

Elektroproduktion neutraler Pionen
und
Entwicklung eines Steuerungssystems
für die Drei-Spektrometer-Anlage
am Mainzer Mikrotron

Dissertation
zur Erlangung des Grades
„Doktor der Naturwissenschaften“
am Fachbereich Physik
der Johannes Gutenberg-Universität
in Mainz

Markus Weis
geboren in Mainz

Institut für Kernphysik
Johannes Gutenberg-Universität Mainz
September 2003

1. Berichterstatter: Prof. Dr. Th. Walcher
2. Berichterstatter: Prof. Dr. H. G. Sander

Inhaltsverzeichnis

1. Motivation	1
2. Formalismus	5
2.1 Chirale Symmetrie	5
2.2 Kinematik und Multipolzerlegung	6
2.3 Streumatrix	10
3. Experimenteller Aufbau	13
3.1 Abbildungseigenschaften	14
3.2 Fokalebeneendetektoren	16
3.3 Datenerfassung	19
3.4 Target	19
4. Kalibrationskinematik	21
4.1 Kinematik	21
4.2 Szintillatorineffizienzen	24
4.2.1 Spektrometer A	24
4.2.2 Spektrometer B	25
4.3 Bahnrückrechnungsverfahren	26
4.4 Spurrekonstruktion	29
4.5 Fazit	32
5. Neutrale Pion-Elektroproduktion	33
5.1 Kinematiken	33
5.2 Ereignisidentifikation	33
5.3 Datenanalyse	35
5.4 MAID-Transport	39
5.5 Fehlerbetrachtungen	42
5.5.1 Zentralimpuls von Spektrometer B	44
5.5.2 Position der Targetzelle	44
5.5.3 Form der Targetzelle	48
5.6 Ergebnisse	49
6. Steuerung	55
6.1 Grundlegende Begriffe in der Regeltechnik	55
6.2 Anforderungen an das System	56
6.3 Ausgangspunkt: Das alte System (ECS)	58
6.4 Das neue Steuerungssystem (Mezzo)	62
6.4.1 Gerätetreiber	62
6.4.2 Zentraler Prozeß	64
6.4.3 Kommunikation	69
6.4.4 Ablaufsteuerung	71

6.4.5	Grafische Benutzerschnittstelle	73
6.4.6	Magnetfeldeinstellung der Spektrometer	74
7.	Zusammenfassung	79
A.	Datenanalyse	81
A.1	χ^2 -Minimierung	81
A.2	Wirkungsquerschnitte in tabellarischer Form	81
A.2.1	Kinematiken 0-6 bis 4-2	81
A.2.2	Kinematik 4-3	86
B.	Steuerung	87
B.1	Befehle und Fehlermeldungen	87
B.2	Datenbank	93
B.2.1	Gerätetreiber	94
B.2.2	Semaphore	94
B.2.3	Alias	94
B.3	Eigenschaftslisten	95
B.3.1	Analog Digital Konverter (ADC)	95
B.3.2	Digital Analog Konverter (DAC)	96
B.3.3	Digital In (DI)	97
B.3.4	Digital Out (DO)	98
B.3.5	Text	98
B.3.6	Alias	99
B.3.7	Database	102
B.3.8	Semaphore	103
B.3.9	TCP Interface	103
B.4	Gerätetreiber	103
B.4.1	ADAM Bus	103
B.4.1.1	4011, 4012, 4014D	104
B.4.1.2	4017, 6017	104
B.4.1.3	4050	105
B.4.1.4	4060	105
B.4.2	D1000 Drucksensor	105
B.4.3	DRC-91CA Temperaturkontroller	106
B.4.4	A&D Instruments FG-150 Waage	107
B.4.5	ILM200 Cryogen Level Meter	107
B.4.6	IPS120-10 Magnetnetzteil	108
B.4.7	LeCroy 4032a HV-Netzteil	109
B.4.8	LeCroy 1440 HV-Netzteil	111
B.4.9	LakeShore LS208 Temperatursensor	112
B.4.10	MPS8000 Magnetnetzteil	112
B.4.11	Out-of-Plane-Winkelauslese	113
B.4.12	Parallelport	115

B.4.13	PC20 MultiIO PC-Meßkarte	116
B.4.14	NRVD Leistungsmessgerät	117
B.4.15	Programmierbares Netzteil DIGI-35-CPU	117
B.4.16	WCSI Winkelauslese	118
B.4.17	KPH Wasserüberwachungssystem	118
B.4.18	Kurzschlußkompensation für Spektrometer C	119
B.4.19	PT2025 NMR Auslese	120
B.4.20	Phoenix Contact Interbus-S	121
	B.4.20.1 Bruker Magnetnetzteil	122
	B.4.20.2 Kollimator-Kontroller	123
	B.4.20.3 MultiIO-Kontroller	123
	B.4.20.4 NMR Kompensation	124
	B.4.20.5 VDC Netzteil	124
B.4.21	Datenerfassung AQUA	125
B.5	Skriptsprache	126
	B.5.1 Variablen und Variablendeklaration	126
	B.5.2 Prozeduren und Funktionen	127
	B.5.3 Anweisungen und Kontrollstrukturen	127
	Literaturverzeichnis	131
	Danksagung	135
	Lebenslauf	137

