unfortunately, this kaon enhancement depends on further quantities, like the trigger execution time, the data acquisition time per event, and the kaon production probability. For the assumptions that formed the basis of the calculations (see chapter 7.1.2), the kaon enhancement factor (chapters 7.2.3 and 7.2.4) reaches between 2 for K^+ in $Au+Au$ and 11 for K^- in $Ni+Ni$. Both values can further be increased by $20-40\,\%$, if the algorithm parameters are carefully optimized.

8.1.2 Comparison and Discussion

Recognition efficiencies for particles of $40-60\,\%$, or for kaon events of $60-70\,\%$ (as obtained for the medium histogram thresholds, that were used to optimize the total kaon yield) appear low at first glance, especially if they are compared to typical tracking efficiencies of e.g. around $80\,\%$, as measured for the STAR third level trigger algorithm [Adle03] (see chapter 3.1.4).

However in all performance measurements discussed above, focus has been set on the maximization of the kaon enhancement instead. Therefore, the results cannot directly be compared to the pure reconstruction efficiencies obtained elsewhere. The measured kaon reconstruction or event selection efficiencies are merely typical values of the intermediate tracking results. Actually, the parameter variations applied in chapter 7.2.5 and the results plotted in fig. 7.9 have clearly shown, that these efficiencies (e.g. for kaons with $p\approx 1\,GeV/c$, that had the best statistics) can easily be pushed beyond $80\,\%$, while on the other hand efficiencies of not much above $40\,\%$ can lead to kaon enhancements of similar magnitudes.

Nevertheless, some additional reasons can be named, why the tracking efficiency cannot be expected to exceed the results that are measured in other experiments.

- Due to FOPI's time consuming FADC readout (cf. chapter 2.3.2) the tracking is bound to use only a fraction of the produced tracking data. This fraction is only two-dimensional (after combining drift times and wire positions, the coordinates ρ and φ, but no z or ϑ coordinates are available), and an evaluation of pulses, possibly based on their shapes or charge deposits, is impossible. Hence, the tracking results can only serve as input to the matching with the time-of-flight data, which only then reveals the polar angle ϑ and makes a particle identification possible. False matches that could otherwise be rejected by comparison of z coordinates, cannot be suppressed by this method.
- A second source for tracking misalignments is the high share of low p_T fragments in the FOPI data, that is simply a consequence of the small radius of the CDC. These particles suffer from a noticeable energy loss along their paths, so that the projections of their tracks are no longer circles. Moreover, the change in a track's radius depends on the particle

species. As the result, tracking of low energy particles is inevitably subject to an enhanced error rate. Furthermore, the trajectories produced by low p_T fragments span across a significant section of the CDC subdetector. This makes the application of local tracking algorithms (cf. chapter 3.1.2) more difficult, and is another source of errors for global approaches. For example in the Hough transform, the presence of low p_T tracks is responsible for background entries in the Hough map, that are distant from the evolving peak for that track.

- The left/right ambiguity of the CDC (see chapter 2.2.1) automatically leads to 50 % of fake hits. These additional hits increase the computing effort (cf. chapter 6.2.4), and in addition are responsible for a high noise level in the track histogram. Though due to the tilting of the CDC's sense wire planes mirror tracks can easily be distinguished from real tracks, the distribution of their contribution to the background noise is not uniform. In events with a high detector occupancy, these elongated background elevations often coincide and appear to be peaks that originate from a real track (see fig. 4.8). If the histogram threshold is accordingly adapted to this situation in order to keep the number of ghost tracks at a tolerable level, the reconstruction efficiency for real tracks is already reduced as well.
- Reconstruction efficiencies are always reduced, if coincidences have to be exploited, like it is done when an inter-detector matching is performed. In order to identify particles based on their invariant masses (obtained from their momenta and velocities), a matching between the CDC and GRPC subdetectors must be performed. Yet this matching is highly error-prone: it is only one-dimensional, because it has to rely on an extrapolated φ coordinate alone; no z coordinates or orientation vectors are available for a refined selection of appropriate matching partners. For comparison: the matching of kaon tracklets from the ATLAS TRT and SCT subdetectors [Bain00] (see chapter 3.1.4) results in a reconstruction efficiency of around 60 – 70 %, which is clearly below an efficiency of almost 90 % for the TRT tracks alone.

Consequently, the results obtained for the efficiency and enhancement performance of the projected kaon trigger system are by all means very encouraging. The most significant conclusions that can in addition be drawn from the simulations are:

- The values of the kaon enhancement E (chapter 7.2.3 and fig. 7.5) determine the trigger's contribution to the desired increase of statistics in all measurements that involve kaons. Depending on the type of event (see section 8.2.1 for more details), the enhancement is between a factor of 2 (e.g. K^+ in $Au+Au$ experiments) and more than an order of magnitude (as measured for K^- in $Ni+Ni$ events).
- The knowledge of the kaon response with respect to their momenta (chapter 7.2.5 and fig. 7.7) is crucial for the extraction of the correct observables from any experiment, because the implications of the trigger's event selection have to be computationally reversed.
- The dependence of E on the average track multiplicity (fig. 7.8) has to be counted back as well in order to restore the original multiplicity distribution and to extract valid kaon production rates.

8.2 Trigger Application

It has been shown, that an effective trigger system for the kaon enhancement in FOPI experiments can be built with a reasonable amount of existing hardware. The field of application as well as the necessary effort to integrate the trigger into the existing data acquisition, is more closely described in this section.

8.2.1 FOPI Experiments

It has already been pointed out, that the strangeness enhancement due to the integration of the trigger system strongly depends on the type of experiment, that is performed with the FOPI detector system. Reasons for this are the slightly reduced tracking efficiency for heavier systems (i.e., reactions, that produce a higher average track multiplicity), and the larger processing time for the histogramming of events, which is also owed to the higher detector occupancy. The main reason, however, is the dramatically enlarged rate of erroneously accepted events, that emerges with the analysis of high multiplicity reactions.

In experiments with light systems, like the $Ni+Ni@1.93\,AGeV$ data that was used for the performance simulations, a strangeness enhancement in the order of 6 for K^+ and 11 for K^- is achieved. This great improvement can either be used to reduce the beam-time that has to be requested for an experiment, or to boost the quality of the physics results in terms of the attainable gain in statistic significance.

The simulation results for a heavy system like $Au+Au@1.5\,AGeV$ suggests a strangeness enhancement in the order of 2 for K^+ and 4 for K^-. It is subject to further investigation, whether the increase in strangeness yield, especially for the K^+, can overcompensate the loss in statistic significance that is due to the additional sources of errors introduced by the necessary corrections for the kaon response with respect to their momenta, and by the multiplicity selection that is done by the trigger.

8.2.2 Implementation Costs

A high-level trigger system for kaons in FOPI experiments, that meets the requirements with respect to decision latency and classification quality, can be implemented employing around 5 to 6 coprocessors of the MPRACE type, that are used for data acquisition and execution of the trigger algorithm (see chapter 7.3.2). Given these dimensions, the costs of the necessary hardware – coprocessors, host computers, and connections to the FOPI subdetectors – will be surprisingly low.

An MPRACE coprocessor board can be assembled for less than 5,000 €, including all necessary parts [Kuge04]. Depending on the features and performance of the host system, several of these boards can share the same PC (see [Müll04] for performance measurements of up to 4 MPRACE boards on a single PCI bus). A computer, that provides $64\,bit/66\,MHz$ PCI slots in order to exploit the full capabilities of MPRACE, is available today for less than 2,000 €. If for the time being cables and additional interfaces are neglected, the total hardware costs for the trigger system can therefore be kept below 36,000 €, even if only 2 coprocessors share the same host PC.

8.2.3 Integration Effort

Besides the installation of the trigger processors, some effort has to be made in order to implement the trigger algorithm and to integrate the system into the existing data acquisition. This comprises the following steps:

- The trigger algorithm has to be implemented using a hardware description language (HDL) such as VHDL, in order to be executable on an FPGA processor. Since a similar tracking algorithm, whose core is also based on a lookup-table guided Hough transform, has been developed for ATLAS [Sess00] and successfully been tested on the same hardware [Hink03], this procedure is essentially straightforward. In case of MPRACE, a powerful software environment, consisting of the device drivers pciDriver (for the PCI bridge chip) [Müll04b] and uelib (for configuration and control of the coprocessor) [Müll04, Hink04], are available to support this task.

- In order to supply the trigger processor boards with the necessary event data, the setup of appropriate interfaces is required as well. The physical interface to the time-of-flight data has already been described in chapter 2.3.3, the interface to the track data can basically consist of a plug, however possibly with hundreds of cables conducted in parallel (cf. chapter 6.3.2 and table 7.4). Alternatively, the number of necessary link channels can be reduced by forming a cascade using simple multiplexers.

- The simulation framework produces a classification of the processed events by determining their kaon status (see “kaon” in the table in appendix A.1.3.1 for details). This result already includes the trigger decision, which alone can also be made available for unknown event data. For the final trigger system, this computation can even be simplified, because it may abort as soon as a kaon has been identified.

- In addition to the processing modules and the hardware implementation, several communication channels between the different modules of the trigger system must be foreseen:

  The start signal provided by the existing data acquisition has to be made available to the trigger in order to make it establish the acceptance of the CDC data and to let it assign the correct drift times. The trigger decision in turn has to be communicated back to the data acquisition, so that it can be reset for the next event, or start the full data readout for the current event. Confer the timing diagram in fig. 3.2: the decision, whether the data acquisition time is augmented by a few percent, or if most of the time is saved for events rejected by the trigger, is made at this point.

  The trigger's submodules, i.e. the coprocessor cards and the host PCs, need to be supplied with the full active hit data from the CDC, and the full time-of-flight data from the GRPC barrel. Hence, a communication between the trigger modules has to be set up in order to distribute these data. Finally, a connection between the tracking coprocessors and their host PCs must transfer the input data for the time-of-flight matching and particle identification modules. Since both communication channels can easily be pipelined after the generation of the data to be distributed or transferred, no considerable additional latency is introduced that way.

  The global trigger decision can easily be generated by combining the results from the different submodules using a simple logic OR operation.

8.3 Outlook

In addition to the success that is achievable with the trigger system up to this point, some improvements to the algorithm and the processing are nonetheless possible. As a matter of course, this will result in an enlarged computing effort, which in turn requires an extension of the hardware dimensions. Within reasonable limits, these enhancements may become important anyway: they can provide a clear enhancement of the trigger performance, which is of significance in particular for experiments with heavier systems.

8.3.1 Algorithm Enhancement

The main problem of the tracking algorithm is the number of emerging ghost kaons, that quickly increases with growing average charged particle multiplicity in an event. However, the reconstruction accuracy and also the suppression of accidentally formed track patterns can be improved by additional algorithm modules. To prevent a resulting rise of processing time from eating up the gained advantage, further parallelization of the algorithm will also be necessary (see next section). Possible algorithm extensions are:

- The parameters assigned to a track candidate can be refined by using Kalman filtering (or an equivalent least square's fit) in order to find the best fitting trajectory for the associated detector hits. This will improve the precision of the particle properties extracted later, and will also reduce the probability of misalignments in the time-of-flight matching. An additional advantage of this step is the possibility to reject track candidates, if the mean least square's deviation of the hit positions from the fitted track exceeds a certain threshold (see fig. 4.11 for an illustration). It is a reasonable assumption, that this will greatly reduce the number of ghost tracks.
- The energy loss along a particle's track leads to a noticeable decrease of the track radius along the path of flight of particles with very low momenta. This fact cannot fully be accounted for in the lookup tables of the Hough transform, because this change in radius is different for different particle species and even for different polar emission angles. However, the lookup tables (and also the Kalman filtering mentioned above) could be optimized for kaons in order to improve the quality of their reconstruction. Moreover, after the matching with the time-of-flight data a further plausibility check can be performed, which rejects a reconstructed particle (or tries to find a better time-of-flight record), if the observed change in the track's curvature is incompatible with the preliminarily assigned particle species.
- Many ghost tracks can also be avoided, if the mirror hits are deleted. Once a particle is successfully reconstructed, its assigned hits can no longer contribute to other tracks, nor can the corresponding mirror hits. To use this fact in practice for the implementation, it is necessary to assume, that the highest histogram peaks belong to real particle tracks. Consequently, there are two possible implementations:

  The first version performs the maximum search in the histogram a few times in succession, each time deleting the contributions from the hits assigned to the found tracks and also their mirrors, and lowering the threshold for the next repetition. By the time the threshold reaches its final value, only a low occupancy is left in the histogram, so that the number of extracted ghost tracks will be correspondingly small. According to eq. (6.1), the processing time for the tracking algorithm is dominated by the time needed for the histogramming. If the maximum search is done in e.g. four passes, this time is increased by around 75 %, because the additional reverse histogramming needs not be done for the hits belonging to tracks found during the last pass. The comparatively small time for the maximum search is of course multiplied by the number of performed passes. For the example with 20 CDC wires per sector, 60 tracks per event, 400 φ_a values and 66 MHz design frequency, the processing time for the complete tracking increases from 42 μs (see table 6.1) to 90 μs.

The second implementation variant eliminates mirror hits after the initial tracking. A list of track candidates is stored along with the assigned hits and sorted by their number. The best track is then accepted, and its hits and their mirrors are removed from the list of hits of the other candidates, if they erroneously have also been assigned also to them. This procedure stops, if no track candidate remains with enough hits to fulfill the quality requirement otherwise guaranteed by the histogram threshold alone. The implementation of this method is yet costly, because hit addresses have to be compared for each initial track candidate in parallel, and the number of remaining hits must be monitored for each candidate. However if done carefully, the processing time increases by roughly the time needed for the histogramming, because most of the initial hits and their mirrors have to be deleted successively.

8.3.2 Implementation Improvements

The possible enhancements of the tracking algorithm discussed above prompt for the introduction of further levels of parallelization into the hardware implementation in order to compensate for the corresponding increase of the processing time. This higher parallelization once more leads to an augmented demand for processing hardware. On the other hand, an enhanced parallelization alone may instead reduce the processing time and will consequently improve the trigger performance as well. A higher parallelization can be introduced as follows:

- The splitting of the *C* blocks, that was suggested in chapter 6.2.2 as an improvement of the histogramming performance, can serve as a basis also for further accelerations. Instead of implementing a pipeline for the histogram filter and the maximum finder for every *C* block, separate pipelines can as well be implemented for every split subblock. Since the time needed to process the complete pipeline is proportional to the size of the blocks, these additional pipelines can reduce the processing time accordingly. Depending on the complexity of the implemented histogram filter, the number of *C* blocks, that can be handled by each processor, might be reduced by this method.
- The time needed for histogramming is proportional to the number of hits in the CDC. Since only one hit can be processed at a time, because its transformation comprises the complete width of the lookup table as well as every *C* block of the histogram, the only way to further parallelize the histogramming is to divide the CDC into a couple of sectors and compute the Hough transform for each sector separately. This will multiply the dimensions of the histogram and the memory width needed for lookup tables by the number of introduced sectors, and will thus also increase the number of required coprocessors. Yet the computing time is reduced as determined by the number of hits found in the CDC sector with the highest hit multiplicity alone. The subsequent filtering and maximum finding must be modified accordingly: the input data for these pipelines need to be generated as the sum from the corresponding values of the different partial histograms.

8.3.3 Other Fields of Application

The benefits of a trigger system for FOPI are not limited to the enhancement of the charged kaon contents in the experiments. The FOPI collaboration is currently proposing an experiment that uses a pion beam on nuclear targets for the investigation of associated strangeness production [FOPI04] via

$$\pi^- A \;\rightarrow\; \Lambda K^0 X\,, \tag{8.1}$$

with $A \in \{C, \ldots, U\}$. From reactions of this extremely small system, only a few particle tracks are expected. Moreover, most of the events can be rejected, since they will not contain any strangeness. Hence, this experiment can greatly benefit from a high-level trigger system (cf. section 8.2.1). The current pion production and beam pipe at GSI are capable of supplying a pion luminosity L_π in the order of $10^3 s^{-1}$ at the target position in "Cave B". The resulting interaction rates can easily be handled by the updated FOPI detector and data acquisition alone. However, the pion luminosity will be increased by three orders of magnitude ($L_\pi \approx 10^6 s^{-1}$) within the next three years [Herr04]. The trigger system is required in order to be able to exploit this enhancement.

Neutral strange particles (e.g. K^0, Λ) cannot be observed directly with the FOPI detector; moreover, K^0 and Λ decay after a few centimeters. However, they can be identified from their decay products,

$$K_S^0 \;\rightarrow\; \pi^+ \pi^- \quad \text{and} \quad \Lambda \;\rightarrow\; p \pi^- . \tag{8.2}$$

Since they are not produced at the target position, no correct time-of-flight information is available. Nevertheless, the sum of their transverse momenta p_T is directly accessible from the radii of their trajectories, and thus the so-called transverse mass of the strange hadron

$$m_T \;=\; \sqrt{\frac{p_T^2}{c^2} \;+\; m_0^2} \tag{8.3}$$

can be calculated.

In order to use the trigger system for the detection of these neutral particles, the trigger algorithm has to be modified accordingly. The tracking must be able to identify trajectories, that do not originate from the center-of-collision, and to reconstruct their common vertex. In return, the matching with the TOF data can be omitted. Yet the selection, evaluation and implementation of an appropriate trigger algorithm is left as a topic for future research.

Chapter 9: Summary

Hadronic matter under extreme conditions is an important research field in modern physics, that promises to extend our knowledge about its basic structure and properties. It can only be explored by means of heavy ion collisions. Such studies are performed at GSI using the FOPI detector, that offers a large geometric acceptance for the detection of charged particles emitted during the nuclear reactions. The focus is in particular directed towards the rarely produced strange particles, especially the K^+ and K^- mesons. However, long run-times of these experiments are required in order to be able to extract physical quantities of acceptable significance from their analysis. Even higher statistics will be necessary for answering the many still open questions in this physics area.

This problem can be solved by employing a high-level trigger system. It is able to abort the data acquisition at an early stage, if no kaon signatures are found in the current event's data. The development and evaluation of a kaon trigger for the FOPI experiment is the subject of this thesis. It must deliver a valid event classification within about $100\,\mu s$ after the collision. Due to the slow readout of the tracking information from the 960 sense wires of the drift chamber CDC, its decision can only rely on the arrival times of pulses from these wires, that are available as discriminated signals within $5\,\mu s$ after the event. In addition, the time and position data from the new GRPC time-of-flight barrel are required for particle identification up to momenta of $1.5\,GeV/c$ and beyond, and available after around $50\,\mu s$. Moreover, the trigger system must be cost-effective and able to be integrated into the current data acquisition within reasonable effort.