Kapitel 1

Motivation

Nach unserer heutigen Vorstellung sind fundamentale Wechselwirkungen stets mit einem Teilchenaustausch verbunden. In der starken Wechselwirkung ist das Gluon das Austauscheteilchen zwischen den Quarks. Die Quantenchromodynamik, die zugrundeliegende Theorie, definiert die Farbladung als zusätzlichen Freiheitsgrad für diese Objekte und fordert gleichzeitig, daß nur farbneutrale Teilchen existieren können. Dieses „Confinement“ erklärt, warum bisher keine freien Quarks, sondern nur farblose Kombinationen aus Farbe und Antifarbe bzw. drei verschiedenen Farbladungen beobachtet werden konnten. Der experimentelle Zugang zur starken Wechselwirkung im Grenzfall kleiner Energien ist daher beschränkt auf die effektiven Freiheitsgrade in Form von Mesonen und Baryonen. In den sogenannten effektiven Feldtheorien wird als Konsequenz nicht mehr mit Quarks und Gluonen, sondern nur noch mit den entsprechenden farblosen Teilchen gearbeitet.

Bei kleinen Energien kann die Hadron-Hadron-Wechselwirkung in sehr guter Näherung durch den Austausch von Mesonen beschrieben werden. Von besonderer Bedeutung ist das Niederenergiethorem (LET), welches auf der grundlegenden Lorentz- und Eichinvarianz sowie der PCAC-Relation (Partially Conserved Axial Current) basiert. Die Gültigkeit des LET sollte durch die Pion-Photoproduktion am Nukleon bestätigt werden.

Nachdem das LET in zwei Experimenten [Maz86, Bec90] bei der elektrischen Dipolamplitude E_{0+} verletzt zu sein schien, wurden zahlreiche Anstrengungen unternommen, diese Diskrepanzen sowohl von theoretischer als auch experimenteller Seite aus zu klären. Im Niederenergiethorem werden die Multipolamplituden an der Pion-Produktionsschwelle in Form einer Taylorreihe entwickelt. Da das LET dabei nicht zwischen kinematischen Variablen und internen Parametern unterscheidet, wird der quantitative Vergleich zwischen der Theorie und den gemessenen Daten erschwert [Sch93]. Die chirale Störungstheorie (ChPTh) hingegen ist eine Niederenergienäherung der QCD, die zwischen modellabhängigen und -unabhängigen Termen differenziert. Sie kommt zu dem Resultat, daß die Vorhersagen des LET bezüglich der E_{0+} -Amplitude unvollständig sind, weil Beiträge der Ordnung μ^2 (wobei $\mu = m_\pi/m_N$) nicht berücksichtigt wurden.

In der Pionproduktion am Proton sind die neutralen Produktionsamplituden proportional zu μ bzw. μ^2 [Sch91]. Dies betrifft sowohl die E_{0+-} als auch die L_{0+-} Multipolamplitude. Es liegt also nahe, zusätzlich zur Photoproduktion auch die Elektroproduktion neutraler Pionen zu verwenden, um die Vorhersagekraft des

Niederenergiethorems und insbesondere der ChPTh weiter zu untersuchen und gleichzeitig Erkenntnisse über die longitudinalen Multipole zu erlangen.

Das erste erfolgreiche Experiment zur π^0 -Elektroproduktion in Schwellennähe wurde am NIKHEF [Wel92] durchgeführt. Bei einem Viererimpulsübertrag von $Q^2 = 0,05 (\text{GeV}/c)^2$ und einer Schwerpunktsenergie bis $\Delta W = 1,5 \text{ MeV}$ über der Produktionsschwelle sowie einem $Q^2 = 0,1 (\text{GeV}/c)^2$ und $\Delta W = 2,5 \text{ MeV}$ wurde der totale Wirkungsquerschnitt aus den Daten ermittelt. Eine Trennung der Ladungs- und Stromanteile des Wirkungsquerschnitts fand nicht statt. Verglichen mit den Vorhersagen der chiralen Störungstheorie ergab sich eine gute Übereinstimmung zwischen den Daten und der Rechnung. Die statistischen Fehler waren allerdings relativ groß, da bei dem Experiment Elektronen aus einem Linearbeschleuniger mit einem Tastverhältnis von lediglich 1% zur Verfügung standen.

Weitere Experimente zur Elektroproduktion neutraler Pionen wurden an der Drei-Spektrometer-Anlage des Mainzer Mikrotrons MAMI durchgeführt. Das Mikrotron ist ein Dauerstrichbeschleuniger mit hoher Luminosität bei gleichzeitig geringem Strahlungsuntergrund, was zusammen mit der Drei-Spektrometer-Anlage ein idealer Standort für Pionproduktionsexperimente ist. Die Experimentreihe wurde mit einem Experiment [Dis96] begonnen, bei dem der gemessene Wirkungsquerschnitt mit Hilfe einer Rosenbluthseparation in den transversalen und longitudinalen Anteil zerlegt, und die helizitätsunabhängigen Interferenzterme ermittelt wurden. Der Viererimpulsübertrag des virtuellen Photons betrug $Q^2 = 0,1 (\text{GeV}/c)^2$ bei einer Schwerpunktsenergie bis $\Delta W = 4 \text{ MeV}$ über der Produktionsschwelle des neutralen Pions. Im Vergleich mit der chiralen Störungstheorie wurde die Winkelabhängigkeit der Wirkungsquerschnitte von der Theorie gut wiedergegeben, absolut jedoch um 20-30% unterschätzt. Für E_{0+} wurde im Experiment ein kleinerer Wert und für die L_{0+} -Amplitude ein größerer Wert als die theoretische Vorhersage bestimmt.

Das Folgeexperiment [Mer02] fand bei einem Viererimpulsübertrag von $Q^2 = 0,05 (\text{GeV}/c)^2$ und einer Schwerpunktsenergie bis ebenfalls $\Delta W = 4 \text{ MeV}$ über der Schwelle statt. Auch bei diesem Experiment wurde eine Rosenbluthseparation angewendet, um die transversalen und longitudinalen Anteile des Wirkungsquerschnitts trennen zu können. Die experimentell ermittelten Wirkungsquerschnitte lagen erheblich unter den Vorhersagen sowohl der chiralen Störungstheorie als auch des Modells MAID.

Im Rahmen dieser Arbeit wurde nun ein Experiment zur Elektroproduktion neutraler Pionen durchgeführt ($Q^2 = 0,05 (\text{GeV}/c)^2$, $\Delta W \leq 40 \text{ MeV}$), bei dem ein polarisierter Elektronenstrahl verwendet wurde, um zusätzlich den helizitätsabhängigen Anteil des Wirkungsquerschnitts bestimmen zu können. Dabei wurde die Möglichkeit der Drei-Spektrometer-Anlage ausgenutzt, Reaktionsprodukte auch außerhalb der Streuebene nachzuweisen, indem Spektrometer B aus der Ebene gekippt wird. Ein besonderes Interesse galt zusätzlich der Produktionsschwelle

des π^+ , die etwa 6 MeV oberhalb der des neutralen Pions liegt und wegen der Unitarität der Streumatrix zu einer Struktur in der s-Wellen-Amplitude führt.

Im apparativen Teil dieser Arbeit wurde das bestehende Steuerungssystem für die Detektorsysteme der Drei-Spektrometer-Anlage neu entworfen und implementiert, um die Portierung auf eine andere Rechnerarchitektur zu erleichtern und gleichzeitig höheren Stabilitätsanforderungen zu genügen.

Kapitel 2

Formalismus

2.1 Chirale Symmetrie

Ausgangspunkt zur Motivation der chiralen Störungstheorie ist die Lagrange-Dichte \mathcal{L} der QCD. Sie setzt sich zusammen aus einem Teil, der die Wechselwirkung zwischen Materie und Eichfeld beschreibt sowie einem Anteil, der die Dynamik der Gluonfelder inklusive ihrer Selbstwechselwirkung widerspiegelt:

$$\mathcal{L}_{QCD} = \sum_f \bar{q}_f (i \not{D} - m) q_f - \frac{1}{4} \sum_{a=1}^8 G_{\mu\nu,a} G_a^{\mu\nu} \quad (2.1)$$

Die Definition der kovarianten Ableitung $D_\mu q_f$ ist unabhängig vom Flavor und wirkt nur auf die Farbindizes. q_f kann geschrieben werden als

$$q_f = \left(\frac{1}{2}(1 + \gamma_5) + \frac{1}{2}(1 - \gamma_5) \right) q_f \equiv q_{R,f} + q_{L,f}, \quad (2.2)$$

wobei die Indizes L und R für den links- und rechtshändigen Teil stehen. Die Lagrange-Dichte lautet somit

$$\mathcal{L}_{QCD} = \sum_f (\bar{q}_{L,f} i \not{D} q_{L,f} + \bar{q}_{R,f} i \not{D} q_{R,f} - \bar{q}_{L,f} m q_{R,f} - \bar{q}_{R,f} m q_{L,f}) - \frac{1}{4} \sum_{a=1}^8 G_{\mu\nu,a} G_a^{\mu\nu}. \quad (2.3)$$

Die Materiefelder werden durch Fermionen mit Spin 1/2, die Quarks, gebildet. Sie kommen in sechs Arten, den sogenannten *Flavors*, vor und lassen sich anhand ihrer Masse in leichte und schwere Quarks unterteilen: $m_{u,d,s} \ll 1\text{GeV}/c^2 \leq m_{c,b,t}$. Betrachtet man beispielweise die Masse des Protons ($m_p = 938\text{MeV}/c^2$), so ist offensichtlich, daß ein anderer Mechanismus für die Erzeugung der Hadronenmassen verantwortlich sein muß, als die Summe der drei Current-Quarkmassen ($m_p \gg 2m_u + m_d$). Es liegt folglich nahe, den Grenzfall¹ $m_u, m_d, m_s \rightarrow 0$ zu untersuchen. Gleichung 2.1 reduziert sich dadurch zu

¹Man spricht in diesem Zusammenhang auch vom chiralen Limes.

$$\mathcal{L}_{QCD}^0 = \sum_f (\bar{q}_{L,f} i \not{D} q_{L,f} + \bar{q}_{R,f} i \not{D} q_{R,f}) - \frac{1}{4} \sum_{a=1}^8 G_{\mu\nu,a} G_a^{\mu\nu}. \quad (2.4)$$

Die Lagrange-Dichte im chiralen Limes besitzt eine globale $SU(3)_L \times SU(3)_R$ -Symmetrie, die sich im realistischeren Grenzfall mit $m_s \neq 0$ auf eine $SU(2)_L \times SU(2)_R$ -Symmetrie reduziert. Anschaulich betrachtet gibt es für jedes Quark einen linkshändigen (Spin antiparallel) und einen rechtshändigen Zustand. QCD-Wechselwirkungen sind die gleichen für beide Helizitäten und im Grenzfall masseloser Teilchen beeinflussen linkshändige $SU(3)$ -Rotationen die rechte Seite nicht und umgekehrt. Dies wird chirale Symmetrie genannt.