Within the scope of this work, an online trigger algorithm has been developed and optimized for maximum strangeness enhancement during a FOPI experiment. It reconstructs particle tracks from the CDC data using a lookup table guided discrete Hough transform, and completes the particle information by matching with appropriate datasets from the GRPC barrel. The Hough transform is a proven approach for the identification of parametrized objects in binary images and is known to be robust against noisy and missing data. In the first step, counters in a track histogram are incremented according to the transformation rule for every CDC hit. The second step associates local maxima above a threshold with the parameters of tracks in the detector image. The tracks' reciprocal radius and their orientation in a plane perpendicular to the beam axis were chosen for the parametrization.

The implementation, that has been elaborated for this thesis, comprises a number of peculiarities. Two of them are connected to the calculation of the lookup table. A track's orientation is not expressed by its starting angle, but rather by its angle at an intermediate distance from the reaction center. The resulting peak in the track histogram gets almost symmetric and its tail more locally confined, thus making it easier to identify it correctly. Second, different granularities of the reciprocal radius parameter were implemented in the histogram in order to investigate the implications on the trigger performance. They are motivated by an equal distribution of track curvatures, a regular granularity for the matching at the GRPC barrel, and an even coverage of the CDC. An additional enhancement consists in the application of a local histogram threshold in order to suppress the background in areas with a high occupancy.

The high demands the track recognition and the supplying of the necessary detector data make on the computing power and the input bandwidth require a careful selection of a qualified hardware platform. Only an architecture like the FPGA and PCI based coprocessor MPRACE developed at Mannheim University offers the desired possibilities in a manageable and affordable system. MPRACE provides up to 144 I/O channels for the input of the detector data. Fast on-board SRAM with a data width of up to $288\,bit$ is ideally suited for the storage of the lookup table. The FPGA supplies 96 blocks with together $1.7\,Mbit$ of additional memory which is needed for the counters of the track histogram. It can carry out 28,000 basic operations simultaneously at $125\,MHz$ or more to perform the histogramming and extract the positions of the local maxima.

The proposed hardware implementation achieves a high level of parallelization by dividing the lookup table, the track histogram and also the logic for the maximum extraction into blocks assigned to equal reciprocal track radii. Since the lookup table contains for every hit exactly one counter address per possible radius, and since all values for this hit can be retrieved as a single large data word, all blocks can be processed in parallel. Moreover, a new hit can be processed, and also a new histogram value can be fed into the maximum recognition with every clock cycle. This behavior is achieved by massive pipelining.

A software implementation of the trigger algorithm has been embedded into a comprehensive simulation environment. This framework provides several displays that visualize in- and outputs of the algorithm, as well as intermediate results of the particle recognition. They serve as an event display, and are important for debugging of the algorithm and the detectors. From the comparison of algorithm in- and outputs, performance results are extracted. The events are classified according to their track multiplicity and an existing kaon's momentum. The kaon enhancement factor E, that relates the kaon yields obtainable in experiments with and without the trigger system, is calculated for different event classes. This analysis also involves kaon occurrence, data acquisition performance, and trigger execution time. A summary of the statistical results – efficiencies, ghost rates and kaon enhancements – is also visualized by the framework.

Performance measurements were done using simulated event data of $Ni+Ni @ 1.93\,AGeV$ and $Au+Au @ 1.5\,AGeV$ experiments. The algorithm settings were optimized to achieve large kaon enhancements E. Kaon events are correctly classified with probabilities between 55% (for K^- in $Ni+Ni$) and 65% (K^+ in $Au+Au$) while at the same time between 1% and slightly above 10% of the events are erroneously associated with either type of kaon. The kaon efficiencies are largest for kaon momenta around $1\,GeV/c$. Using pessimistic assumptions concerning the parameters for the calculation of E, like a fast data acquisition and a slow FPGA clock, the resulting kaon enhancement factors reach between 2 (K^+ in $Au+Au$) and 11 (K^- in $Ni+Ni$); they drop dramatically with increasing average track multiplicity of the investigated events.

The performance was also measured against different search pattern and detector resolutions. It turned out, that the benefits of a better accuracy and ghost suppression are just compensated by the required larger computing effort. Thus, a considerably coarse granularity of the track parameters in combination with the use of only a fraction of the CDC sense wires ($1/3 \triangleq 320$ wires) proved sufficient for building an effective kaon trigger. It can be implemented on only 5 to 6 MPRACE boards, so that the hardware for a complete trigger system – ignoring cabling and additional interfaces but including state-of-the-art host PCs – is available for less than 36,000 €. If a larger system is used, additional improvements to the trigger algorithm, that aim at a more powerful ghost suppression and a higher degree of parallelization of the processing, can be introduced in order to further enhance the trigger performance.

In experiments with light systems, like the examined $Ni + Ni @ 1.93\, AGeV$ data, the use of the proposed kaon trigger system is mandatory; it is able to increase the experiment's kaon and anti-kaon yields by easily around an order of magnitude. However in heavy systems like $Au + Au @ 1.5\, AGeV$, kaon enhancements between 2 and 4 can be expected. The additional statistical error, that is owed to the uncertainties in the dependences of the trigger response on kaon momenta and track multiplicities, must be taken into account for the evaluation of the trigger's benefits in these experiments. The application of a refined algorithm at the expense of an increased hardware demand is anyhow possible. A trigger using a modified algorithm for the detection of neutral strange particles is also of great interest for the pion beam experiment proposed for the near future.

References

[Acke03] K.H. Ackermann et al., *STAR detector overview*, Nucl. Instrum. Meth. **A499** (2003) 624-632

[Adap94] Adaptive Solutions, Inc., *CNAPS* (1994)

[Adle03] C. Adler et al., *The STAR Level-3 trigger system*, Nucl. Instrum. Meth. **A499** (2003) 778-791

[Afan02] S.V. Afanasiev and the NA49 Collaboration, *Energy Dependence of Pion and Kaon Production in Central Pb+Pb Collisions*, Phys. Rev. **C66** (2002) 054902 arXiv:nucl-ex/0205002

[Aich00] J. Aichelin, C. Hartnack, *Kaon production at subthreshold and threshold energies*, J. Phys. **G27** (2000) 571-578 arXiv:nucl-th/0011050

[Aich91] J. Aichelin, *"Quantum" molecular dynamics - a dynamical microscopic n-body approach to investigate fragment formation and the nuclear equation of state in heavy ion collisions*, Phys. Rep. **202** (1991) 233-260

[ALIC00] The ALICE Collaboration, *ALICE Technical Design Report of the Time Projection Chamber* (2000) CERN/LHCC/2000-001

[ALIC01] The ALICE Collaboration, *ALICE Technical Design Report of the Transition Radiation Detector* (2001) CERN/LHCC/2001-021

[ALIC02] The ALICE Collaboration, HLT Working Group, *ALICE High-Level Trigger - Conceptual Design* (2002)

[ALIC96] The ALICE Collaboration, *A Large Ion Collider Experiment - Technical Proposal* (1996) CERN/LHCC/95-71

[ALIC99] The ALICE Collaboration, *ALICE Technical Design Report of the Inner Tracking System (ITS)* (1999) CERN/LHCC/99-12

[Ande03] M. Anderson, *The STAR time projection chamber: a unique tool for studying high multiplicity events at RHIC*, Nucl. Instrum. Meth. **A499** (2003) 659-678

[Andr01] A. Andronic and the FOPI Collaboration, *Transition from in-plane to out-of-plane azimuthal enhancement in Au+Au collisions*, Nucl. Phys. **A679** (2001) 765-792 arXiv:nucl-ex/0008007

[Andr98] A. Andronic, *Aspecte noi privind procesele de tip colectiv in interactia ionilor grei la energii relativiste*, PhD thesis, IFIN-HH Bucharest (1998)

[Andr99] A. Andronic and the FOPI Collaboration, *Elliptic flow and equation of state in heavy ion collisions at SIS energies*, Nucl. Phys. **A661** (1999) 333-336

[Anze95] G. Anzellotti et al., *Totem: a highly parallel chip for triggering applications with inductive learning based on the reactive tabu search*, Int. J. Modern Phys. **C6** (1995) 555-560

[ATLA02] The ATLAS Collaboration, *ATLAS High-Level Trigger, Data Acquisition and Controls - Technical Design Report* (2002) `ATLAS TDR-016`

[ATLA94] The ATLAS Collaboration, *ATLAS Technical Proposal for a General-Purpose pp Experiment at the Large Hadron Collider at CERN* (1994) `CERN/LHCC/94-43`

[ATLA97] Tha ATLAS Collaboration, *ATLAS Inner Detector - Technical Design Report* (1997) `CERN/LHCC/97-16+17`

[Bain00] J. Baines et al., *B-Physics Event Selection for the ATLAS High Level Trigger*, ATLAS DAQ Note (2000) `CERN/ATL-DAQ-2000-031`

[Bair01] A. Baird et al., *A Fast High Resolution Track Trigger for the H1 Experiment*, IEEE Trans. Nucl. Sci. **48** (2001) 1276-1285 `arXiv:hep-ex/0104010`

[Bass98] S.A. Bass, C. Hartnack, N. Amelin et al., *Microscopic Models for Ultrarelativistic Heavy Ion Collisions*, Prog. Part. Nucl. Phys. **41** (1998) 225-370 `arXiv:nucl-th/9803035`

[Bern95] V. Bernard, N. Kaiser, U.-G. Meißner, *Chiral Dynamics in Nucleons and Nuclei*, Int. J. Mod. Phys. **E4** (1995) 193-346 `arXiv:hep-ph/9501384`

[Best92] D. Best, *Tracking mit der Hough-Transformation für die Zentrale Driftkammer des GSI-4pi-Experiments*, diploma thesis, Mainz University (1992)

[Best96] D. Best, *K+ Production in the System Ni+Ni at an Incident Energy of 1.93 AGeV*, Proc. 12th Winter Workshop on Nuclear Dynamics, Snowbird, Utah (1996)

[Best97] D. Best and the FOPI Collaboration, *K+ production in the reaction 58Ni+58Ni at incident energies from 1 to 2 AGeV*, Nucl. Phys. **A625** (1997) 307-324 `arXiv:nucl-ex/9704005`

[Bock90] R.K. Bock et al., *Data analysis techniques for high energy physics experiments*, Cambridge University Press (1990) `ISBN 0-521-34195-7`

[Bros00] O. Brosch et al., *ATLANTIS - A Hybrid FPGA/RISC Based Re-configurable System*, Proc. 7th Reconfigurable Architectures Workshop, Intl. Parallel & Distrib. Proc. Symp., Cancun, Mexico (2000) 890-897

[Brow92] G.E. Brown et al., *A novel mechanism for kaon condensation in neutron star matter*, Phys. Lett. **B291** (1992) 355-362

[Brow92b] G.E. Brown, *Equation of state and the physics of stellar collapse*, Proc. 1st Symp. on Nucl. Phys. in the Universe, Oak Ridge, Tennessee, USA (1992) `ISBN 0-750-30279-8`

[Brow94] G.E. Brown, H.A. Bethe, *A Scenario for a Large Number of Low-Mass Black Holes in the Galaxy*, Astrophys. J. **423** (1994) 659

[Croc00] P. Crochet, N. Herrmann, K. Wisniewski, Y. Leifels and the FOPI Collaboration, *Sideward flow of K+ mesons in Ru+Ru and Ni+Ni reactions near threshold*, Phys. Lett. **B486** (2000) 6-12 `arXiv:nucl-ex/0006004`

[Croc00b] P. Crochet and the FOPI Collaboration, *Results from FOPI on strangeness in nuclear matter at SIS energies*, Proc. 5th Intl. Conf. on Strangeness in Quark Matter, Berkeley, California (2000)

[Croc98] P. Crochet and the FOPI Collaboration, *Strangeness and in-medium effects in heavy ion collisions at SIS energies*, Proc. Workshop Nucl. Matter in Different Phases and Transitions, Les Houches, France (1998) 361-374 `ISBN 0-792-35660-8`

[Croc99] P. Crochet and FOPI collaboration, *Strangeness and in-medium effects in heavy ion collisions at SIS energies*, Nucl. Phys. **A661** (1999) 460-463

[Dani01] P. Danielewicz, *Flow and the equation of state of nuclear matter*, Nucl. Phys. **A685** (2001) 368-383 `arXiv:nucl-th/0009091`

[Dani02] P. Danielewicz, *Hadronic Transport Models*, Acta Phys. Polon. **B33** (2002) 45-64 `arXiv:nucl-th/0201032`

[Dani85] P. Danielewicz, G. Odyniec, *Transverse momentum analysis of collective motion in relativistic nuclear collisions*, Phys. Lett. **B157** (1985) 146-150

[Dani99] P. Danielewicz, *Flow and equation of state in heavy-ion collisions*, Nucl. Phys. **A661** (1999) 82-92 `arXiv:nucl-th/9907098`

[Davi99] C. David, C. Hartnack, J. Aichelin, *On the flow of kaons produced in relativistic heavy ion collisions*, Nucl. Phys. **A650** (1999) 358-368 `arXiv:nucl-th/9805017`

[Devi00] A. Devismes, *K+ Production in Heavy Ion Collisions at 1.5 AGeV*, PhD thesis, Darmstadt University of Technology (2000)

[Devi00b] A. Devismes and the FOPI Collaboration, *Results from FOPI on strangeness production and propagation in hot and dense nuclear matter*, Proc. Int. Workshop XXVIII Hadrons in Dense Matter, Hirschegg, Austria (2000) 104-111 `http://crunch.ikp.physik.tu-darmstadt.de/nhc/hirschegg/Proceedings2000.html`

[Devi02] A. Devismes, *Results from FOPI on strangeness production and propagation at SIS energies*, J. Phys. **G28** (2002) 1591-1598 `arXiv:nucl-ex/0202017`

[Ecke95] G. Ecker, *Chiral perturbation theory*, Prog. Part. Nucl. Phys. **35** (1995) 1-80 `arXiv:hep-ph/9501357`

[Flor00] W. Florkowski, W. Broniowski, *In-Medium modifications of hadron properties and chemical freeze-out in ultra-relativistic heavy-ion collisions*, Proc. Int. Workshop XXVIII Hadrons in Dense Matter, Hirschegg, Austria (2000) 275-279 `http://crunch.ikp.physik.tu-darmstadt.de/nhc/hirschegg/Proceedings2000.html`

[FOPI03] The FOPI Collaboration, *The FOPI Collaboration at SIS/ESR* (2003) `http://www-aix.gsi.de/~fopiwww/people/`

[FOPI04] The FOPI Collaboration, *Associated Strangeness Production with Pion Beams (Proposal)* (2004)

[Früh87] R. Frühwirth, *Application of Kalman filtering to track and vertex fitting*, Nucl. Instrum. Meth. **A262** (1987) 444-450

[Fuch01] C. Fuchs et al., *Probing the nuclear equation of state by K+ production in heavy ion collisions*, Phys. Rev. Lett **86** (2001) 1974-1977 `arXiv:nucl-th/0011102`

[GEAN93] CERN, *GEANT - Detector Description and Simulation Tool* (1993) `http://wwwasd.web.cern.ch/wwwasd/geant/`

[GSI01] GSI, *An International Accelerator Facility for Beams of Ions and Antiprotons (Conceptual Design Report)* (2001) 493-499

[Gyul91] M. Gyulassy, M. Harlander, *Elastic tracking and neural network algorithms for complex pattern recognition*, Comp. Phys. Comm. **66** (1991) 31-46

[Hagi02] K. Hagiwara et al., *Review of Particle Physics*, Phys. Rev. D **66** (2002) 010001 `http://pdg.lbl.gov/`

[Hahn03] H. Hahn et al., *The RHIC design overview*, Nucl. Instrum. Meth. **A499** (2003) 245-263

[Hans88] M. Hansroul, H. Jeremie, D. Savard, *Fast circle fit with the conformal mapping method*, Nucl. Instrum. Meth **A270** (1988) 498-501

[Hard99] S.J. Hardy, *The effective electron mass in core-collapse supernovae*, Astron. Astrophys. **342** (1999) 614-621 `arXiv:astro-ph/9811466`

[Harr96] J.W. Harris, B. Müller, *The Search for the Quark-Gluon Plasma*, Ann. Rev. Nucl. Part. Sci. **46** (1996) 71-107 `arXiv:hep-ph/9602235`

[Hart00] C. Hartnack, J. Aichelin, *The production of K+ mesons in heavy ion reactions*, Proc. Int. Workshop XXVIII Hadrons in Dense Matter, Hirschegg, Austria (2000) 129-136 `http://crunch.ikp.physik.tu-darmstadt.de/nhc/hirschegg/Proceedings2000.html`

[Hart03] C. Hartnack, H. Oeschler, J. Aichelin, *What determines the K- multiplicity at energies around 1-2 AGeV?*, Phys. Rev. Lett. **90** (2003) 102302 `arXiv:nucl-th/0109016`

[Herr04] N. Herrmann, *private communication* (2004)

[Herr96] N. Herrmann and the FOPI Collaboration, *Particle Production and Flow at SIS Energies*, Nucl. Phys. **A610** (1996) 49c-62c `arXiv:nucl-ex/9610007`

[Hink00] C. Hinkelbein et al., *Prospects of FPGAs for the ATLAS LVL2 Trigger*, ATLAS DAQ Note (2000) `CERN/ATL-DAQ-2000-006`

[Hink03] C. Hinkelbein et al., *Timing measurements of some tracking algorithms and suitability of FPGA's to improve the execution speed*, ATLAS DAQ Note (2003) `CERN/ATL-DAQ-2003-026`

[Hink04] C. Hinkelbein, *Design of Control Software for Reconfigurable Co-processors*, PhD thesis, Mannheim University (in preparation)

[Hink99] C. Hinkelbein et al., *Pattern Recognition in the TRT for the ATLAS B-Physics Trigger*, ATLAS DAQ Note (1999) `CERN/ATL-DAQ-99-012`