Zur chiralen $SU(3)_L \times SU(3)_R$ -Symmetrie gehören laut Noether-Theorem 16 erhaltene Ströme mit entsprechenden Ladungsoperatoren Q_L^a und Q_R^a ($a = 1, \dots, 8$). Es ist üblich, die Linearkombinationen $Q_V^a = Q_R^a + Q_L^a$ und $Q_A^a = Q_R^a - Q_L^a$ zu verwenden, die positive und negative Parität besitzen. Nimmt man nun an, daß der Grundzustand $|0\rangle$ nur invariant unter der vektoriellen $SU(3)$ ist, also

$$Q_V^a |0\rangle = 0, \quad (2.5)$$

andererseits aber die axialen Ladungen das Vakuum nicht vernichten,

$$Q_A^a |0\rangle \neq 0, \quad (2.6)$$

dann folgt aus dem Goldstone-Theorem [Ber74], daß jeder axiale Generator Q_A^a , der den Grundzustand nicht vernichtet, zu einem masselosen Boson führt, welches als pseudoskalares Pion bzw. Kaon identifiziert wird. Da der Grundzustand eine geringere Symmetrie als die Lagrange-Dichte besitzt, spricht man von spontaner Symmetriebrechung.

Die Vorhersagen der chiralen Störungstheorie können durch die bei Streuexperimenten gemessenen Wirkungsquerschnitte getestet werden. Dabei ist die Parametrisierung des Wirkungsquerschnitts in Strukturfunktionen eine geeignete Wahl der Observablen. Sie führt zu einer Entwicklung in Kugelflächenfunktionen, der sogenannten Multipolzerlegung. Im Rahmen des in dieser Arbeit vorgestellten Experimentes sind nur die Drehimpulse $l \leq 1$ (s- und p-Wellen-Näherung) von Bedeutung, da die Beiträge der höheren Multipole in der Nähe der Pion-Produktionsschwelle verschwinden.

2.2 Kinematik und Multipolzerlegung

Die Pion-Elektroproduktion wird in niedrigster Ordnung durch das Ein-Photon-Austauschdiagramm (Abb. 2.1) beschrieben [Dre92].

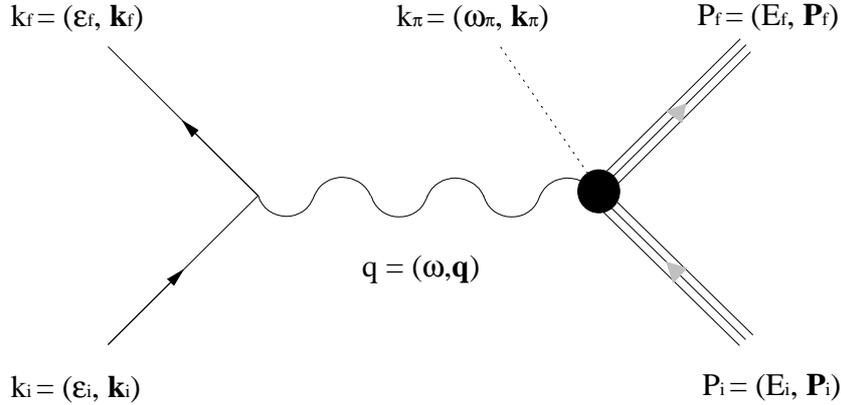


Abb. 2.1: Ein-Photon-Austauschdiagramm für die Pion-Elektroproduktion.

Der Viererimpuls des ausgetauschten Photons $q = (\omega, \mathbf{q})$ wird durch die Viererimpulse des ein- und auslaufenden Elektrons $k_i = (\epsilon_i, \mathbf{k}_i)$ bzw. $k_f = (\epsilon_f, \mathbf{k}_f)$ und das Targetnucleon durch die Vierervektoren $P_i = (E_i, \mathbf{P}_i)$ im Eingangskanal und $P_f = (E_f, \mathbf{P}_f)$ im Ausgangskanal definiert. Für das auslaufende Pion gilt $k_\pi = (\omega_\pi, \mathbf{k}_\pi)$. Es ist üblich, bei Formfaktoren und Strukturfunktionen die positive Größe $Q^2 = -q^2$ zu verwenden.

In der Pion-Elektroproduktion wird unterschieden zwischen der Multipolentwicklung des einlaufenden virtuellen Photonfeldes, dargestellt durch elektrische (EL), magnetische (ML) und coulombartige (CL) Multipole mit dem Drehimpuls L , sowie der Zerlegung des πN -Endzustandes. Auch dort wird in Abhängigkeit vom elektromagnetischen Charakters des Übergangs getrennt zwischen elektrischen ($E_{l\pm}$), magnetischen ($M_{l\pm}$) und longitudinalen ($L_{l\pm}$) Amplituden. Die Multipolamplituden sind Funktionen der Schwerpunktsenergie W , des Impulsübertrages Q^2 sowie des Pionstreuwinkels θ_π^* im Schwerpunktsystem. In s- und p-Wellen-Näherung können die Wirkungsquerschnitte σ_I , $I = L, T, LT, TT, LT'$, als Funktion der Multipole wie folgt dargestellt werden [Be98]:

$$\begin{aligned}
\sigma_T &= |E_{0+} + \cos \theta_\pi^* P_1|^2 + \frac{1}{2} \sin^2 \theta_\pi^* (|P_2|^2 + |P_3|^2) \\
\sigma_L &= |L_{0+} + \cos \theta_\pi^* P_4|^2 + \sin^2 \theta_\pi^* |P_5|^2 \\
\sigma_{LT} &= -\sin \theta_\pi^* \Re [(E_{0+} + \cos \theta_\pi^* P_1) P_5^* + (L_{0+} + \cos \theta_\pi^* P_4) P_2^*] \\
\sigma_{TT} &= \frac{1}{2} \sin^2 \theta_\pi^* (|P_2|^2 - |P_3|^2) \\
\sigma_{LT'} &= \sin \theta_\pi^* \Im [(E_{0+} + \cos \theta_\pi^* P_1) P_5^* + (L_{0+} + \cos \theta_\pi^* P_4) P_2^*] \quad (2.7)
\end{aligned}$$