[Högl95] H. Högl, A. Kugel, J. Ludvig, R. Männer, K.-H. Noffz, R. Zoz, *Enable++: A Second Generation FPGA Processor*, Proc. 3rd IEEE Symp. on FPGAs for Custom Computing Machines, Napa, CA, USA (1995) 45-53 `http://www-li5.ti.uni-mannheim.de/publications/`

[Högl97] H. Högl et al., *MicroEnable: A FPGA-based Coprocessor proposed for Preprocessing in ATLAS*, Proc. Computing in High Energy Physics, Berlin, Germany (1997) 182-184

[HONE97] The H1 Collaboration, *The H1 detector at HERA*, Nucl. Instrum. Meth. **A386** (1997) 310-347

[HONE97b] The H1 Collaboration, *The tracking, calorimeter and muon detectors of the H1 experiment at HERA*, Nucl. Instrum. Meth. **A386** (1997) 348-396

[Houg62] P.V.C. Hough, *Methods and means for recognising complex patterns*, U.S. Patent **3 069 654** (Dec. 1962)

[Jähn01] B. Jähne, *Digitale Bildverarbeitung*, Springer Verlag (2001) `ISBN 3-540-41260-3`

[Kalm60] R.E. Kalman, *A New Approach to Linear Filtering and Prediction Problems*, Trans. ACME - J. Basic Engineering **D82** (1960) 35-45

[Klef92] F. Klefenz, R. Zoz, K.-H. Noffz, R. Männer, *The ENABLE Machine - A Systolic Second Level Trigger Processor for Track Finding*, Proc. Comp. in High Energy Physics, Annecy, France (1992) 799-802 `http://www-li5.ti.uni-mannheim.de/publications/`

[Klef93] F. Klefenz et al., *Track Recognition in 4 µs by a Systolic Trigger Processor Using a Parallel Hough Transform*, IEEE Tr. Nucl. Sci. **40** (1993) 688-691 `http://www-li5.ti.uni-mannheim.de/publications/`

[Klug97] A. Kluge, *The Hardware Track Finder Processor in CMS at CERN*, PhD thesis, Vienna University of Technology (1997)

[Ko96] C.M. Ko, G.Q. Li, *Medium effects in high energy heavy-ion collisions*, J. Phys. **G22** (1996) 1673-1726 `arXiv:nucl-th/9611027`

[Koch86] P. Koch, B. Müller, J. Rafelski, *Strangeness in relativistic heavy ion collisions*, Phys. Rep. **142** (1986) 167-262

[Köhn97] J.K. Köhne, *Realization of a second level neural network trigger for the H1 experiment at HERA*, Nucl. Instrum. Meth. **A389** (1997) 128-133

[Kola95] H. Kolanski, *Application of Artificial Neural Networks in Particle Physics*, Nucl. Instrum. Meth. **A367** (1995) 14-20

[Kreß02] T. Kreß, *Elliptischer Fluß von Protonen und leichten Kernen in Au+Au-Reaktionen bei Strahlenergien zwischen 400 und 1490 AMeV*, PhD thesis, Darmstadt University of Technology (2002)

[Kube98] T. Kuberka, *Leistungsanalyse von Gleitkommaarithmetik auf FPGA-Prozessoren an Hand der Implementierung des SPH-Algorithmus (Schritt 1)*, diploma thesis, Mannheim University (1998)

[Kuge01] A. Kugel, *RACE-1 Product Brief* (2001) `http://www-li5.ti.uni-mannheim.de/fpga/race/`

[Kuge04] A. Kugel, *private communication* (2004)

[Kuts99] R. Kutsche, *Untersuchungen der in-medium Eigenschaften von K0S-Mesonen und Lambda-Hyperonen an der Produktionsschwelle*, PhD thesis, Darmstadt University of Technology (1999)

[Leif02] Y. Leifels, *Event data created with the GEANT package* (2002)

[Leif04] Y. Leifels, *private communication* (2004)

[Li95] G.Q. Li, C.M. Ko, B.A. Li, *Kaon flow as a probe of the kaon potential in nuclear medium*, Phys. Rev. Lett. **74** (1995) 235-238 `arXiv:nucl-th/9410017`

[Li95b] G.Q. Li, C.M. Ko, *Subthreshold kaon production and the nuclear equation of state*, Phys. Lett. **B349** (1995) 405-410 `arXiv:nucl-th/9411024`

[Li97] G.Q. Li, G.E. Brown, C.-H. Lee, C.M. Ko, *Strangeness production and flow in heavy-ion collisions*, Proc. 13th Winter Workshop on Nuclear Dynamics, Marathon, Florida (1997) `arXiv:nucl-th/9703040`

[Li97b] G.Q. Li, C.-H. Lee, G.E. Brown, *Kaons in dense matter, kaon production in heavy-ion collisions, and kaon condensation in neutron stars*, Nucl. Phys. **A625** (1997) 372-434 `arXiv:nucl-th/9706057`

[Li97c] G.Q. Li, C.-H. Lee, G.E. Brown, *Kaon production in heavy-ion collisions and maximum mass of neutron stars*, Phys. Rev. Lett. **79** (1997) 5214-5217 `arXiv:nucl-th/9711002`

[Li98] G.Q. Li, G.E. Brown, *K-/K+ ratios in relativistic heavy-ion collisions*, Phys. Rev. **C58** (1998) 1698-1705 `arXiv:nucl-th/9804013`

[Lisa95] M.A. Lisa and the EOS Collaboration, *Radial Flow in Au + Au Collisions at E = (0.25 - 1.15)A GeV*, Phys. Rev. Lett. **75** (1995) 2662-2665

[Lutz00] M.F.M. Lutz, E.E. Kolomeitsev, *Effective chiral theory of kaon-nucleon scattering*, Proc. Int. Workshop XXVIII Hadrons in Dense Matter, Hirschegg, Austria (2000) 117-128 `arXiv:nucl-th/0004021`

[Meer02] D. Meer et al., *A Multifunctional Processing Board for the Fast Track Trigger of the H1 Experiment*, IEEE Trans. Nucl. Sci. **49** (2002) 357-361 `arXiv:hep-ex/0107010`

[Müll04] M. Müller, *Evaluation of an FPGA and PCI Bus based Readout Buffer for the Atlas Experiment*, PhD thesis, Mannheim University (in preparation)

[Müll04b] M. Müller, *private communication* (2004)

[Nave03] C.R. Nave, *HyperPhysics* (2003) `http://hyperphysics.phy-astr.gsu.edu/hbase/hph.html`

[Nogu86] Y. Noguchia, A. Ono, *Global pattern recognition in layered track chambers*, Nucl. Instrum. Meth. **A253** (1986) 27-37

[Perl86] J. Perl et al., *Track finding with the Mark II/SLC drift chamber*, Nucl. Instrum. Meth. **A252** (1986) 616-620

[Petr02] M. Petrovici et al., *A large-area glass resistive-plate chamber with multistrip readout*, Nucl. Instrum. Meth. **A487** (2002) 337-345

[Pich94] A. Pich, *Quantum Chromodynamics*, Proc. 1994 European School of High-Energy Physics, Sorrento (1994) `arXiv:hep-ph/9505231`

[Pich99] A. Pich, *Aspects of Quantum Chromodynamics*, Proc. ICTP Summer School in Particle Physics, Trieste, Italy (1999) `arXiv:hep-ph/0001118`

[Pink95] C. Pinkenburg, *Flußeffekte geladener Pionen im System Au+Au bei Ep = 1 AGeV*, PhD thesis, Frankfurt University (1995)

[Plet99] C. Plettner, *Strangenessproduktion bei kleinen transversalen Impulsen und mittleren Rapiditäten in der Reaktion 96Ru + 96Ru @ 1.69 AGeV*, PhD thesis, Dresden University of Technology (1999)

[Rama95] U. Ramacher et al., *SYNAPSE-1: A High-Speed General Purpose Parallel Neurocomputer System*, Proc. 9th Int. Parallel Proc. Symp., Santa Barbara, CA, USA (1995) 774-781

[Reis02] W. Reisdorf, *The Quest for the Equation of State of Nuclear Matter in the Energy Range (0.1-2) AGeV*, Act. Phys. Pol. **B33** (2002) 107-124

[Ritm95] J. Ritman and the FOPI Collaboration, *The FOPI Detector at SIS/GSI*, Nucl. Phys. B (Proc. Suppl.) **44** (1995) 708-715

[Ritm95b] J. Ritman and the FOPI Collaboration, *On the transverse momentum distribution of strange hadrons produced in relativistic heavy ion collisions*, Z. Phys. **A352** (1995) 355-357 `arXiv:nucl-ex/9506002`

[Schm02] R. Schmidt, *Status of the LHC*, Proc. 8th European Particle Accelerator Conference EPAC, La Vilette, Paris, France (2002) 6 p. `CERN-LHC-Project-Report-569`

[Seng00] P. Senger for the KaoS Collaboration, *Strange mesons in dense matter*, Proc. Int. Workshop XXVIII Hadrons in Dense Matter, Hirschegg, Austria (2000) 96-103 `http://crunch.ikp.physik.tu-darmstadt.de/nhc/hirschegg/Proceedings2000.html`

[Seng01] P. Senger, *Strangeness and charm production in nucleus-nucleus collisions at beam energies near the thresholds*, Proc. Int. Workshop XXIX Structure of Hadrons, Hirschegg, Austria (2001) 259-273 `http://theory.gsi.de/hirschegg/2001/Proceedings/`

[Seng99] P. Senger for the KaoS Collaboration, *Strange Mesons as a Probe for Dense Nuclear Matter*, Prog. Part. Nucl. Phys. **42** (1999) 209-219 `arXiv:nucl-ex/9901008`

[Sess00] M. Sessler, *Algorithms on CPUs and FPGAs for the ATLAS LVL2 Trigger*, PhD thesis, Heidelberg University (2000)

[Sess98] M. Sessler, M. Smizanska, *Global Pattern Recognition in the TRT for the ATLAS LVL2 Trigger*, ATLAS DAQ Note (1998) `CERN/ATL-DAQ-98-120`

[Siem79] P.J. Siemens, J.O. Rasmussen, *Evidence for a Blast Wave from Compressed Nuclear Matter*, Phys. Rev. Lett. **42** (1979) 880-883

[Sing00] H. Singpiel, *Der ATLAS LVL2-Trigger mit FPGA-Prozessoren - Entwicklung, Aufbau und Funktionsnachweis des hybriden FPGA/CPU-basierten Prozessorsystems ATLANTIS*, PhD thesis, Heidelberg University (2000)

[Stoc02] M.R. Stockmeier, *Pion Production in Relativistic Heavy Ion Collisions*, PhD thesis, Heidelberg University (2002)

[Stöc82] H. Stöcker et al., *Jets of nuclear matter from high energy heavy ion collisions*, Phys. Rev. **C25** (1982) 1873–1876

[Stur01] C. Sturm and the KaoS Collaboration, *Evidence for a Soft Nuclear Equation-of-State from Kaon Production in Heavy Ion Collisions*, Phys. Rev. Lett. **86** (2001) 39-42 `arXiv:nucl-ex/0011001`

[Uehl33] E.A. Uehling, G.E. Uhlenbeck, *Transport Phenomena in Einstein-Bose and Fermi-Dirac Gases. I*, Phys. Rev. **43** (1933) 552–561

[Uehl34] E.A. Uehling, *Transport Phenomena in Einstein-Bose and Fermi-Dirac Gases. II*, Phys. Rev. **46** (1934) 917–929

[Volo96] S. Voloshin, Y. Zhang, *Flow study in relativistic nuclear collisions by Fourier expansion of azimuthal particle distributions*, Z. Phys. **C70** (1996) 665-671

[Wisn00] K.Wisniewski, *Kaon and Antikaon production in Heavy-Ion Collisions for the reaction Ru+Ru and Ru+Zr at 1.69 AGeV beam kinetic energy*, PhD thesis, Warsaw University (2000)

[Wisn00b] K. Wisniewski, P. Crochet, N. Herrmann and the FOPI Collaboration, *Direct comparison of phase-space distributions of K- and K+ mesons in heavy-ion collisions at SIS energies - evidence for in-medium modifications of kaons?*, Eur. Phys. J. **A9** (2000) 515-519 `arXiv:nucl-ex/0101009`

[Xili04] Xilinx, Inc., *Virtex-II Platform FPGA User Guide* (2004) `http://direct.xilinx.com/bvdocs/userguides/ug002.pdf`

[Xili97] Xilinx, Inc., *An Introduction to Xilinx Products* (1997) `http://www.xilinx.com/`

[Yepe96] P. Yepes, *A fast track pattern recognition*, Nucl. Instrum. Meth. **A380** (1996) 582-585

Appendix A: Application User Documentation

Description of all elements of Hough Online's user interface - dialog pages, input and select boxes, viewer windows, and necessary files containing configuration settings or event and detector data. In analogy to the user interface this description is divided into the following sections - representing the tabs in the main dialog:

1 The dialog tab "Simulation"

The first dialog tab - which is active at program startup - holds the settings for the operation mode (simulation, optimization), for the storage location of results and optimization range, and for the storage location of all program settings.

1.1 Simulation properties (dialog tab "Simulation")

Elementary settings regarding Hough Online's operation modes - which type of events to choose and for what purpose, and how many of them - can be customized here.

1.1.1 Parameter scheme

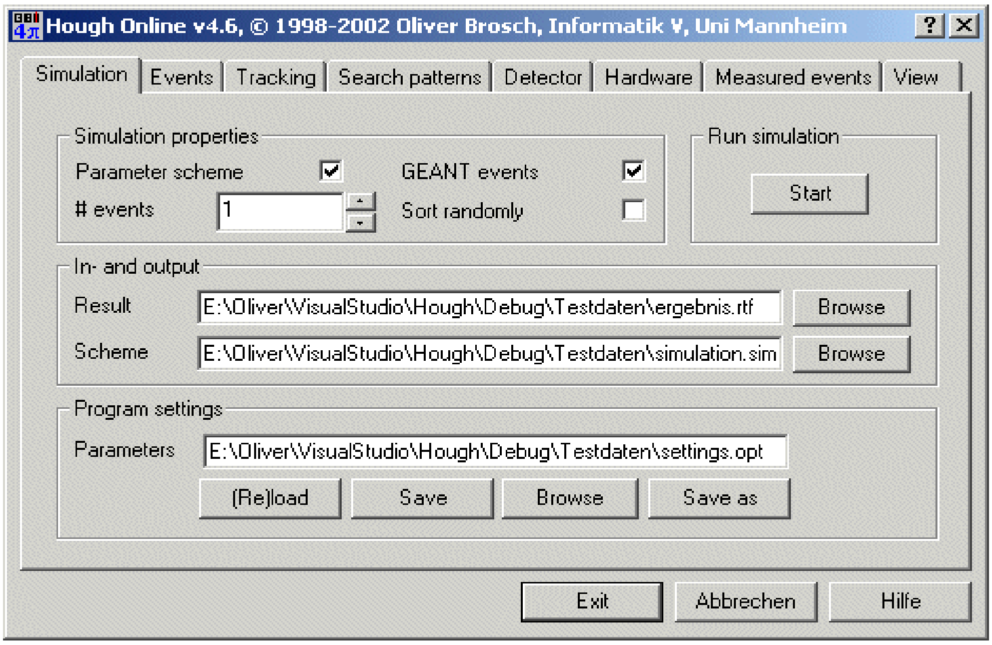
Checking this box enables the use of a parameter scheme. This means that multiple sets of parameters are used, one after another, for a simulation run with the given number of events each. The parameters in the special configuration file are used instead of the parameters shown or adjusted in the dialog boxes. This setting can be used for a search for the best parameter set for a given type of event, e.g. a search for the most appropriate histogram threshold.

1.1.2 GEANT events

If this box is checked, externally created (GEANT) events are used for simulation, rather than events created by Hough Online. GEANT events should be more realistic, but simulation consumes more time, because the scanning of the event files is slow. Moreover, their number is limited in most cases. Please read more about the internal event creation to decide whether to use them instead or not. When using GEANT events, their sorting can become important.

1.1.3 # events

This is the number of events that are used for a simulation run with a given parameter set, or for each parameter set in a parameter scheme. Every single event will be unique, because even



Hough Online v4.6, © 1998-2002 Oliver Brosch, Informatik V, Uni Mannheim

Simulation properties

Parameter scheme ☑

events 1

GEANT events ☑

Sort randomly □

In- and output

Result E:\Oliver\VisualStudio\Hough\Debug\Testdaten\ergebnis.rtf

Scheme E:\Oliver\VisualStudio\Hough\Debug\Testdaten\simulation.sim

Program settings

Parameters E:\Oliver\VisualStudio\Hough\Debug\Testdaten\settings.opt

when using a smaller number of GEANT events, a statistic uncertainty that accounts for the limited resolution of the time-of-flight detector is added to the raw data during event analysis.

1.1.4 Sort randomly

GEANT events by default are sorted by their path and file names. However, if the number of events to use does not match the number of events available, this might lead to statistical errors. For example, if events with kaons and with anti-kaons are stored in different subdirectories, no anti-kaons are used at all, if the number of events does not exceed the number of available kaon events. If this box is checked, events are used in random order to avoid this problem. On the other hand, different events are used for each run, and for each parameter set.

1.2 Run simulation (dialog tab "Simulation")

The button contained here serves for actually starting a simulation by processing one event after another.

1.2.1 Start

A click on this button starts the a simulation run, applying the computation of the Hough algorithm to the specified events and parameters. This opens a new dialog window to display the computation's progress. If a simulation is done for the first time, or if settings for the predefined search patterns or the data rate of the central drift chamber have been modified since the last run, a new lookup table is calculated as the first step. If GEANT events are used for the simulation, and a new set of events is used, or a simulation is done for the first time, the first step is to build a database containing the paths and file names of all event data files.

1.3 In- and output (dialog tab "Simulation")

The storage locations for the simulation result and for the set of parameters to be used for it are specified here.