Dabei sind P_1 bis P_5 zusammengesetzt aus den p-Wellen Multipolen E_{1+} , $M_{1\pm}$ und $L_{1\pm}$:

$$\begin{aligned}
P_1 &= 3E_{1+} + M_{1+} - M_{1-}, P_2 = 3E_{1+} - M_{1+} + M_{1-} \\
P_3 &= 2M_{1+} + M_{1-}, P_4 = 4L_{1+} + L_{1-}, P_5 = L_{1-} - 2L_{1+}
\end{aligned} \tag{2.8}$$

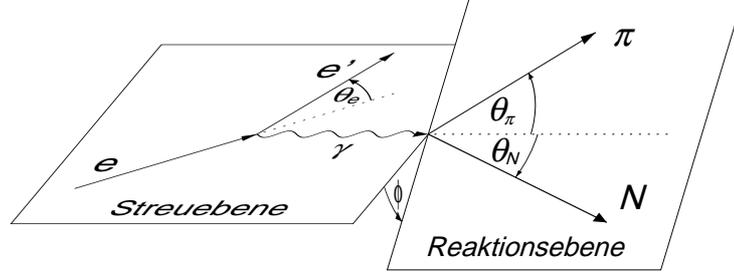


Abb. 2.2: Kinematik der Pion-Elektroproduktion

Abbildung 2.2 zeigt schematisch die Kinematik der untersuchten Reaktion vom Typ $p(e,e'p)\pi^0$. Das Rückstoßproton wird in Spektrometer A und das gestreute Elektron in Spektrometer B detektiert. Der differentielle Wirkungsquerschnitt dieser Reaktion läßt sich schreiben als [Am79]:

$$\begin{aligned}
\frac{d^3\sigma(\theta_\pi^*, \phi_\pi^*)}{d\Omega' dE' d\Omega_\pi^*} &= \Gamma \left(\sigma_T(\theta_\pi^*) + \epsilon_L \sigma_L(\theta_\pi^*) \right. \\
&\quad + \epsilon \sigma_{TT}(\theta_\pi^*) \cos 2\phi_\pi^* \\
&\quad + \sqrt{2\epsilon_L(1+\epsilon)} \sigma_{LT}(\theta_\pi^*) \cos \phi_\pi^* \\
&\quad \left. + h \sqrt{2\epsilon_L(1-\epsilon)} \sigma_{LT'}(\theta_\pi^*) \sin \phi_\pi^* \right)
\end{aligned} \tag{2.9}$$

Dabei ist h die Helizität des Elektrons, Γ der virtuelle Photonenfluß, festgelegt durch den Elektronenarm und ϵ die transversale Polarisation des virtuellen Photons:

$$\Gamma = \frac{\alpha}{2\pi^2} \frac{E'}{E} \frac{k_\gamma}{Q^2} \frac{1}{1-\epsilon} \tag{2.10}$$

$$\epsilon = \left(1 + (2\mathbf{q}^2/Q^2) \tan^2 \frac{\theta_e}{2} \right)^{-1} \tag{2.11}$$

Weiterhin ist α die Feinstrukturkonstante und $k_\gamma = (W^2 - m_p^2)/2m_p$ die sogenannte Photon-Äquivalentenergie. Das Sternchen (*) bezieht sich auf die Koordinaten im πN -Schwerpunktssystem.

Die fünf differentiellen Wirkungsquerschnitte σ_I aus Gleichung 2.9 können durch eine geeignete Wahl der kinematischen Parameter separiert werden. Die Wirkungsquerschnitte σ_T und σ_L lassen sich mittels einer Rosenbluth-Separation trennen, bei der mehrere Messungen mit unterschiedlichen ϵ -Werten durchgeführt werden. Wird wie in dem vorliegenden Experiment nur ein einziger ϵ -Punkt vermessen, so kann nur die Summe $\sigma_0 = \sigma_T + \epsilon_L \sigma_L$ angegeben werden. Durch eine Variation des Azimutwinkels ϕ_π^* werden die Interferenzterme σ_{LT} und σ_{TT} isoliert und $\sigma_{LT'}$ schließlich ist wegen der expliziten $\sin \phi_\pi^*$ -Abhängigkeit nur durch einen Nachweis des gestreuten polarisierten Elektrons außerhalb der Reaktionsebene zu bestimmen. Der experimentelle Zugang zu den differentiellen Wirkungsquerschnitten und die resultierenden Kinematiken für das Experiment zur π^0 -Produktion sind Bestandteil der Betrachtungen in Kapitel 5.