1.3.1 Result

The storage location for simulation results is specified in this field. The file contains the settings for most of the parameters that were used, and the efficiency results for single or all events, depending on the mode of operation that is used. If a parameter scheme is used, the file starts with a header showing the parameters common for all parameter sets. The header is followed by a table that lists for each set the remaining parameters and values for the kaon and algorithm efficiencies, separately for different ranges of tracks per event. If no parameter scheme is used, the table lists a detailed analysis for each event. In a second table the complete result of kaon and algorithm efficiencies is listed. The individual header and table entries have the following names and meanings and appear in the following operation modes:

- **single**: single parameter set, Hough Online or externally created events
- **multi**: multiple parameter set, Hough Online or externally created events

Keyword	Mode		Description
	single	multi	
Parameters			
no.	X	X	Consecutive number of current event or parameter set
Events	X	X	Total number of events, or events per parameter set
GEANT	X	X	GEANT events are used for simulation • 0: events created by Hough Online • 1: externally created events
Random	X	X	GEANT events are sorted randomly • 0: sorted by pathname / filename • 1: sorted randomly
Discriminator	X	X	Value of the discriminator threshold
Original	X	X	Original histogram is used for maximum search • 0: new histogram, null filtered • 1: original histogram, no null filter
Matching mm, match	X	X	Tolerance of the CDC-TOF matching in mm
Acceptance MeV, accept	X	X	Value of the kaon acceptance in MeV / c^2
pMax, vMax	X	X	Largest allowed particle momentum p_{max} in MeV / c or velocity v_{max} in units of c
Threshold, tresh	X	X	Value of the histogram threshold

Keyword	Mode		Description
	single	multi	
Filter, filt	X	X	Type of used null filter • 0: Average (5 x 5) • 1: Average (7 x 7) • 2: Minimum (3 x 3) • 3: Minimum (5 x 5) • 4: 4 fields (7 x 7) • 5: 3-diagonal (7 x 7) • 6: No filter
H Filter, hfilt	X	X	Type of used max filter • 0: H filter (5-3-5) • 1: Square (3 x 3) • 2: Square (5 x 5) • 3: 2-diagonal (5 x 5) • 4: 3-diagonal (7 x 7) • 5: 3-rhombus (5 x 5) • 6: 5-rhombus (7 x 7) • 7: X filter (7-5-3-5-7)
Pattern phi, rc	X	--	Number of search patterns in phi or p_t direction
Tracks	X	--	Total number of search patterns
DeltaPhi mm, delta	X	X	phi resolution in mm. When using multiple parameter sets the p_t resolution is calculated from the number of p_t patterns that would result from a p_t distribution of equal TOF spacing and a p_t resolution equal to the phi resolution.
DeltaPt mm	X	--	Value of the p_t resolution in mm
Spacing, space	X	X	Used p_t distribution • 0: Equal TOF spacing • 1: Equal track spacing • 2: Proportional 1 / r
RAvg mm, ravg	X	X	Value of the p_t crossing distance in mm
Track Width	X	X	Setting for the track width • 0: Calculate • 1: Fixed +/- 1 • 2: Zero width +/- 0
Wires, wires	X	X	CDC wires per sector used for data input
Input Wires	X	--	The total number of used CDC wires, which is 16 times the number of wires per sector.
Frequency MHz, freq	X	X	Sampling frequency for the CDC data in MHz
Samples	X	--	Sampling points per used CDC wires. This is equal to 5 times the sampling frequency in MHz.

Keyword	Mode		Description
	single	multi	
Straws	X	--	Number of different possible CDC hits. It is equal to 80 times the number of wires per sector times the sampling frequency in MHz.
TOF Resolution ps, cm	X	X	The time resolution in ps or the spatial resolution in z direction in cm of the time-of-flight detector
FPGA Frequency MHz	X	X	The design frequency the used FPGA is capable of when using a complex design
DualPort	X	X	The FPGA internal RAM blocks can be configured as dual-ported • 0: no dual-port configuration possible • 1: dual-port configuration is possible
Results			
evnt	X	--	Number of events found within the given track multiplicity range
trck	X	X	Number of tracks in the current event, or track multiplicity range
good	X	--	Number of correctly recognized particles in the current event
fake	X	--	Number of erroneously recognized particles in the current event
kaon	X	--	The kaon status of the current event. The single bits of this value mean, if set • bit 0 (LSB): event contains a K+ • bit 1: event contains a K- • bit 2: a K+ was found in the event • bit 3: a K- was found in the event • bit 4: the found kaon matches the original kaon Example: a kaon status of 5 means that a K+ was found in an event that contained a K+, but these two kaons are not identical.
err/mm	X	--	Average deviation of recognized particles from their original counterparts at the location of the time-of-flight detector in mm
err/phi	X	--	Average deviation of recognized particles from their original counterparts at the location of the time-of-flight detector in mm, if the phi values of the recognized and the p_t values of the original tracks are used
err/pt	X	--	Average deviation of recognized particles from their original counterparts at the location of the time-of-flight detector in mm, if the p_t values of the recognized and the phi values of the original tracks are used
nrg	X	--	Kaon momentum range in GeV / c
K+, K-	X	X	Number of K+ or K- events contained in the given track multiplicity and/or kaon momentum range

Keyword	Mode		Description
	single	multi	
effK+, effK-	X	X	Percentage of K+ or K- found in the given track multiplicity and/or kaon momentum range
fakeK+, fakeK-	X	X	Number of fake K+ or K- erroneously found in events within the given track multiplicity range
enhK+, enhK-	X	X	Kaon enhancement factor for an experiment with events within the given track multiplicity range

1.3.2 Scheme

If multiple sets of parameters are used for a simulation, the file containing the values to be varied must be specified here. Clicking the "Browse" button opens a new dialog box for selection of a file with the mouse. The file must meet the following format convention:

- Each line starts with a keyword representing a variable, followed by the number of different values to assign for this variable, and finally the values for this variable.
- All possible keywords must be included in the file in the order specified below. If a variable shall not be varied, its multiplicity must be 1 followed by its unique value.
- The numeric values for some histogram settings can be taken from the table in the section above.

The keywords have the following form, meaning, and order of appearance:

Keyword	Values	Description
Delta	floating point	phi resolution in mm. The p_t resolution is calculated from the number of p_t patterns that would result from a p_t distribution of equal TOF spacing and a p_t resolution equal to the phi resolution.
Matching	floating point	Tolerance of the CDC-TOF matching in mm
Frequency	integer	Sampling frequency for the CDC data in MHz
Wires	integer	CDC wires per sector used for data input. Possible values are • 1: 60 wires per sector (960 wires total) • 2: 30 wires per sector (480 wires total) • 3: 20 wires per sector (320 wires total) • 4: 15 wires per sector (240 wires total) • 5: 12 wires per sector (192 wires total) • 6: 10 wires per sector (160 wires total)
Threshold	integer	Value of the histogram threshold
Filter	integer	Type of used null filter • 0: Average (5 x 5) • 1: Average (7 x 7) • 2: Minimum (3 x 3) • 3: Minimum (5 x 5) • 4: 4 fields (7 x 7) • 5: 3-diagonal (7 x 7) • 6: No filter

Keyword	Values	Description
Tracks	integer	Average number of tracks per event. This value is only used for events created by Hough Online.
Accep-tance	floating point	Value of the kaon acceptance in MeV / c^2
H-Filter	integer	Type of used max filter • 0: H filter (5-3-5) • 1: Square (3 x 3) • 2: Square (5 x 5) • 3: 2-diagonal (5 x 5) • 4: 3-diagonal (7 x 7) • 5: 3-rhombus (5 x 5) • 6: 5-rhombus (7 x 7) • 7: X filter (7-5-3-5-7)
RAvg	floating point	Value of the p_t crossing distance in mm
Spacing	integer	Used p_t distribution • 0: Equal TOF spacing • 1: Equal track spacing • 2: Proportional 1 / r

A multi parameters simulation file may look as follows (the number of different variable values is highlighted):

```
Delta 1 10
Matching 1 20
Frequency 1 25
Wires 3 2 3 4
Threshold 5 32 36 40 44 48
Filter 1 2
Tracks 1 50
Acceptance 3 80 100 120
H-Filter 1 2
RAvg 3 200 400 600
Spacing 3 0 1 2
```

A simulation with this configuration file will use 405 different parameter sets, each with the same number of events, so the total number of events can become huge!

1.4 Program settings (dialog tab "Simulation")

The parameters adjusted in (almost) every input control are stored in a file, which is specified and may be reloaded or overwritten here.

1.4.1 Parameters

All program, algorithm, and detector parameters are stored in a file, or can be saved to a file, that is specified here. The four buttons have the following functions:

- **(Re)load**: all settings are overwritten (or restored) by the contents of the specified file.

- **Save**: all settings are stored into the specified file, which is overwritten, if it already exists.
- **Browse**: a new dialog box is opened to select a file with the mouse. All settings are then overwritten by the contents of this file.
- **Save as**: a new dialog box is opened to select a folder and a file with the mouse, or to enter the name of a new file, which will be created. The contents of this file is then overwritten by the current program settings.

The specified file must have, and will have after it is written by Hough Online, the following format:

- The file is grouped into sections. Each section begins with the section name in [brackets] and ends with an empty line.
- Each line in a section starts with a keyword representing a variable, followed by the value for this variable.
- All possible keywords must be included in the file in the order specified below.
- For some variables only a few values are allowed. The possible values and their meaning are shown below.

The keywords have the following form, meaning, and order of appearance:

Keyword	Value	Description
[Histogram]		
PtLut	• 0: Equal TOF spacing • 1: Equal track spacing • 2: Proportional 1 / r	pt distribution
numPhi	integer	# phi patterns
numPt	integer	# pt patterns
deltaPhi	floating point	phi resolution
deltaPt	floating point	pt resolution
overMin	floating point	Inner overlap
overMax	floating point	Outer overlap
TrackWidth	• 0: Calculate • 1: Fixed +/- 1 • 2: Zero width +/- 0	Track Width
ptMin	floating point	Track smallest r
ptMax	floating point	Track biggest r
RAvg	floating point	pt cross. distance
[Detector]		
Frequency	integer	Sampling frequency
Wires	• 1: 60 • 2: 30 • 3: 20 • 4: 15 • 5: 12 • 6: 10	Wires / sector

Keyword	Value	Description
Time	floating point	Time-of-flight
zAxis	floating point	z direction
RPestof	floating point	TOF radius
MagnetField	floating point	Magnetic field
thetaCDCmin	floating point	theta CDC from
thetaCDCmax	floating point	theta CDC to
thetaTOFmin	floating point	theta TOF from
thetaTOFmax	floating point	theta TOF to
[Algorithm]		
Discriminator	integer	Threshold
DeadTime	integer	Dead Time
Eliminator	integer	Eliminator
Threshold	integer	Threshold
Filter	• 0: Average (5 x 5) • 1: Average (7 x 7) • 2: Minimum (3 x 3) • 3: Minimum (5 x 5) • 4: 4 fields (7 x 7) • 5: 3-diagonal (7 x 7) • 6: No filter	Null filter
Original	• 0: unchecked • 1: checked	Max search
HFilter	• 0: H filter (5-3-5) • 1: Square (3 x 3) • 2: Square (5 x 5) • 3: 2-diagonal (5 x 5) • 4: 3-diagonal (7 x 7) • 5: 3-rhombus (5 x 5) • 6: 5-rhombus (7 x 7) • 7: X filter (7-5-3-5-7)	Max filter
Gap	integer	Allowed gap
TrackFit	integer	Fit tracks
Matching	floating point	CDC-TOF
Acceptance	floating point	Kaons
pMax	floating point	p (max)
vMax	floating point	v (max)
Invert	• 0: unchecked • 1: checked	Enable
[Simulation]		
Events	integer	# events
Geant	• 0: unchecked • 1: checked	GEANT events

Keyword	Value	Description
Random	- 0: unchecked - 1: checked	Sort randomly
Scheme	- 0: unchecked - 1: checked	Parameter scheme
[Events]		
Tracks	integer	Tracks / event
Noise	integer	Noise
vDrift	floating point	Drift velocity
Lorentz	floating point	Lorentz angle
PiPercent	integer	Fraction pi +/-
PPercent	integer	Fraction p
KPlusMinus	- 0: K+ and K- - 1: only K+ - 2: only K-	K+/-
PiMoment	floating point	Avg. momentum pi+/-
PMoment	floating point	Avg. momentum p
KMoment	floating point	Avg. momentum K+/-
[Analysis]		
DesignFreq	integer	Design frequency
RAMBlocks	integer	Block RAM / board
RAMBytes	integer	Block RAM 8/9 bit x
RAMDualPort	- 0: unchecked - 1: checked	Block RAM dual-pot
LUTRAM	integer	LUT bits w/ RAM module
IOBits	integer	I/O bits w/o B2B/module
Scanner	floating point	Scanner time
DSP	floating point	Au+Au: DSP time
eKplus	integer	Ni+Ni: 1 K+ per
eKminus	integer	Ni+Ni: 1 K- per
BoardType	- 0: User defined - 1: Atlantis AIB (2 x XCV600) - 2: microEnable II (XCV1000) - 3: RACE-1 (XC2V3000)	Board type
[View]		

Keyword	Value	Description
Zoom	• 1: 25% • 2: 50% • 3: 75% • 4: 100% • 5: 125% • 6: 150% • 8: 200% • 10: 250% • 14: 350% • 20: 500% • 28: 700% • 40: 1000%	Zoom
Wires	integer	Wires
Hits	integer	CDC hits
Tracks	integer	Tracks
Numbers	integer	TOF hits
Highlight	integer	Marked
[Files]		
Geometry	string	Geometry
Settings	string	Parameters
Simulation	string	Scheme
Result	string	Result
CDCFolder	string	CDC / PID
CDCData	string	CDC data
TOFData	string	TOF data
OfflineData	string	Offline data

2 The dialog tab "Events"

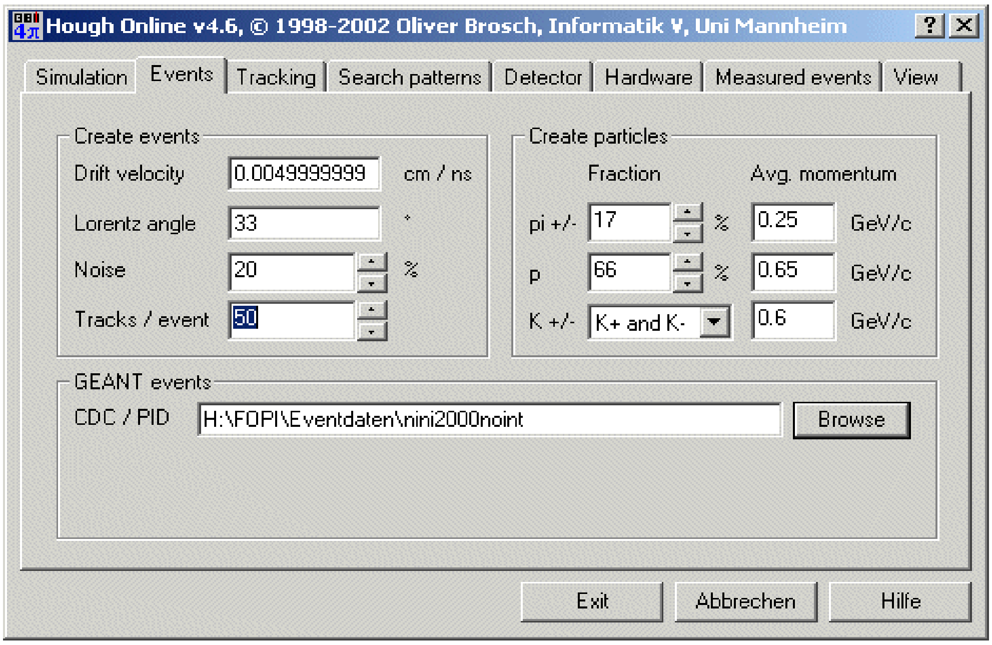
Depending on the selected operation mode this dialog tab enables the selection of the storage location of externally created events or settings for stand-alone creation of events.

2.1 Create events (dialog tab "Events")

If events are created by Hough Online independently, some global parameters - like the average number of tracks per event - must be specified in this group.

2.1.1 Drift velocity [cm / ns]

Here the velocity is entered, with which the charge resulting from the particle tracks avalanches to the signal wires. The velocity is calculated from values that can be measured inside the detector (magnetic field, gas pressure, drift field) and is usually slightly below 0.005 cm/ns. In



Create events

Drift velocity 0.0049999999 cm / ns

Lorentz angle 33 °

Noise 20 %

Tracks / event 50

Create particles

	Fraction		Avg. momentum	
pi +/-	17	%	0.25	GeV/c
p	66	%	0.65	GeV/c
K +/-	K+ and K-		0.6	GeV/c

GEANT events

CDC / PID H:\FOPI\Eventdaten\nini2000noint

principal, this velocity can be chosen arbitrarily for events created by Hough Online. The only thing to make sure is that the signal wires (the maximum radius of the drift chamber is 80 cm, it consists of 16 sectors, the charges drift to the signal wires under a certain angle towards the mid of a sector) can be reached within the maximum drift time, which is hard-coded to be 5 µs.

2.1.2 Lorentz angle [°]

The Lorentz angle is the deviation of the path of the charge drift to a line horizontal to the signal wires' plane of the CDC. It is calculated from values that can be measured inside the detector (magnetic field, gas pressure, drift field) and is usually around 33°.

2.1.3 Noise [%]

Signals coming from real events are naturally more or less superposed by noise. By specifying a signal-to-noise ratio in percent this can be simulated for events created by Hough Online in a rather primitive way. The number should be around 20% or more.

2.1.4 Tracks / event

Here the type of the events to be created is defined. Events of the type Ni+Ni @ 2 AGeV display an average of 25 tracks, events of the type Au+Au @ 1.5 AGeV have 60 tracks at average. The type of events influences, among others, the trigger and DSP time (the latter does not have to be modified accordingly, because it is always related to Au+Au @ 1.5 AGeV events).

2.2 Create particles (dialog tab "Events")

Settings for particles contained in the events created by Hough Online - percentual share of particle species and average momenta.

2.2.1 Fraction pi+/- [%]

If no pre-simulated or measured events are available, or if the simulation of many events should be accelerated, Hough Online can create events on its own. This is not a physical exact simulation though, but a composition of reasonable particle tracks, so that the input data look like they would if they came from real events. The trigger does not care about physics either, but simply tries to recognize and classify the existing tracks. The necessary information about the percentual fraction of total tracks which is equal for positively and negatively charged pions should be in the order of 17%.

2.2.2 Fraction p [%]

In analogy to the information about the percentual fraction of pions the fraction of protons should be in the order of 66%.

2.2.3 K+/-

Self-created as well as pre-simulated events contain exactly one single kaon - either positively or negatively charged. The creation of kaons can be restricted to positively or negatively charged particles alone:

a) K+ and K-

Following a random distribution the created events will contain about equal as many positively and negatively charged kaons.

b) only K+

Each event will contain exactly one kaon, but no anti-kaon.

c) only K-

Each event will contain exactly one anti-kaon, but no kaon.

2.2.4 Avg. momentum pi+/- [GeV/c]

This input control asks for the average momentum, that created pions should have. The distribution of the momenta then follows a standard distribution, with a standard deviation of the same value. Particles with momenta below 0.1 GeV/c are discarded. The momenta's distribution is not physically exact, but rather designed to make the particle spectra produced after them look like real ones. A reasonable average value for the pions' momenta is 0.25 GeV/c.

2.2.5 Avg. momentum p [GeV/c]

In analogy to the information about the average momentum of pions a reasonable average value for the protons' average momenta is 0,65 GeV/c.

2.2.6 Avg. momentum K+/- [GeV/c]

In analogy to the information about the average momentum of pions a reasonable average value for the kaons' average momenta is 0,6 GeV/c.

2.3 GEANT events (dialog tab "Events")

If externally created events are used for the simulation, their storage location must be specified here.

2.3.1 CDC / PID

The storage location for files containing pre-simulated events is specified here. Clicking the "Browse" button opens a dialog box for selection of a folder using the mouse. Subfolders of the selected folder are also browsed for event data. The files containing the events must meet the following requirements:

a) File names

A single event always consists of a file with CDC raw data and a file with particle information. Both files must be located within the same folder and must meet the following naming convention:

- CDC raw data: **cdcxxxxyyyyy0000.raw**, where xxxx can represent a 4-digit run number and yyyyy a 5-digit event number. Example: cdc0003016320000.raw.
- PID particle information: **pidxxxxyyyyy.dat**, where xxxx can represent a 4-digit run number and yyyyy a 5-digit event number. Example: pid000301632.dat.
- **xxxxyyyyy** must be identical for both files and sum up to exactly 9 characters in length.

b) CDC data

A file with CDC raw data meets the following format convention: the first line contains one after another

- the underlying drift velocity in cm/ns,
- the used Lorentz angle in ° and three other numbers, that may represent
- a global drift time offset (all floating point values),
- a run and
- an event number (both integer values)

and are of no further importance. From the second line pulse snippets are listed, that are built as follows: at the beginning stand

- the sector number
- the wire number and
- the offset, from which the snippet starts, in units of 10 ns (all integer values).

An even number of digitized pulse height values follows, alternately for the signal read from the left and the right end of the wire, in FADC units. All these values are integer. They are equally spaced in time by 10 ns. At the end of each pulse the magic word "FF" closes the snippet. The beginning of such a file may e.g. look as follows (sector and wire number, offset, and end-of-pulse word are marked):

```
 0.00499999989  33.  0. 1 131144
1 5 114 9 9 10 10 11 11 13 14 14 16 14 16 14 15 13 14 12 13
11 12 FF
2 5 96 9 9 9 9 10 11 14 18 24 34 34 51 38 57 38 57 36 55 35
52 33 49 30 46 28 42 26 38 23 33 20 28 17 23 15 20 14 17 12
15 12 13 11 12 FF
```

```
2 6 92 11 9 10 9 12 10 13 16 24 34 41 62 53 79 55 82 55 81
52 78 50 74 46 69 43 65 40 60 36 54 32 47 27 39 23 33 20 27
17 23 15 19 13 16 12 14 11 13 11 12 FF
2 7 92 9 9 9 10 11 12 16 23 28 45 40 66 44 71 43 71 42 69 40
65 37 61 35 57 32 52 29 46 25 40 22 33 19 28 16 23 14 19 13
17 12 15 11 13 11 12 FF
2 8 87 6 9 8 9 10 10 13 14 22 27 33 46 39 57 41 59 39 58 37
55 35 52 33 49 31 45 29 42 26 38 23 33 20 28 18 24 15 20 14
17 13 15 12 13 11 12 FF
2 9 86 6 9 7 10 10 13 19 27 34 53 47 74 50 79 49 79 48 76 44
72 41 67 38 62 35 57 31 51 27 43 23 36 20 29 17 24 15 20 13
17 12 15 11 13 11 12 FF
```

c) PID data

The file containing particle information uses the following format: each line carries information about a single particle contained in the event. This information contains one after another:

- the ID[1] for the particle species
- the starting angle[2] phi_S of the particle's track in the CDC in °
- the polar angle[3] theta of the particle's track in °
- the radius of curvature[4] r_C of the track in cm
- the time-of-flight[5] t of the particle to the time-of-flight detector in ps and
- the location[5] phi_{TOF} where the particle hit the time-of-flight detector in °.

Comments: [1] Among others, particle IDs may be: 8: pions, 9: anti-pions, 11: kaons, 12: antikaons, 14: protons. [2] The starting angle is measured directly at the center-of-collision in the plane perpendicular to the beam direction. [3] The polar angle is the deviation from the flight direction of the projectile. [4] Tracks of positively charged particles are right-curved (positive radius), tracks of negatively charged particles are left-curved. [5] If a particle does not leave a hit on the time-of-flight detector, its time-of-flight is 0 ps and the corresponding angle 400°.

A PID file may e.g. look as follows:

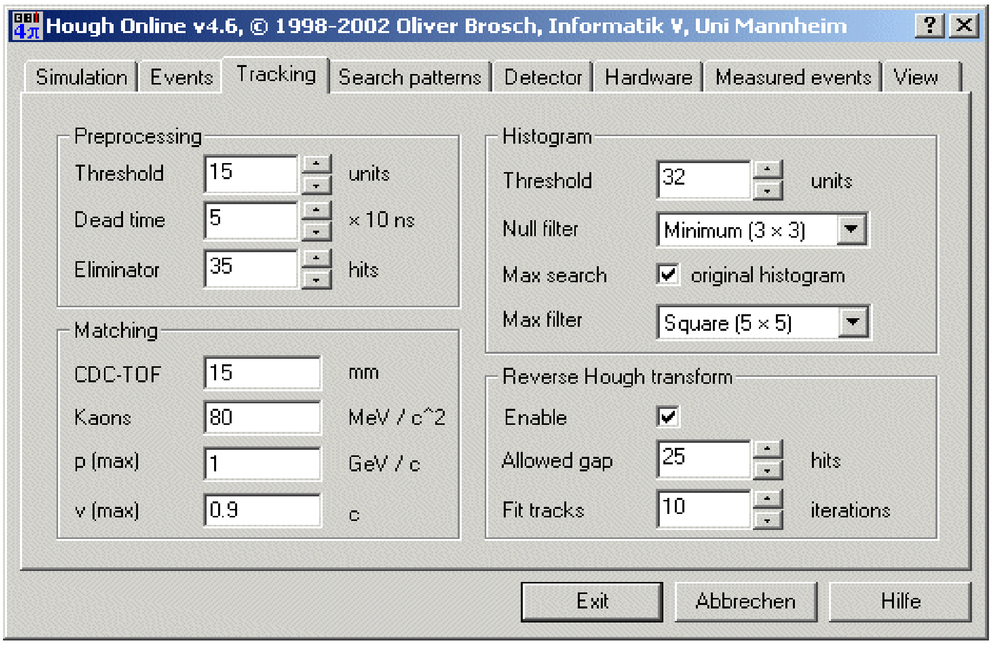
```
14     338.04     127.10      97.89        0.00     400.00
14     229.38      52.60      54.12    41344.09     161.88
14     222.74      48.14     303.79     7304.11     213.27
 9     122.24      34.41    -156.22        0.00     400.00
 8     178.99      37.24      64.41     7810.27     128.06
12     239.41      36.77    -974.19     5653.86     242.35
```

3 The dialog tab "Tracking"

Here the parameters for the track recognition algorithm - thresholds, filters, and additional program steps - should be set.

3.1 Preprocessing (dialog tab "Tracking")

The Preprocessing deals with the morphing of detector raw data into hit information.



Preprocessing

Threshold 15 units

Dead time 5 × 10 ns

Eliminator 35 hits

Matching

CDC-TOF 15 mm

Kaons 80 MeV / c^2

p (max) 1 GeV / c

v (max) 0.9 c

Histogram

Threshold 32 units

Null filter Minimum (3 × 3)

Max search ☑ original histogram

Max filter Square (5 × 5)

Reverse Hough transform

Enable ☑

Allowed gap 25 hits

Fit tracks 10 iterations

3.1.1 Threshold [units]

The algorithm function "threshold" emulates the discriminator output of the CDC subdetector's flash ADCs. The digitized (and possibly zero-suppressed) CDC signal is examined for rising edges. Every time the signal exceeds the threshold value defined here, it is replaced by a rectangle pulse, otherwise by a zero. After a certain dead time the function is ready for the creation of the next rectangle. Useful values for this threshold can - depending on the CDC signals' quality - be around 20 - 100. If the raw data originate from a measured event with CDC data in .atim format, this function and thus the threshold is not applied.

3.1.2 Dead time [x 10 ns]

The time that has to go by between two rectangle pulses created by the FADCs' discriminator outputs. It is typically 5 units, which means 50 ns. If the raw data originate from a measured event with CDC data in .atim format, a setting of the dead time has no meaning.

3.1.3 Eliminator [hits]

The FOPI detector's central drift chamber (CDC) may possibly produce artifacts that can be in the order of magnitude of real signals. These are often signals that run in parallel to the plane of sense wires, which means that they cause a pulse on many or even all wires within a sector at the very same time. Because of the tilted geometry of the signal wires' planes real tracks can never run in parallel to these planes. Therefore it is reasonable to assume, that parallel signals are artifacts, and to suppress their pulses accordingly. The number of hits adjusted here is the upper limit for real signals. If more signals occur at the same (or up to one time sample earlier or later) within the same sector and on different wires, they will be deleted. A useful value for this upper boundary is around a third up to half of the number of readout wires per sector. If the number is chosen to be equal or higher than the number of readout wires per sector, this function is disabled.

3.2 Histogram (dialog tab "Tracking")

Algorithm settings around the track histogram - threshold and filters.

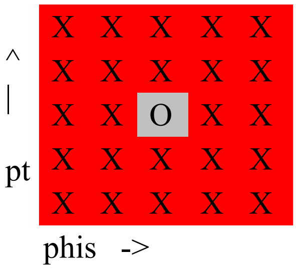
3.2.1 Threshold [units]

Prior to finding maxima within the track histogram and after applying the null filter, all entries below this threshold are cut off. This ensures that only maxima within peaks rather than all local maxima within the histogram will be found. The setting for this threshold is independent from the number of readout wires.

3.2.2 Null filter

The null filter adjusts the values in the track histogram to their neighbors in such a way that the later application of the threshold leaves relative peaks instead of global largest values. In other words: the null filter subtracts the histogram's background. The null filter offers the following settings:

a) Average (5x5), average (7x7) [avg5, avg7]

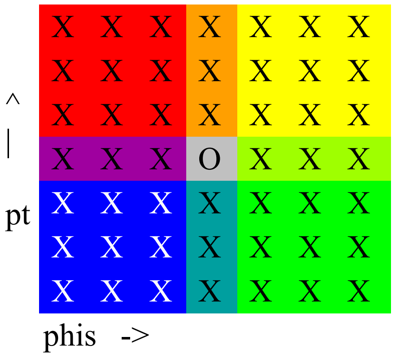


With this setting the average value of the 24 (5x5 - 1, see draft left) or 48 (7x7 - 1) neighboring values (red) will be subtracted from each histogram value (gray).

b) Minimum (3x3), minimum (5x5) [min3, min5]

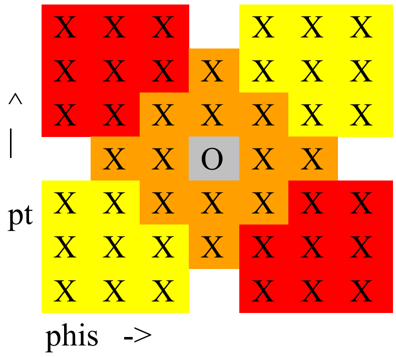
Similar to the first setting, however the smallest of the 8 (3x3 - 1) or 24 (5x5 - 1, as in the draft above left) neighboring values will be subtracted from each histogram value.

c) 4-fields (7x7) [4field]



With this setting the largest average value of 4 diagonal adjacent 4x4 squares (15 values each, see draft on the left: the 4 fields overlap in a few values each, indicated by intermediate colors) will be subtracted from each histogram value. This accommodates the fact that tracks that are close to each other in the CDC, often lead to diagonal adjacent peaks in the histogram. Background enhanced by directly adjacent tracks will thus be reduced extra efficiently.

d) 3-diagonal (7x7) [3diag]



This setting subtracts the largest average value of both diagonals (width of 3 values, 28 values each, see draft left: the diagonals overlap in some values, indicated by an intermediate color) from each histogram value. This accommodates the fact that in particular peaks originating from mirror tracks are diagonally blurred in the histogram.

e) No filter [no-flt]

The setting "no filter" disables the null filter.

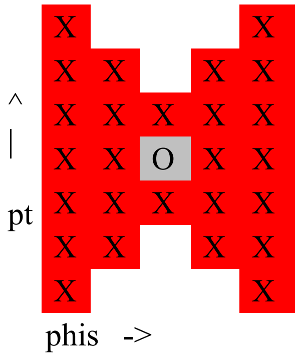
3.2.3 Max search - original histogram

If this box is checked, the null filter will only be used to decide which entries of the track histogram fall victim to the threshold. The search for local maxima will then be executed using the original values. In the other case the new values will be kept for the maxima search. The position of the maxima and thus the exact parameters of found track candidates may vary slightly with the change of this setting. However, when executing an additional fit with a sufficient number of iterations, both results will assimilate in most cases, because most of the underlying hits are identical within a single peak.

3.2.4 Max filter

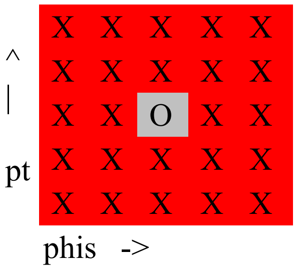
When searching for local maxima within the null-filtered and cut track histogram multiple track candidates, resulting from slight variations within the peak height of directly adjacent entries and originating mainly from exactly the same hits, have to be avoided. The max filter makes sure that within a local environment only the largest local maximum is considered. The geometry of the possible variants of the max filter is explained as follows:

a) H filter (5-3-5), X filter (7-5-3-5-7) [H, X]



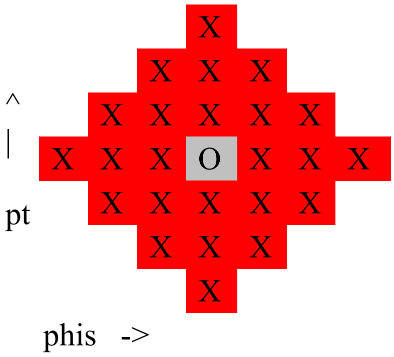
H and X filter (see draft left) assume that tracks that are close together in the CDC are preferably diagonal adjacent in the histogram. So local maxima that are below the current value (gray) are not considered, especially in diagonal direction. The H filter lacks the X filter's first and last column.

b) Square (3x3), square (5x5) [sq3, sq5]



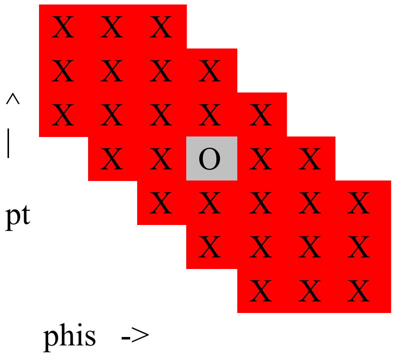
The square filter (see draft on the left: 5x5 version) equally handles all directions within the histogram. By the metric used (diagonal counts single) concurring local maxima in diagonal direction are slightly stronger suppressed.

c) 3-rhombus (5x5), 5-rhombus (7x7) [3rhom, 5rhom]



The rhombus filter (see draft left: 7x7 version) handles all directions in the histogram equally. By the metric used here (diagonal counts double) concurring local maxima in diagonal direction are slightly more weakly suppressed.

d) 2-diagonal (5x5), 3-diagonal (7x7) [2diag, 3diag]



The diagonal filter (see draft left: 7x7 version), in opposition to the rhombus filter, accommodates the fact that tracks, that with growing curvature have their starting angle shifted in the opposite direction, represent hits in the CDC that are closer together than those of tracks with starting angles shifted in the same direction.

3.3 Matching (dialog tab "Tracking")

Assignment of track and time-of-flight information, kaon windows in mass, momentum and velocity.

3.3.1 CDC-TOF [mm]

After extraction of track candidates from the track histogram and eventual application of additional algorithm functions all track candidates are compared with all time-of-flight data sets. For this procedure the track candidate is extrapolated to the location of the time-of-flight detector. Track candidates, that don't meet the time-of-flight detector within the distance from a time-of-flight data set specified here, are dropped. To the others the nearest time-of-flight information is assigned. This is how track candidates become kaon candidates.

3.3.2 Kaons [MeV / c^2]

The properties of the kaon candidates that remained after matching are used to compute the particle properties momentum, mass, and velocity. If the mass of a particle candidate meets the window around the kaon mass of 493,677 MeV/c^2 specified here, that particle is identified as a kaon. Consideration of the momentum's sign distinguishes between positively charged kaons (K+) and negatively charged anti-kaons (K-).

3.3.3 p (max) [GeV / c]

With growing particle momenta the particle branches in the p/v diagram move closer together and the detector resolution becomes worse. Thus the chance to safely distinguish between particle species decreases. By setting an upper boundary the number of fake kaons found thereby can be limited. Useful values for this boundary are in the order of 1 - 1.5 GeV/c. Above all, the boundary limits the interfering influence of possible protons, therefore the boundary is only active for positive particle momenta. A value of 0 disables this upper boundary.

3.3.4 v (max) [c]

The setting of an upper boundary for the particle velocity rejects all possible pions right from the scratch. Useful values for this boundary are in the order of 0.9 c. A value of 0 disables this upper boundary.