Die Darstellung der Strukturfunktionen in Form von Multipolamplituden wird häufig aufgrund ihrer leichten Anschaulichkeit verwendet. Eine andere nützliche Parametrisierung ist die Zerlegung des hadronischen Stromes \vec{J} in die sphärischen Komponenten $J_{\pm,0}$ mit $J_{\pm} = \mp(J_x \pm iJ_y)/\sqrt{2}$ und $J_0 = J_z$. Die Komponenten des Stromoperators \vec{J} können als 2×2 Matrizen durch die Helizitätsamplituden H_i ausgedrückt werden:

$$J_+ = \begin{pmatrix} H_1 & H_2 \\ H_3 & H_4 \end{pmatrix}, J_- = \begin{pmatrix} H_4 & -H_3 \\ -H_2 & H_1 \end{pmatrix}, J_0 = \begin{pmatrix} H_5 & H_6 \\ H_6 & -H_5 \end{pmatrix} \quad (2.12)$$

Gemäß der Notation in MAID2000 ² [MAI00] gilt für die Amplituden:

$$\begin{aligned} H_1 &= -\frac{1}{\sqrt{2}} \sin \theta_\pi^* \cos \frac{\theta_\pi^*}{2} (F_3 + F_4) \\ H_2 &= \sqrt{2} \cos \frac{\theta_\pi^*}{2} \left[(F_2 - F_1) + \frac{1}{2} (1 - \cos \theta_\pi^*) (F_3 - F_4) \right] \\ H_3 &= \frac{1}{\sqrt{2}} \sin \theta_\pi^* \sin \frac{\theta_\pi^*}{2} (F_3 - F_4) \\ H_4 &= \sqrt{2} \sin \frac{\theta_\pi^*}{2} \left[(F_1 + F_2) + \frac{1}{2} (1 + \cos \theta_\pi^*) (F_3 + F_4) \right] \\ H_5 &= \cos \frac{\theta_\pi^*}{2} (F_5 + F_6) \\ H_6 &= \sin \frac{\theta_\pi^*}{2} (F_6 - F_5) \end{aligned} \quad (2.13)$$

Die CGLN-Amplituden F_1 bis F_6 wiederum können in die bekannten elektro-magnetischen, magnetischen und coulombartigen Multipole als Funktion der Ableitung von Legendre-Polynomen zerlegt werden [MAI00].

²Diese Notation wurde in Kapitel 5.4 zur Berechnung der Koeffizienten für die Multipolamplituden verwendet.

$$\begin{aligned}
F_1 &= \sum_{l \geq 0} \left[(lM_{l+} + E_{l+})P'_{l+1} + ((l+1)M_{l-} + E_{l-})P'_{l-1} \right] \\
F_2 &= \sum_{l \geq 1} [(l+1)M_{l+} + lM_{l-}]P'_l \\
F_3 &= \sum_{l \geq 1} \left[(E_{l+} - M_{l+})P''_{l+1} + (E_{l-} + M_{l-})P''_{l-1} \right] \\
F_4 &= \sum_{l \geq 2} (M_{l+} - E_{l+} - M_{l-} - E_{l-})P''_l \\
F_5 &= \sum_{l \geq 0} \left[(l+1)L_{l+}P'_{l+1} - lL_{l-}P'_{l-1} \right] \\
F_6 &= \sum_{l \geq 1} [lL_{l-} - (l+1)L_{l+}]P'_l
\end{aligned} \tag{2.14}$$

2.3 Streumatrix

Zur Beschreibung von Streuprozessen wird häufig der Streuoperator S verwendet, dessen Matrixelemente die sogenannte Streumatrix bilden. Die Streumatrix verbindet den Eingangszustand eines Systems mit dessen Ausgangszustand nach der Reaktion. Dabei wird angenommen, daß die Teilchen vor und nach dem Stoß unendlich weit voneinander entfernt sind, deren Wellenfunktionen also keine Wechselwirkung untereinander mehr haben. Die Betragsquadrate der Matrixelemente $\langle f|S|i \rangle$ entsprechen den Übergangswahrscheinlichkeiten vom Zustand i in den Zustand f . Zusätzlich ist die Unitarität der Streumatrix ($S^\dagger S = 1$) gleichbedeutend mit Erhaltungsgrößen. Die Unitaritätsbedingung für die S-Matrix entspricht der Aussage, daß die Summe aller Übergangswahrscheinlichkeiten gleich 1 ist.

Im untersuchten Energiebereich bis 40 MeV oberhalb der π^0 -Produktionsschwelle wird die Struktur der s-Wellen-Multipolamplitude durch das Ladungsaustauschdiagramm, dargestellt in Abbildung 2.3, dominiert. Da der Strom eine Erhaltungsgröße ist, bedingt der Schwellenaufbruch für die π^+ -Produktion etwa 6 MeV oberhalb der π^0 -Schwelle eine entsprechende Stärkezunahme im zugehörigen Kanal. Abbildung 2.4 zeigt die Struktur im E_{0+} -Multipol als Funktion der Schwerpunktsenergie.

Experimentell ist der Imaginärteil der Strukturfunktion $R_{LT'}$ durch eine Messung des helizitätsabhängigen Anteils des Wirkungsquerschnitts zugänglich (siehe Abschnitte 5.3 und 5.6).

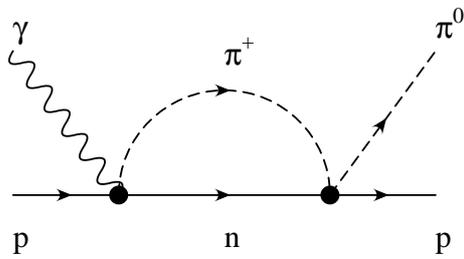


Abb. 2.3: Diagramm für die Ladungsaustauschreaktion $\gamma p \rightarrow \pi^+ n \rightarrow \pi^0 p$.