3.4 Reverse Hough transform (dialog tab "Tracking")

Activation and configuration of additional algorithm steps that require a significantly enhanced computing effort.

3.4.1 Enable

This check box enables the creation of an additional lookup table for reverse transformation of found track candidates into the CDC hits space. This reverse transformation is necessary for performing the algorithm functions explained below. The reverse lookup table requires a significantly larger memory for storage compared to the standard lookup table.

3.4.2 Use max filter

This check box determines whether the neighboring tracks - as determined by the setting for the maximum filter - are included in the track fit. This is useful in order to account for hits, that contributed to the same peak in the histogram, but were discarded by the maximum filter.

3.4.3 Allowed gap [hits]

Found track candidates might consist of hits that only accidentally meet a predefined track. In this case we assume that the found track is not closed, but rather displays a gap spreading over several neighboring CDC sense wires. Tracks that carry a gap that exceeds the threshold adjusted here will be removed from the list of track candidates. The number of maximum possible hits per track depends on the number of wires to be read. A threshold that is comparable to or larger than the number of read out wires will thus disable this function.

3.4.4 Fit tracks [iterations]

With the help of the reverse lookup table the found track candidates will be fitted to the hits that contributed to them in the histogram. This procedure is repeated iteratively, alternately computing only a new starting angle or a new radius of curvature. The underlying method converges quite reliably, but also slowly. Reasonable values for the number of iterations lie about between 10 and 100. A number of 0 deactivates the fitting function.

4 The dialog tab "Search patterns"

The Hough algorithm is based on a comparison with previously calculated tracks. These are defined here.

4.1 Number of search patterns (dialog tab "Search patterns")

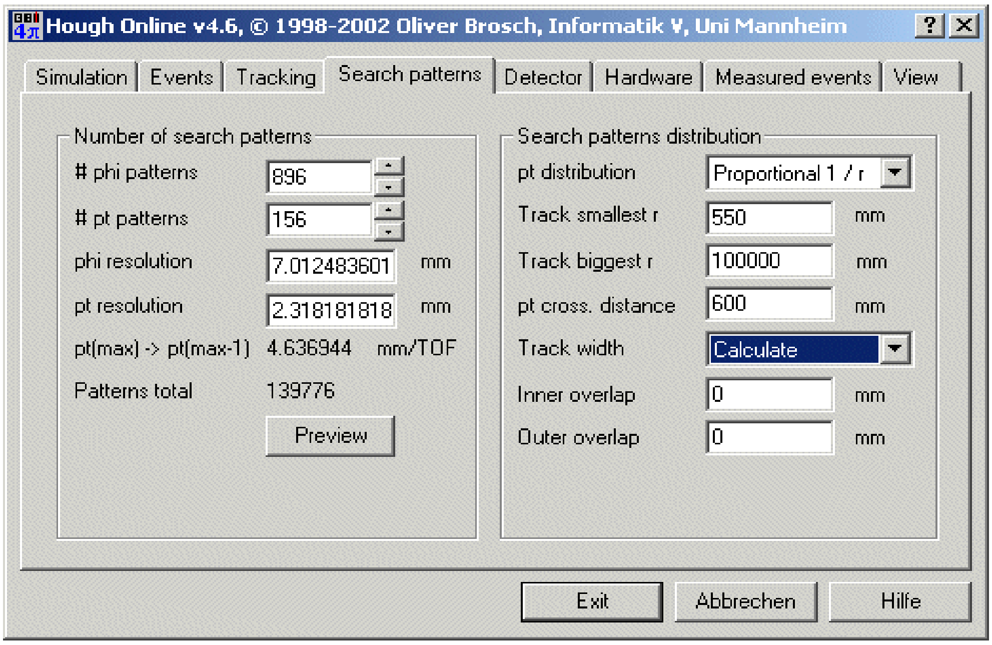
The number of predefined tracks determines the resolution of the particle classification, and is also responsible for computing time and memory consumption.

4.1.1 # phi patterns

This is the number of different angles phi (in the x-y plane perpendicular to the beam direction), under which a track that has to be found can have started from the center-of-collision. The angles are equally spaced over the whole range of 360°. In order to be able to copy to the lookup table the point symmetry of the track detector which consists of 16 sectors, this number must be divisible by 16 and will thus, if necessary, be corrected, when another input control is selected. Straight from the number of phi values results the resolution at the TOF detector, which is immediately re-calculated when another input control is selected. The number of phi values also determines the bit width of the lookup table, the necessary size of the RAM blocks and the computing time of the maximum search. Reasonable values for this number are in the order of 400 to 2,000, many simulations were done using a value of 896.

4.1.2 # pt patterns

Here the number of possible radii (p_t = transversal momentum, which is directly correlated to the track's radius) of the tracks to be found is specified. Because exactly straight tracks can not occur and would cause numerous problems in the calculations, the radii are divided half and half into right (for positively charged particles) and left (for negatively charged particles) curved tracks. Thus, their total number must be an even number and will be corrected when selecting another input control, if necessary. Straight from the number of p_t values results the resolution at the TOF detector, which is immediately re-calculated when another input control is focused. The number of p_t values also determines the bit width of the lookup table and the required number of RAM blocks. Reasonable values for this number are - depending on the distribution of the p_t values - in the range of 50 to 330. Many simulations were done using a number of 156.



Number of search patterns

# phi patterns	896	
# pt patterns	156	
phi resolution	7.012483601	mm
pt resolution	2.318181818	mm
pt(max) -> pt(max-1)	4.636944	mm/TOF
Patterns total	139776	

Search patterns distribution

pt distribution	Proportional 1 / r	
Track smallest r	550	mm
Track biggest r	100000	mm
pt cross. distance	600	mm
Track width	Calculate	
Inner overlap	0	mm
Outer overlap	0	mm

4.1.3 phi resolution [mm]

The resolution in phi is the distance between the spots where two tracks with same radius and directly adjacent starting angles hit the time-of-flight detector. As the number of possible starting angles calculated from this can assume only certain values, the resolution is automatically corrected, when another input control is selected. According to the reasonable values mentioned for the number, reasonable values for the resolution are between 3.2 and 15 mm. The resolution should be better than the tolerance for the assignment of time-of-flight data.

4.1.4 pt resolution [mm]

The resolution in p_t is the distance between two tracks with same starting angle and directly adjacent values for their radii at a certain location determined by the type of distribution of the tracks. As the number of possible track radii can only assume certain values, the resolution is automatically corrected, when another input control is selected.

4.1.5 pt(max) -> pt(max-1) [mm/TOF]

This is the distance between the spots, where the two tracks with same starting angle and the largest and second largest radii (assuming the same direction of curvature) hit the time-of-flight detector. This number is also a measure for the granularity of the selected track distribution with respect to the radius. It is always newly calculated from the other values on this page and should be smaller than the tolerance for the assignment of the time-of-flight data. This value is especially important for the momentum resolution of the Hough algorithm for particles at high momenta and thus the ability of identifying kaons at high energies.

4.1.6 Patterns total

The total number of patterns to be searched results from the number of different phi and p_t values and is always newly calculated from these values. It also represents the number of histogram counters and is thus a measure for the memory requirements of the histogram.

4.1.7 Preview

A click on the button "Preview" opens a new window, which schematically shows all predefined tracks. If the window has been opened before, its contents are refreshed. The window shows for one starting angle all tracks with different radii and for one radius within a limited region all tracks with different starting angles. Additionally, for a few neighboring wires in a sector of the drift chamber all - determined by the sampling frequency - possible hits are shown. This demonstrates their granularity and allows a comparison with the resolution of search patterns.

4.2 Search patterns distribution (dialog tab "Search patterns")

Beside the number also the distribution of the tracks to be found is of importance. For the same resolution e.g. the memory consumption can be minimized.

4.2.1 pt distribution

Not only the number of predefined patterns, but also their distribution with respect to track curvature are of importance for the resolution of the trigger algorithm. 3 different approaches can be focused here.

a) Equal TOF spacing [TOF]

This setting makes mutually adjacent tracks, that share the same starting angle, hit the time-of-flight detector at equal intervals. Tracks, that in contrast share the same radius and direction of curvature, hit the TOF detector in equal intervals anyway. This setting focuses the later matching of track and time-of-flight information: Each peak in the histogram is equal, because independent from the curvature an equal number of neighboring tracks are in a vicinity of a hit on the TOF detector. This distribution's disadvantage is the preference of strongly curved tracks - according to particles with low momenta. This possibly affects the resolution at high particles' momenta.

b) Equal track spacing [CDC]

This setting for the track distribution is based on the half distance between the center-of-collision - that is the origin of all tracks that should be found - and the point where the starting angle of a track is measured. If for the latter a distance not too close to the origin is chosen, for tracks with same starting angles an extra equal coverage of the detector's cross section is achieved. This means that tracks, independent of their curvature, have the same width and cause equally massive peaks in the histogram. Moreover, this distribution offers a good compromise between equally accounting for large and small particle momenta with respect to momentum resolution, and an equal handling with respect to the matching with time-of-flight data.

c) Proportional 1/r [p_t]

This setting distributes all tracks equally with respect to their reciprocal radius and thus equally accounts for all particle momenta. The algorithm's momentum resolution is thus independent of the particle's momentum. This distribution also allows saving RAM blocks in the FPGA, because for tracks with low momenta a lower resolution in space is needed. It is important though, that tracks can still be matched to time-of-flight data and the spacing does not become to large for tracks with low momenta.

4.2.2 Track smallest r [mm]

Here the smallest radius is specified, that a track may have, to make its recognition possible. It must be larger than half of the radius of the time-of-flight detector, because all tracks must be able to reach it. A good value for this radius is e.g. 550 mm. If tracks with relatively low momenta should not be recognized anyway, block RAM resources can be saved by increasing this radius.

4.2.3 Track biggest r [mm]

The largest radius, that a track may have, to make its recognition still possible, should be chosen as large as possible, to enable the identification of particles with high momenta. Example: 100,000 mm. On the other hand a very large maximum radius can lead to problems when calculating the other parameters of the lookup table. If an upper threshold for allowed momenta is set, this value can be adjusted accordingly, in order to save resources.

4.2.4 pt cross. distance [mm]

The p_t crossing distance is the distance from the center-of-collision, where the starting angle of all to-be-found tracks is defined. All tracks with the same starting angle intersect in a single point exactly at this distance from the origin. By variation of this value tracks, that have the same starting angle and lie close together in the histogram, can be tuned to also consist of hits, that lie close together in the CDC. It also influences - depending on the selected metric - the number of searched tracks in p_t direction: The further the point of intersection is off the center, the less tracks are predefined. The distance must be greater than 0 and smaller than the radius of the time-of-flight detector. Reasonable values are e.g. between 200 and 600 mm, most simulations were done using a value of 600 mm.

4.2.5 Track width

Depending on how the bunch of predefined tracks and the grid of CDC hits fit together, several neighboring hits are assigned to the same track or vice versa. The lookup tables are designed to assign to each p_t value the best-fitting track (phi value). This property can be modified here.

a) Calculate

When using this setting, the maximum deviation in phi values from the calculated value, which does not lead to the next neighboring hit, is exactly calculated. This number is of course dependent from many parameters, e.g. the wire number, the drift time, the phi and p_t distribution etc. An additional Overlap is added. The calculation of the lookup table is rather computative with this setting. Also the histogramming can cause problems: As not for every hit the same number of histogram counters must be incremented, the acceleration of this task on an FPGA can be difficult.

b) Fixed +/-1

This setting assigns for each hit to each p_t value the track with the best fitting phi value and in addition both tracks adjacent in phi. When histogramming exactly 3 counters in each p_t block must be incremented for each hit.

c) Zero width +/-0

This setting assigns for each hit to each p_t value only the track with the best fitting phi value. When histogramming exactly 1 counter in each p_t block must be incremented for each hit. This setting bears, depending on the distribution of predefined tracks and CDC hits, the risk, that not every possible track is assigned to hits and thus can cause gaps in the histogram, that in turn can lead to wrong track candidates.

4.2.6 Inner Overlap [mm]

If in the selector width is set to exactly calculate the width of each track, an additional overlap with neighboring tracks can be defined here. This overlap can linearly vary from the inner (r = 20 cm) to the outer bound (r = 80 cm) of the drift chamber. Here the limit at the inner bound of the CDC is specified. A possible value is e.g. 2 mm.

4.2.7 Outer Overlap [mm]

If in the selector width is set to exactly calculate the width of each track, an additional overlap with neighboring tracks can be defined here. This overlap can linearly vary from the inner (r = 20 cm) to the outer bound (r = 80 cm) of the drift chamber. Here the limit at the outer bound of the CDC is specified. A possible value is e.g. 4 mm.

5 The dialog tab "Detector"

This dialog tab collects information about the connected detectors (drift chamber, time-of-flight) like geometry and resolution. The used bandwidth - channels and time-of-flight resolution - is also adjusted here.

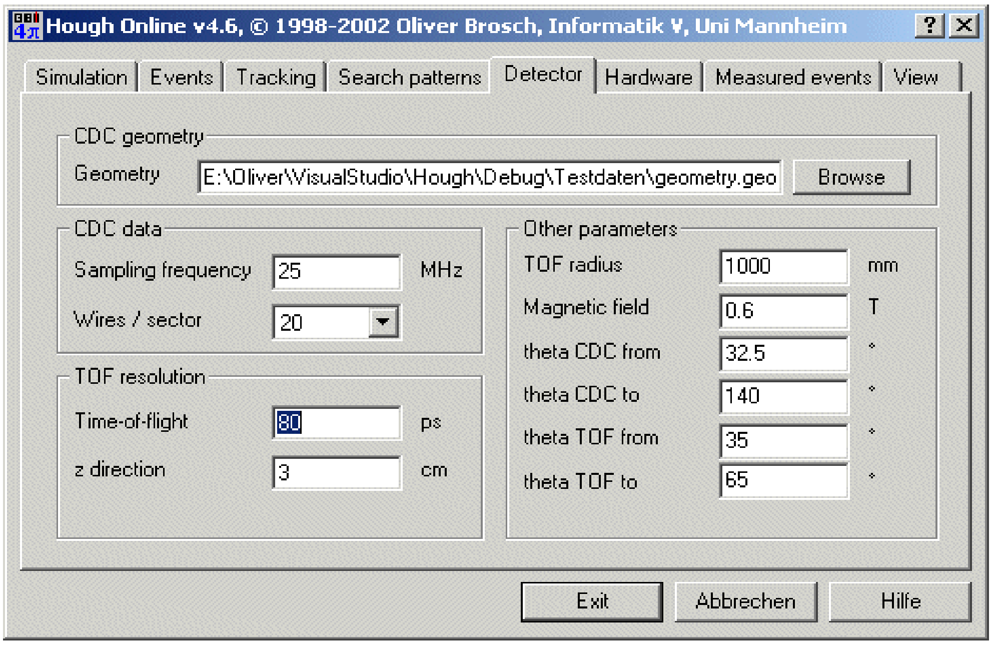
5.1 CDC geometry (dialog tab "Detector")

The file containing the geometrical composition of the drift chamber's signal wires is specified here.

5.1.1 Geometry

Here the storage location of the file containing information about the geometrical arrangement of all signal wires of the central drift chamber (CDC) is specified. Clicking the button "Browse" opens a dialog box for selecting a file using the mouse. The file must meet the following format convention:

- Each line names the position of a sense wire in x and y direction, that is perpendicular to the beam direction.
- Each line starts with two integer values for orientation, that should represent a sector number (counterclockwise) and a wire number (from inner to outer wires). After that



CDC geometry

Geometry: E:\Oliver\VisualStudio\Hough\Debug\Testdaten\geometry.geo

CDC data

Sampling frequency: 25 MHz

Wires / sector: 20

TOF resolution

Time-of-flight: 80 ps

z direction: 3 cm

Other parameters

TOF radius: 1000 mm

Magnetic field: 0.6 T

theta CDC from: 32.5 °

theta CDC to: 140 °

theta TOF from: 35 °

theta TOF to: 65 °

follow the x and y values of the wire's position as floating point numbers in cm. The beam axis represents the origin.

- The lines must be sorted first by sector and then by wire number in ascending order.

If all requirements are met, the beginning of this file looks like follows:

```
1 1  19.1793003  7.94280005
1 2  20.1795998  7.95949984
1 3  21.1753006  8.05609989
1 4  22.1756001  8.07279968
1 5  23.1713009  8.16940022
1 6  24.1716003  8.18620014
```

It is important that both pre-simulated and measured events are based on the same detector geometry, that is contained in the specified file. The whole file (valid since 1998) can be downloaded here.

5.2 CDC data (dialog tab "Detector")

Here the bandwidth of the drift chamber's data that should be used is specified. This concerns the number of channels and the time-resolution of the drift chamber's signals.

5.2.1 Sampling frequency [MHz]

The drift time resolution, which gives the number of possible values, that the drift time of ionization tracks created by the charged particles can assume, is responsible for the accuracy of the track recognition on one hand, but on the other hand also for the memory consumption of the lookup tables. The sampling frequency has no influence on the computing speed. The total drift time is maximum about 5 µs. Assuming a sampling frequency of 50 MHz this results in 250 different possible values for the drift time. Hits created from the drift time occupy a grid of 1 mm granularity when assuming a drift velocity of 0.005 cm/ns. The sampling frequency can e.g. be set to 25 MHz (2 mm grid). It is important to take into account neighboring tracks, that must remain distinguishable.

5.2.2 Wires / sector

The readout of the signals not of all 960 wires of the central drift chamber (CDC) has a few advantages:

- Less cabling. For each readout wire a cable must be laid to the trigger.
- Less effort for preprocessing of the data. All channels have to be computed in parallel on the employed hardware.
- Faster computation: All hits, whose number is proportional to the number of used wires, enter the histogram one after another using a Hough transform.

When using less wires only the disturbing influence of single, wrongly recognized hits on the histogram is growing, and thresholds can be calibrated less precisely. Reasonable numbers of wires per sector to be used are 15, 20, or 30, corresponding to 240, 320, or 480 wires altogether. When using the setting "60", all wires are readout and used for track finding.

5.3 TOF resolution (dialog tab "Detector")

The time-of-flight detector has a non-zero resolution in space (z direction) and time, that have to be specified here.

5.3.1 Time-of-flight [ps]

The time-of-flight of particles hitting the time-of-flight detector can not be measured exactly. Responsible for that is not only the detector technology used, but also the readout electronics. After creation or reading of particle information the time-of-flight data will thus be smeared using a standard distribution, the resolution being its standard deviation. According to recent measurements the achievable resolution is in the order of 80 ps.

5.3.2 z direction [cm]

The limited resolution of the time-of-flight detector in the direction of the beam axis can be explained by the used technology. From the created or read particle information first the point, where the time-of-flight detector is hit in z direction, is calculated. Then the z direction is smeared with a standard distribution to take care of this fact. The resolution is the distribution's standard deviation and is around a few (e.g. 3) cm.

5.4 Other parameters (dialog tab "Detector")

Geometrical dimensions of drift chamber and time-of-flight detector. Above this, the magnetic field is necessary for the classification of recognized particles.

5.4.1 TOF radius [mm]

The time-of-flight detector consists of tubes, that are mounted around and in parallel to the beam axis. The distance of the detector elements mounted inside the tubes to the beam axis must be specified here. It is 1,000 mm by default. It is important, that both pre-simulated and measured events are based on the same radius, that is specified here.

5.4.2 Magnetic field [T]

The magnetic field is responsible on the one hand for the curvature of the tracks of charged particles, on the other hand for the direction of the charge drift. The magnetic field, which is nearly homogeneous in the middle of the detectors, is the basis for both pre-simulated and measured events and is later necessary for the calculation of important particle properties like the mass. Thus, in both cases the same value must be used. The default value for the magnetic field inside the FOPI detector is 0.6 T.

5.4.3 theta CDC from [°]

This is the smallest polar angle that is covered by the FOPI detector's central drift chamber (CDC). It is measured as the deviation from the beam direction and is 32.5° for the CDC in default position.

5.4.4 theta CDC to [°]

This is the largest polar angle that is covered by the FOPI detector's central drift chamber (CDC). It is measured as the deviation from the beam direction and is 140° for the CDC in default position.

5.4.5 theta TOF from [°]

This is the smallest polar angle that is covered by the FOPI detector's time-of-flight detector (TOF). It is measured as the deviation from the beam direction and is 35° for the TOF in default position.

5.4.6 theta TOF to [°]

This is the largest polar angle that is covered by the FOPI detector's time-of-flight detector (TOF). It is measured as the deviation from the beam direction and is 65° for the TOF in default position.

6 The dialog tab "Hardware"

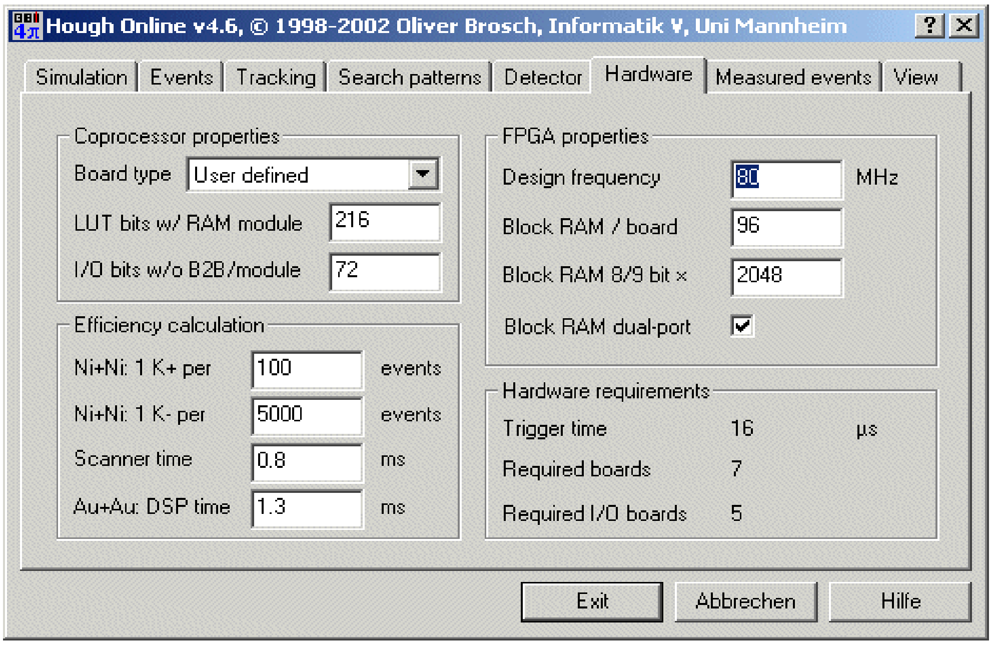
The hardware simulator can give an estimation of the amount of hardware required based on information about the features of the hardware employed. With the help of further adjustables regarding readout electronics and kaon acceptance an estimation of the efficiency of the simulated trigger system is also possible.

6.1 Coprocessor properties (dialog tab "Hardware")

Defines the architecture of the accelerator boards used. It is important, which data width is available for access to the predefined tracks and for data in- and output.

6.1.1 Board type

This drop-down list adds for some known types of FPGA based accelerator hardware the necessary values autonomously. These are the memory bandwidth, the I/O bandwidth, the number and the size of available RAM blocks. The individual settings are:



Coprocessor properties

Board type: User defined

LUT bits w/ RAM module: 216

I/O bits w/o B2B/module: 72

Efficiency calculation

Ni+Ni: 1 K+ per 100 events

Ni+Ni: 1 K- per 5000 events

Scanner time 0.8 ms

Au+Au: DSP time 1.3 ms

FPGA properties

Design frequency 80 MHz

Block RAM / board 96

Block RAM 8/9 bit × 2048

Block RAM dual-port ☑

Hardware requirements

Trigger time 16 µs

Required boards 7

Required I/O boards 5

a) User defined

This disables the preselection, so all values can be tuned individually.

b) Atlantis AIB (2xXCV600)

By this the Atlantis I/O board is selected as the target architecture. This board offers connectors for a CompactPCI backplane, 4 slots for expansion modules (memory, I/O) and 4 banks of fast SRAM (each 36 bits wide). Two modules of the type Virtex XCV600 from Xilinx are the available FPGA modules.

c) microEnable II (XCV1000)

This PCI plugin card carries an FPGA Xilinx Virtex XCV1000. In addition, 2 banks of SRAM and 3 expansion slots (for memory or I/O modules) are available. microEnable II is a product of Silicon Software GmbH.

d) RACE-1 (XC2V3000)

RACE-1 is a plugin card for the PCI bus with 64 bits and 66 MHz support. As FPGA a Xilinx Virtex II (XC2V3000) is used. In addition, the board offers integrated SRAM (4 banks with 36 bits width each) and 4 slots for additional expansions (memory or I/O modules, 36 bits data width each).

6.1.2 LUT bits w/ RAM module

How many bits of memory bandwidth are available for use as a lookup table, must be specified here. If expansion slots are equipped with additional memory, it will be taken into account as well. The memory depth will be sufficient anyway and is therefore not listed separately. The setting is based on the assumption that every board will initially be used equally for data collection and Hough computation. Ideally the number of necessary computing and I/O boards are more or less identical.

6.1.3 I/O bits w/o B2B/module

The total available I/O bandwidth is specified here. It has to be taken into account that when equipping the boards with additional memory modules not all channels are available any more. They thus must not be declared here. Furthermore, the setting is based on the assumption that every board will initially be used equally for data collection and Hough computation. Ideally the number of necessary computing and I/O boards are more or less identical.

6.2 FPGA properties (dialog tab "Hardware")

For the estimation of the amount of hardware needed additional information about the FPGAs applied is necessary.

6.2.1 Design frequency [MHz]

The achievable design frequency on the FPGA in use. Please note that all steps (reading and writing of RAM blocks, reading of the lookup table,...) have to be executed at this speed. It has a major influence on the usefulness of the whole trigger system. It is assumed that the design frequency is at least as high as the frequency at which the detector signals are digitized.

6.2.2 Block RAM / board

The histogram is internally stored inside the FPGAs. Depending on how many RAM blocks are available within an FPGA, the number of histogram blocks that can be stored in the FPGA varies accordingly. A single RAM block is always used exclusively for a single histogram block, that is a block containing entries with the same p_t values. Depending on the histogram dimensions and the size of the RAM blocks, possibly several RAM blocks are used for a single p_t block.

6.2.3 Block RAM 8/9 Bit x

The size of an FPGA internal RAM block - when configured for 8 or 9 bits per data word - together with the number of these blocks determines how many FPGAs are necessary for the storage of the track histogram.

6.2.4 Block RAM dual-port

With certain FPGAs - e.g. Xilinx Virtex II - the internal block RAM can be configured to be dual-ported. As this enables simultaneous read and write access to this memory, the latency between the histogramming of two CDC hits vanishes, so the histogramming is then twice as fast.

6.3 Efficiency calculation (dialog tab "Hardware")

Additional information about the detector's kaon acceptance and the speed of the readout electronics make an estimation of the planned trigger's efficiency possible.

6.3.1 Ni+Ni: 1 K+ per ... events

Because events that were simulated or created by Hough Online always contain exactly one kaon (that need not necessarily be able to be found, because the information about its track and momentum need not be complete), the algorithm for efficiency calculation needs the additional information, how many positively charged kaons (K+) can be expected in real events - taking also into account the acceptance of the FOPI detectors. This information is based upon events with Ni+Ni @ 2 AGeV. This value rises approximately with the square of the average number of expected tracks within the CDC. Because this number is small its reciprocal value - meaning how many events are passing in average before one containing a kaon can be found - is inserted here.

6.3.2 Ni+Ni: 1 K- per ... events

In analogy to the information about expected kaons the algorithm for efficiency calculation also needs information about the number of expected anti-kaons (K-) within real events. This information is again based upon events with Ni+Ni @ 2 AGeV.

6.3.3 Scanner time [ms]

The scanner time is the main reason for the employment of a trigger in the FOPI experiment: after the detection of an event - which in the central drift chamber (CDC) and the connected FADCs is completed after approximately 5 µs - the FADCs must hold their data until they are fetched by the FADC scanners. For this time the detector is thus blocked for new events. The trigger, that directly and without delay computes the FADCs' data, can avoid the scanning progress for events that are rated non-relevant, thus saving time. The time lost by the scanners is about 0.8 ms. For events that the trigger rates relevant the scanning progress has of course to be called anyway.

6.3.4 Au+Au: DSP time [ms]

The FADC scanners already during the scan progress pass the readout data to a DSP stage used for data reduction. This stage needs time and is thus during its calculations not available for new events. For events of the type Au+Au @ 1.5 AGeV the DSPs need approximately 1.3 ms per event. This time scales linearly with the average number of tracks in the CDC. If scanners and DSPs work together ideally, the average time at least passing between two events is the maximum out of scanner and DSP time. The trigger can avoid this step for events rated non-relevant.

6.4 Hardware requirements (dialog tab "Hardware")

The requirements calculated based upon the parameters specified above and the additional trigger latency per event are displayed here.

6.4.1 Trigger time [µs]

For the time, that the trigger finally consumes to make its decision, the following variables are relevant: the time for histogramming which is composed of the number of tracks per event, the number of readout wires, the width of the predefined tracks and the design speed of the used FPGAs, and the time for the maximum search including filters which depends on the number of possible starting angles of the predefined tracks and again the design frequency. The trigger time can be calculated only after a simulation run, because not all relevant variables are known in advance.

6.4.2 Required boards

All variables relevant for the size of the track histogram and the lookup table as for the FPGA and board resources are used for the estimation of the number of computing boards that are necessary for the calculation of the histogram and the extraction of the track candidates. If all boards are used simultaneously for the calculations and for data input and preprocessing, this number should be in the order of the number of required I/O boards.

6.4.3 Required I/O boards

Above all, the boards' free I/O bandwidth and the number of wires to be read from the CDC are responsible for the number of boards needed for data acquisition. If all boards are used simultaneously for the calculations and for data input and preprocessing, this number should be in the order of the number of required computing boards.

7 The dialog tab "Measured events"

The event display for real, measured events - including track recognition - is an own mode of operation of Hough Online. The required files are specified here.

7.1 Event files (dialog tab "Measured events")

In this area the files, that contain the data from a single event to be used for analysis, must be specified. The three single files must unconditionally belong to the same event, otherwise the computations and displays would make no sense. Files belonging to the same event can in most cases be identified by their file names, which should be composed in a similar way as postulated for externally created events.

7.1.1 CDC data

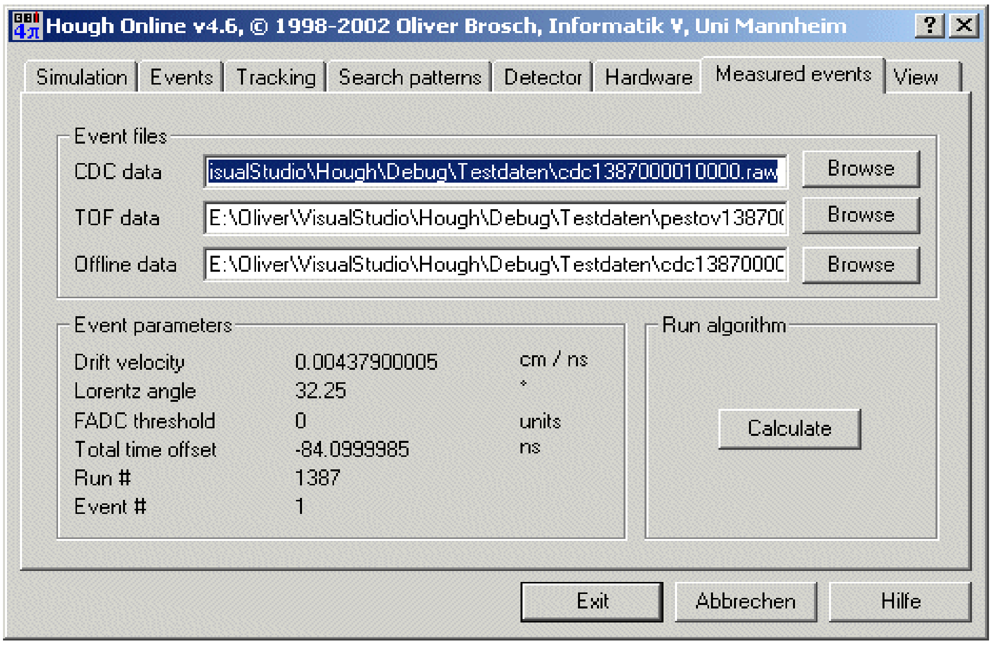
The path to a file with data from the central drift chamber (CDC) is entered in this field. This file may either contain CDC raw data or hit information generated from them. In the first case the data must be in ".raw" format, in the second case in ".atim" format. Clicking the "Browse" button opens a dialog box for choosing a file with the mouse.

a) Data in ".raw" format

Files of this type must have the extension ".raw" and exactly meet a specified format. The first line contains, one after another,

- the drift velocity v_D in cm/ns,
- the measured Lorentz angle phi_L in °,
- a global drift time offset t_{off} (all floating point values) and two other numbers, that may represent
- a run and
- an event number (integer values).

From the second line follow digitized pulse information, that are built as follows: the lines begin with



CDC data isualStudio\Hough\Debug\Testdaten\cdc13870000100000.raw

TOF data E:\Oliver\VisualStudio\Hough\Debug\Testdaten\pestov138700

Offline data E:\Oliver\VisualStudio\Hough\Debug\Testdaten\cdc1387000C

Event parameters

```
Drift velocity      0.00437900005   cm / ns
Lorentz angle       32.25           °
FADC threshold      0               units
Total time offset   -84.0999985     ns
Run #               1387
Event #             1
```

- the sector number,
- the wire number and
- the offset, from which the actual pulse snippet starts, in units of 10 ns (all integer values).

Two digitized pulse samples follow, for the signal read from the left and the right end of the sense wire, in FADC units. These values are integer. The values of two adjacent lines with the same beginning belong to the same pulse snippet and have a difference in time of 10 ns each. The beginning of such a file may e.g. look like this (sector and wire numbers as well as the offset are highlighted):

```
 0.00437900005  32.25 -84.0999985 1387 1
1 1 26 8 1
1 1 26 27 0
1 1 26 31 0
1 1 26 21 0
1 1 45 0 5
1 1 45 0 6
1 1 45 43 11
1 1 45 122 42
```

b) Data in ".atim" format

These files must have the extension ".atim" and exactly meet the following format convention. The first line contains, one after another,

- the drift velocity v_D in cm/ns,
- the measured Lorentz angle phi_L in ° (both floating point values),
- the threshold, that was used to externally generate the hit information, and two other numbers, that may represent
- a run and
- an event number (all integer values).

From the second line follow the hit information, that are built as follows: the lines start with

- the sector number and
- the number of the wire, on which the hit occurred (both integer values).

Then follow the drift time t_D of the detected edge of the hit's pulse in ns (floating point value) and the consecutive number of the hit on the current wire (integer), which is not further used. The beginning of this file may e.g. look as follows (sector and wire numbers are highlighted):

```
 0.00437900005  32.25 50 1387 1
1  1  415.9  1
1  1 1335.9  2
1  2 1375.9  1
1  3  855.9  1
1  3  985.9  2
1  4 1335.9  1
1  5 1095.9  1
1  5 1355.9  2
```

7.1.2 TOF data

The storage location of a file with (measured or calculated after an external analysis) time-of-flight information is specified in this field. Clicking the "Browse" button opens a dialog box for choosing a file with the mouse. The file must have the following format: the first line has the same contents as the file with CDC data in ".raw" format. From the second line follow time-of-flight records, that are built as follows: the first and the last number is a consecutive number (integer) after and before possibly incomplete records were deleted. They are not used in the algorithm. In between stand, one after another,

- the time-of-flight t_F of the particle in ps,
- the angle phi$_{TOF}$ of the spot, where the TOF detector was hit, in °, measured in the plane perpendicular to the beam direction,
- the component in z direction of the distance of the TOF hit from the origin in cm and
- the angle-of-emission of the particle from the CDC, which is not further used (all floating point values).