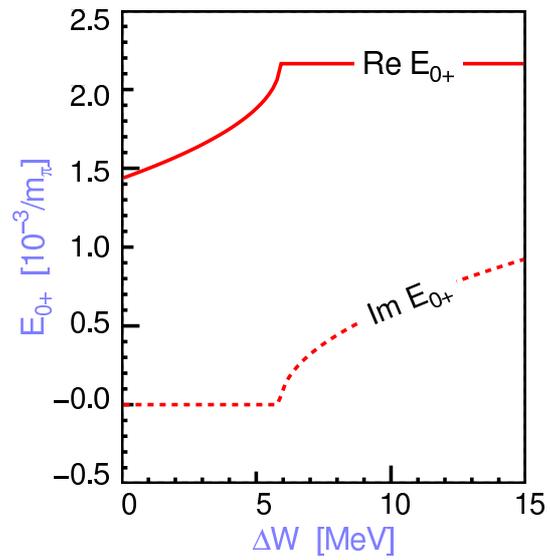


Abb. 2.4: E_{0+} -Multipolamplitude als Funktion der Schwerpunktsenergie

Kapitel 3

Experimenteller Aufbau

Die Experimente zur π^0 -Elektroproduktion wurden an der Drei-Spektrometer-Anlage des Mainzer Elektronenbeschleunigers (MAMI [Mam94]) durchgeführt. Bei dem Dauerstrichbeschleuniger handelt es sich um ein dreistufiges Rennbahn-Mikrotron mit einer derzeit maximalen Energie von 880 MeV. Es liefert Ströme von bis zu $100 \mu\text{A}$ für unpolarisierte Elektronen und $40 \mu\text{A}$ für Elektronen mit einem Polarisationsgrad von 80%. Eine vierte zur Zeit im Aufbau befindliche Stufe wird Energien von 1,5 GeV ermöglichen.

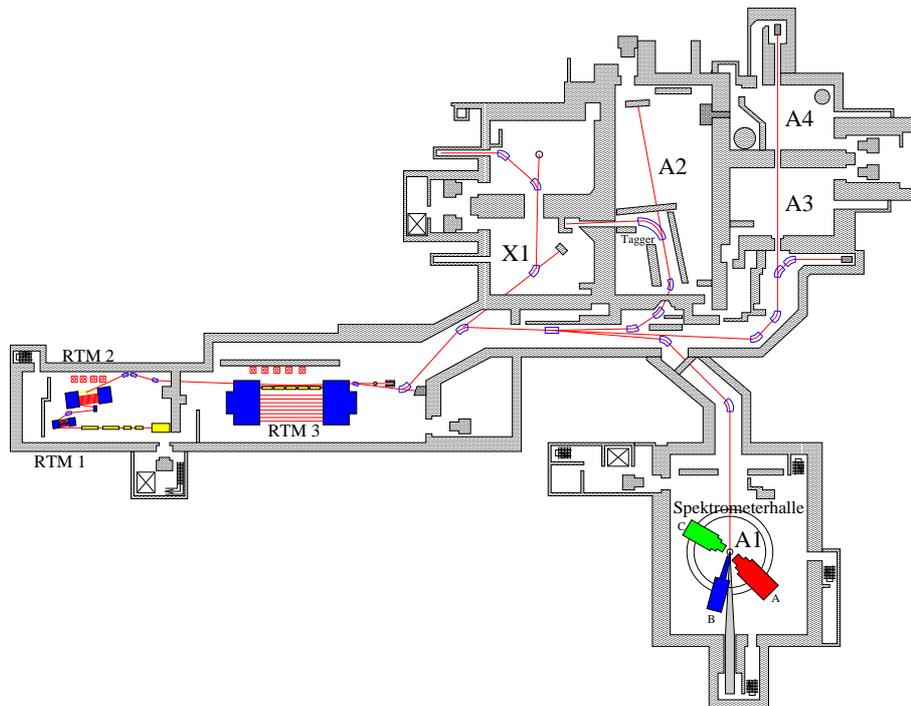


Abb. 3.1: Lageplan des Mainzer Mikrotrons (MAMI B), der Strahlführungssysteme sowie der Experimentierhallen im Herbst 1999. Die neue vierte Beschleunigerstufe wird in der Halle der X1 Kollaboration aufgebaut.

Abbildung 3.1 zeigt den Grundriß des Strahlführungssystems und der Beschleunigerstufen, sowie die Lage der Experimentierhallen. Das Experiment wurde in

der Spektrometerhalle der A1-Kollaboration unter Verwendung eines polarisierten Elektronenstrahls durchgeführt.

Die Drei-Spektrometer-Anlage [Blo98] besteht aus drei hochauflösenden Magnetspektrometern, die um ein gemeinsames Zentrum verfahrbar sind und in Koinzidenz bis zu drei geladene Reaktionsprodukte nachweisen können. Im Zentrum befindet sich das Target für die Streuexperimente, bei dem es sich je nach Bedarf um einen Festkörper, ein gasförmiges oder ein flüssiges Material handelt. Abbildung 3.2 zeigt ein Foto von der Anlage.