The last line contains three integer values, that are also not used, and that represent the number of rejected tracks and the total number of time-of-flight hits after and before records were deleted. All information, that is not used, is only available after an external track analysis. To ensure consistency with the file format, they must be present (with arbitrary values) in any case. This file may e.g. look as follows:

```
 0.00437900005  32.25 -84.0999985 1387 1
1  4959.04492  94.3754883  80.0731125  53.3329391 1
2  24590.9414  104.344574  74.763176  54.1729774 2
3  8717.58887  206.014648  55.568409  62.3951035 6
4  12875.5225  331.425171  13.8388004  83.3348389 8
5  22306.4805  340.970978  59.6928825  60.6469803 9
6  4800.33984  0.728515625  47.2390175  53.174633 10
7  5110.29053  5.87234497  39.0371132  75.7743988 11
4 7 12
```

7.1.3 Offline data

Files carrying offline data contain information about the tracks that were found in an external analysis, and the hits in the central drift chamber (CDC) that were assigned to them. They are necessary not for tracking, but for comparison of the results with those of alternative tracking systems. Clicking the "Browse" button opens a dialog box for choosing a file with the mouse. This file must have the following format: the first line contains, one after another,

- the drift velocity v_D in cm/ns,
- the measured Lorentz angle phi_L in ° (both floating point values),
- the threshold, that was used to externally generate the hit data, and three additional numbers, that may represent
- the total number of found tracks,
- a run and
- an event number (all integer values).

Information about the found tracks follow, that are built as follows: the first line contains

- a consecutive number (integer),
- the radius[1] r_C in cm and
- the starting angle[2] phi_S in ° (both floating point values).

Comments: [1] The tracks of positively charged particles are right-curved, the tracks of negatively charged particles are left-curved. [2] The starting angle is measured directly at the center-of-collision in the plane perpendicular to the beam direction.

After that follows a list of the hits that were assigned to the current track - one per line: each with sector and wire number and the consecutive number of the hit on the current wire, if all recognized hits on a wire are sorted and ascendingly numbered by their drift time. These information are fixedly linked both to the underlying threshold and also to the one to be used for the algorithm. They will though not be used for the algorithm and can thus be left out. If they however exist, they must be integer values, each followed by a dot ".". The beginning of this file may followingly look like this (the consecutive track numbers are highlighted):

```
 0.00437900005  32.25 50 54 1387 1
1  126.231041  52.0290756
 2.  60.  1.
 2.  59.  1.
 2.  56.  1.
 2.  53.  2.
 2.  52.  2.
2  258.265076  115.53772
 5.  60.  1.
 5.  59.  2.
 5.  58.  2.
```

7.2 Event parameters (dialog tab "Measured events")

Some detector parameters, that were measured together with the event, are necessary for computing the Hough algorithm and are displayed here.

7.2.1 Drift velocity [cm / ns]

Here the drift velocity is displayed, with which the charge created by the particles' tracks move towards the signal wires. It is calculated from quantities measured inside the detector (magnetic field, gas pressure, electric drift field).

7.2.2 Lorentz angle [°]

The Lorentz angle is the deviation of the path of the charge drift to a line horizontal to the signal wires' plane of the CDC. It is calculated from quantities measured inside the detector (magnetic field, gas pressure, electric drift field).

7.2.3 FACD threshold [units]

In the case of measured events with drift chamber data in ".atim" format the data are already available in hit format, which is otherwise generated by the algorithm step "Discriminator". The threshold that was used for the external creation of these data is displayed here.

7.2.4 Total time offset [ns]

The total time offset is a constant, that must be added to all drift times contained in a file with drift chamber data in ".raw" format, in order to obtain the real drift time of the pulse samples.

7.2.5 Run

The number of the FOPI run specified in the files is displayed here.

7.2.6 Event

The consecutive number of the current event within the FOPI run specified in the files is displayed here.

7.3 Run algorithm (dialog tab "Measured events")

By pressing this button the computation of the Hough algorithm is initiated.

7.3.1 Calculate

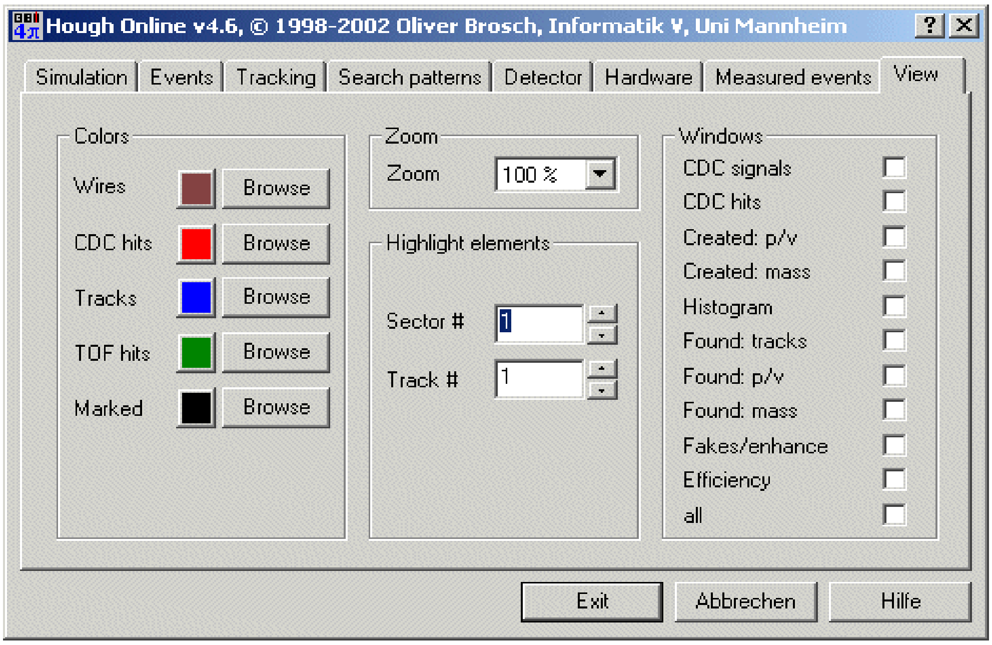
A click on this button starts the computation of the Hough algorithm, using a single, previously measured event, that has to be specified above. This opens a new dialog window to display the computation's progress. If a computation is done for the first time, or if settings for the predefined search patterns or the data rate of the central drift chamber have been modified since the last run, a new lookup table is calculated as the first step.

8 The dialog tab "View"

The dialog tab "view" defines, which windows of the event display and the simulation should be visible. This visualization can be further adapted by adjusting colors and size.

8.1 Colors (dialog tab "View")

With these controls the colors used in the different display windows can be modified for better visibility or contrast. Clicking one of the attached "Browse" buttons opens a color dialog where a new color can be chosen or defined. The current color is displayed on the button next to the browse button. The background color is always white.



Wires 🟫
CDC hits 🟥
Tracks 🟦
TOF hits 🟩
Marked ⬛

Zoom
Zoom 100 %

Highlight elements
Sector # 1
Track # 1

Windows
CDC signals □
CDC hits □
Created: p/v □
Created: mass □
Histogram □
Found: tracks □
Found: p/v □
Found: mass □
Fakes/enhance □
Efficiency □
all □

8.1.1 Wires

This button chooses the color that is used to display

- coordinate systems and ticks
- raw sense wire signals
- CDC sector boundaries
- CDC sense wires

8.1.2 CDC hits

The color that is chosen here is used for

- FADC comparator signals
- CDC hits
- boundaries for maximum momentum, velocity and drift time
- range of kaon acceptance
- track histogram bars above threshold
- efficiency, fake and enhancement bars for anti-kaons

8.1.3 Tracks

The items displayed in the color chosen here are

- tracks in the CDC
- bars in the mass histograms
- track histogram bars below threshold
- efficiency, fake and enhancement bars for kaons

8.1.4 TOF hits

Clicking this button chooses a color used to display the following items

- TOF hits in the CDC
- matched tracks in the CDC

- kaon tracks in the CDC
- all diagram labels and numbers
- borders around efficiency, fake, enhancement and track histogram bars

8.1.5 Marked

The color that can be chosen here is used to display

- highlighted tracks in the CDC
- parameters of highlighted tracks
- values of efficiency, fake and enhancement bars

8.2 Zoom (dialog tab "View")

The zoom factor used in all event display windows is adjusted with this control.

8.2.1 Zoom

This drop-down selection box allows to set the scale, with which the contents of the windows of the event display are painted. For normal display resolutions a zoom factor of 75% to 150% should be best suited to have the images in total on the screen. But the scale is a little bit more than just a zoom factor: only the dimensions of objects, but not the line widths are subject to the scale. So with a larger zoom factor smaller details might come into view.

8.3 Highlight elements (dialog tab "View")

In some event display windows certain elements (tracks) can be highlighted to display their exact parameters. For some windows the sector to be displayed is adjusted.

8.3.1 Sector

In two of the windows of the event display - CDC signals and histogram - only one of the 16 sectors of the full angular range is displayed. Which sector this should be is adjusted here. The contents of the windows are refreshed when another input control is selected, or if the selection is done by clicking one of the arrow buttons.

8.3.2 Track

In two of the windows of the event display - CDC hits and found tracks - particle tracks are displayed. For a single particle additional information - starting angle and radius - can be displayed by selecting the desired track with this control. The track will, as soon as another input control is selected, or if the selection is done by clicking one of the arrow buttons, be highlighted as bold and with a different color. The number of tracks displayed in the two windows is usually different. If the track number adjusted here exceeds the number of tracks displayed in one of the windows, no track is highlighted in that window.

8.4 Windows (dialog tab "View")

Here the single windows of the event display are switched on and off. At program startup, all windows are hidden by default. Checking the box next to the label opens the corresponding window. The window can be closed again by unchecking the box or by clicking the close button in the window's caption. All windows can also be maximized or minimized with the corresponding buttons, or resized by dragging one of the window's borders or edges.

8.4.1 CDC signals

If not measured events with CDC data in ".atim" format are used, this window displays for the current event the raw data of the CDC's sense wires' signals for each wire (not only those that are used later) of a single sector. The wires are numbered from the inner to the outer bound of the CDC. The signals start from the left at the time-of-collision, and end at the left after the maximum drift time of 5 µs, which is equal to 250 sampling points at 50 MHz (not only those that are used later due to a lower sampling frequency). In addition, the comparator output signal of the FADCs is painted for each of the wires. It is indicated by horizontal lines of fixed length, only if it is high. The maximum drift time, that results from charges drifting from the left and right borders of the current sector to the wires in the middle, is indicated by two diagonal lines.

8.4.2 CDC hits

In this window, a scratch of the CDC is shown, together with the hits, tracks and TOF hits of the current event. These are the input data for the Hough algorithm, so only the relevant data are shown. All data - except the CDC hits if not measured events with CDC data in ".atim" format are used - come directly from the event files or the event generation. One of the tracks is highlighted and its parameters - starting angle and radius - are displayed. Tracks that belong to kaons are also marked in a different color. An event ID is displayed in the upper left corner of the window, that is, depending on which type of events are used, composed from the path name, and the run and event number.

8.4.3 Created: p/v

All particles that were created or read for the current run and parameter set are displayed in this particle histogram. This histogram is sorted by the particles' transversal momenta and velocity. Each square stands for a bin that particles were able to be assigned to. The color of the square corresponds to the number of particles in that bin, according to the scale at the right of the window. The upper half of the diagram represents positively charged particles, the lower half negatively charged particles ("negative momentum"). For a sufficient number of events, the particles group around curved lines, where the different particle species are supposed to be found. The strongest lines are (upper half from bottom right) pions and protons, and (lower half) anti-pions. In between the pions and protons is the line, where the kaons are expected to be. A region around this line is indicated by two curves in a special color. The corresponding region for anti-kaons can be found down left from the anti-pions. The upper boundaries for kaon momenta and velocities are also indicated by lines.

8.4.4 Created: mass

The same particles as in the p/v diagram are plotted in this mass diagram. The particles are assigned to mass bins, and the mass bins are represented by bars, whose heights stand for the particle counts in that bins. Positively charged particles can be found on the right, negatively charged particles on the left ("negative mass"). For a sufficient number of events, the mass bins group to form peaks. The strongest peaks stand for (from left to right) anti-pions, pions, and protons. Kaons and anti-kaons are expected to be found at the center of the regions between pions and protons, and left from the anti-pions, respectively. These regions are marked in a special color on the mass axis.

8.4.5 Histogram

Every bin in the track histogram represents one of the predefined tracks. The bins are counters that hold the number of coincidences of Hough-transformed CDC hits with the predefined patterns. So a circular track in the CDC that originates from the center-of-collision will cause a peak in the histogram that is centered just around the bin that represents the parameters of that track. The histogram bins are sorted by the starting angle phi_S and radius r_C and represented by bars. The color of the bars visualizes the value of that bins in the null-filtered histogram. A legend at the bottom left shows the assignment of counter values to colors. If this value exceeds the histogram threshold, the bin is displayed in another color. The height of the bars represent the same values, if the maximum search is done in the null-filtered histogram. If it is done in the original histogram instead, the corresponding values in this histogram are used. So the histogram shows both values that are used for peak search and maximum finding at the same time. In analogy to the 16 sectors of the CDC, the histogram is divided into 16 sectors, too. Only one of these sectors (and half of both neighboring sectors) is displayed.

8.4.6 Found: tracks

The found tracks window is similar to the CDC hits window, the difference is that the tracks that were found by the Hough Online algorithm are displayed rather than tracks that were created or read from a file. Particle candidates, i.e. tracks that could be matched to hits of the time-of-flight detector, are drawn in a different color.

8.4.7 Found: p/v

This is the same display as the created p/v diagram. But this time the particle candidates found by the Hough Online algorithm are shown rather than tracks that were created or read from a file.

8.4.8 Found: mass

This is the same display as the created mass diagram. But this time the particle candidates found by the Hough Online algorithm are shown rather than tracks that were created or read from a file.

8.4.9 Fakes/enhance

This window contains six different diagrams that visualize different aspects of the simulation analysis on a per-event basis, using all events of the current run and parameter set. The difference between the upper and lower row of diagrams is that they show results for different track multiplicity ranges. The track multiplicity is the number of particles that are contained in the original event data. The upper row of diagrams shows results for the multiplicity ranges alone to allow a close look at possible problems that may occur with these events, in the lower row multiplicity ranges are combined up to a maximum multiplicity, which gives an estimation for the full experiment with events in the particular range.

a) Found kaons

This diagram shows the kaon efficiency, i.e. the percentage of kaons and anti-kaons that could be found by Hough Online with the given parameter set. The numbers in or above the bars are the absolute numbers of kaons that were found, and in [brackets] the number of kaons contained in the original events.

b) Fake kaons / # events

The bars in this diagram represent the percentage of events, where a kaon was found though no kaon was present in the original data. The numbers in or above the bars are the absolute numbers of events with falsely found kaons, and in [brackets] the total number of analyzed events.

c) Kaon yield / experiment time

The kaon yield or kaon enhancement factor is the factor by which a trigger based on the Hough Online algorithm and the specified hardware can increase the number of kaons that can be found during a given period of time, when compared to the same experiment done without the use of a trigger. Note that especially this information is only valid for real experiments in the case of a track multiplicity range that starts with 0 tracks / event.

8.4.10 Efficiency

These diagrams show the kaon efficiency, i.e. the percentage of kaons and anti-kaons that could be found by Hough Online within all events of the current run and parameter set. The colors of the fields represent the percentages, and their meaning is explained by the scale on the right. The numbers in the fields are the absolute numbers of kaons that were found, and in [brackets] the numbers of kaons contained in the original events. In contrast to the diagrams in the fakes/enhance window the events are separated not only by their track multiplicities, but also by the momenta of the kaons that were contained in the original data. Two rows at a time belong to the same kaon momentum range, the first representing kaon (K+) and the second anti-kaon (K-) events. The first two rows of the two lower diagrams show the same data as the found kaons diagrams in the fakes/enhance window.

8.4.11 all

A click on this control opens or closes all of the above windows depending on the check state of this control.

Appendix B: Application Development

This is the collection of software which is necessary or useful for the development of Hough Online. The stated versions are those which were actually used - other versions of these programs should also work in most cases.

1 Hough Online Application

- **Microsoft Windows** 2000 Professional with Service Pack 4 (German). A 32-bit Windows OS is necessary to run the other Microsoft applications.
- **Microsoft Visual C++** 6.0 Professional with Service Pack 5 (German). Service Pack 3 or higher is necessary.
- **Microsoft MSDN Library** October 2001. This includes the documentation for the MFC class library and the Windows API.
- **Microsoft HTMLHelp** 1.2. This is the help compiler which creates compressed HTML (.chm files).

2 Hough Online Documentation

- **Doxygen** 1.3rc3, available on www.doxygen.org. Doxygen automatically generates a source code documentation using "Doxygen-style" comments. A documentation is available at the same place.
- **Graphviz** 1.9, available on www.research.att.com. Graphviz is used by Doxygen to generate inheritance and collaboration diagrams.
- **HTML Editor Phase 5** 01.11.2001 (beta), available on www.meybohm.de. A good HTML editor for Windows to build additional documentation like this (German).
- **SelfHTML** 8.0, available on selfhtml.teamone.de. The best documentation for HTML with JavaScript and CSS (German).
- **Microsoft HTMLHelp** 1.2. The help compiler can automatically be called by Doxygen.

Acknowledgments

I would like to thank all the people, that have helped to make this work possible and a pleasant and valuable experience.

First of all, I want to thank Prof. R. Männer for giving me the opportunity to work in his research group, and Prof. N. Herrmann for supporting my project within the context of FOPI.

I would also like to thank my current and former colleagues in Mannheim, especially those who share or shared the same office with me – Stefan, Gerhard, Christian, Matthias M., Peter, Stephan, Gerrit, Sebastian, Kubi, Rüdiger – and all other members of the FPGA group and Acconovis – Aku, Harald, Holger, Matthias S., Erich, Andrei, Maoyuan, Klaus; the people in Heidelberg – Marc, Markus – and all members of the FOPI collaboration at GSI – in particular Yvonne. I could greatly benefit from their work, their experience, the fruitful discussions and the enjoyable working atmosphere.

Special thanks also for the excellent organizational support from Andrea and Christiane. Thanks to Marc and Yvonne for producing and supplying me with gigabytes of recorded and simulated event data. Thanks to Norbert, Reiner, Marc and Dirk B. for the helpful comments on my manuscript.

In particular, I want to thank my parents, who have done a lot to support me in every possible respect. And last but not least, thanks to all my friends and colleagues, especially those that I have not mentioned above, for a great time.

This work has been funded by the GSI University Program, the DFG Graduates Support Program and the Chair for Computer Science V of the University of Mannheim.