3.1 Abbildungseigenschaften

Die Impulsselektion eines geladenen Teilchens erfolgt durch Dipolmagnete, die das Teilchen in dispersiver Richtung ablenken. Aus dem Durchtrittspunkt in der Fokalebene der Magneto-optik läßt sich dann der absolute Impuls des Teilchens bestimmen. Aufgrund des geforderten Impulsaufhebungsvermögens von $\Delta p/p \leq 10^{-4}$ ist für alle drei Spektrometer in dispersiver Richtung eine Punkt-zu-Punkt-Abbildung notwendig.

Spektrometer		A	B	C
Konfiguration		QSDD	D	QSDD
maximaler Impuls	[MeV/c]	735	870	551
minimaler Streuwinkel		18°	7°	18°
Abbildungsverhalten				
-dispersiv	[mrad]	pt → pt	pt → pt	pt → pt
-nicht dispersiv		→ pt	pt → pt	→ pt
Winkelakzeptanz				
-dispersiv	[mrad]	70	70	70
-nicht dispersiv		100	20	100
Impulsauflösung		$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$
Winkelauflösung am Target	[mrad]	< 3	< 3	< 3
Ortsauflösung am Target	[mm]	3	1	3

Tab. 3.1: Optische Eigenschaften der Magnetspektrometer [Kor94]

Obwohl universell einsetzbar, wurde beim Entwurf der Spektrometer eine Vorauswahl für deren primären Verwendungszweck getroffen. Um eine große Akzeptanz für die Polarisation ϵ des virtuellen Photons zu erreichen, muß mit den Spektrometern ein möglichst großer Winkelbereich abgedeckt werden. Die Aufgabe des „Vorwärts“-Spektrometers fällt dabei auf Spektrometer B, das aufgrund seiner schmalen Bauweise einen minimalen Streuwinkel von 7° zuläßt. Zusätzlich erlaubt



Abb. 3.2: Das Foto zeigt die Drei-Spektrometer-Anlage bestehend aus den Spektrometern A (rot), B (blau) und C (grün). Der Elektronenstrahl kommt von rechts und trifft auf das im Drehzentrum befindliche Target.

seine besondere Dipolform¹ eine Punkt-zu-Punkt-Abbildung in nicht-dispersiver Richtung, womit es das bessere Ortsauflösungsvermögen als die beiden anderen Spektrometer hat. Begünstigt durch die kompakte Bauweise kann es zudem bis zu 10° aus der Ebene gekippt werden. Spektrometer A und C verfügen über

¹Bei Spektrometer B handelt es sich um einen Clamshell-Dipol mit gegeneinander geneigten Polschuhflächen.

einen Quadrupol und einen Sextupol, mit denen die Winkelakzeptanz in nicht-dispersiver Richtung bis auf ± 100 mrad vergrößert und Abbildungsfehler korrigiert werden. Sie haben in nicht-dispersiver Richtung ein Parallel-zu-Punkt-Abbildungsverhalten, welches das Winkelaufhebungsvermögen in der Streuebene verbessert. Tabelle 3.1 listet die wichtigsten Eigenschaften der Magnetspektrometer auf.

3.2 Fokalebenendetektoren

In der Fokalebene der Magnetspektrometer werden mehrere Detektorpakete zum Nachweis und der Identifikation der geladenen Reaktionsprodukte eingesetzt. Abbildung 3.3 zeigt eine Fotografie der Detektoren im Abschirmhaus von Spektrometer A.

Driftkammer: Für die Spurrekonstruktion eines Teilchens werden in allen Spektrometern nahezu identisch aufgebaute vertikale Driftkammern (VDC²) eingesetzt [Dis90]. Diese bestehen aus jeweils zwei Doppelpaketen gasgefüllter Drahtkammern, um mit hoher Genauigkeit den Durchtrittsort sowie den Winkel in dispersiver und nicht-dispersiver Richtung zu bestimmen. Jedes der Doppelpakete besteht aus zwei Drahtebenen, die mit x- und s-Ebene bezeichnet werden. Mit Hilfe einer Polynomparametrisierung werden anschließend in der Datenanalyse die gemessenen Koordinaten in der Fokalebene auf die Koordinaten am Reaktionsort abgebildet. Die vier Targetkoordinaten sind: Impuls des Teilchens, Streuwinkel in dispersiver und nicht-dispersiver Richtung sowie die horizontale Ablage gegenüber der Referenzbahn am Kollimator.

In Kapitel 4 wird das Verfahren zur Spurrekonstruktion im Detail beschrieben.

Szintillationszähler: In die Spektrometer sind zwei Lagen Plastikszintillationszähler eingebaut. Die in Teilchenflugrichtung erste Ebene (dE-Ebene) ist 3 mm dick und wird für die Trennung minimalionisierender Teilchen wie z.B. Elektronen und Pionen von Protonen und Deuteronen benutzt. Die zweite Ebene (ToF³-Ebene) ist 10 mm dick und erlaubt die Messung des Durchtrittszeitpunktes relativ zu einem Teilchen in einem anderen Spektrometer. Mit Hilfe der Koinzidenzzeit läßt sich der im Experiment untersuchte Prozeß vom Untergrund durch andere Prozesse trennen.

Beide Ebenen sind in mehrere getrennt auslesbare Segmente unterteilt, um die integrale Zählratenbelastbarkeit⁴ einer Szintillatorebene zu erhöhen. Zudem

²Vertical Drift Chamber

³Time of Flight

⁴Die Zählratenbelastbarkeit eines Segments wird im wesentlichen bestimmt durch die Ratenbelastbarkeit der angeschlossenen Photomultiplier.

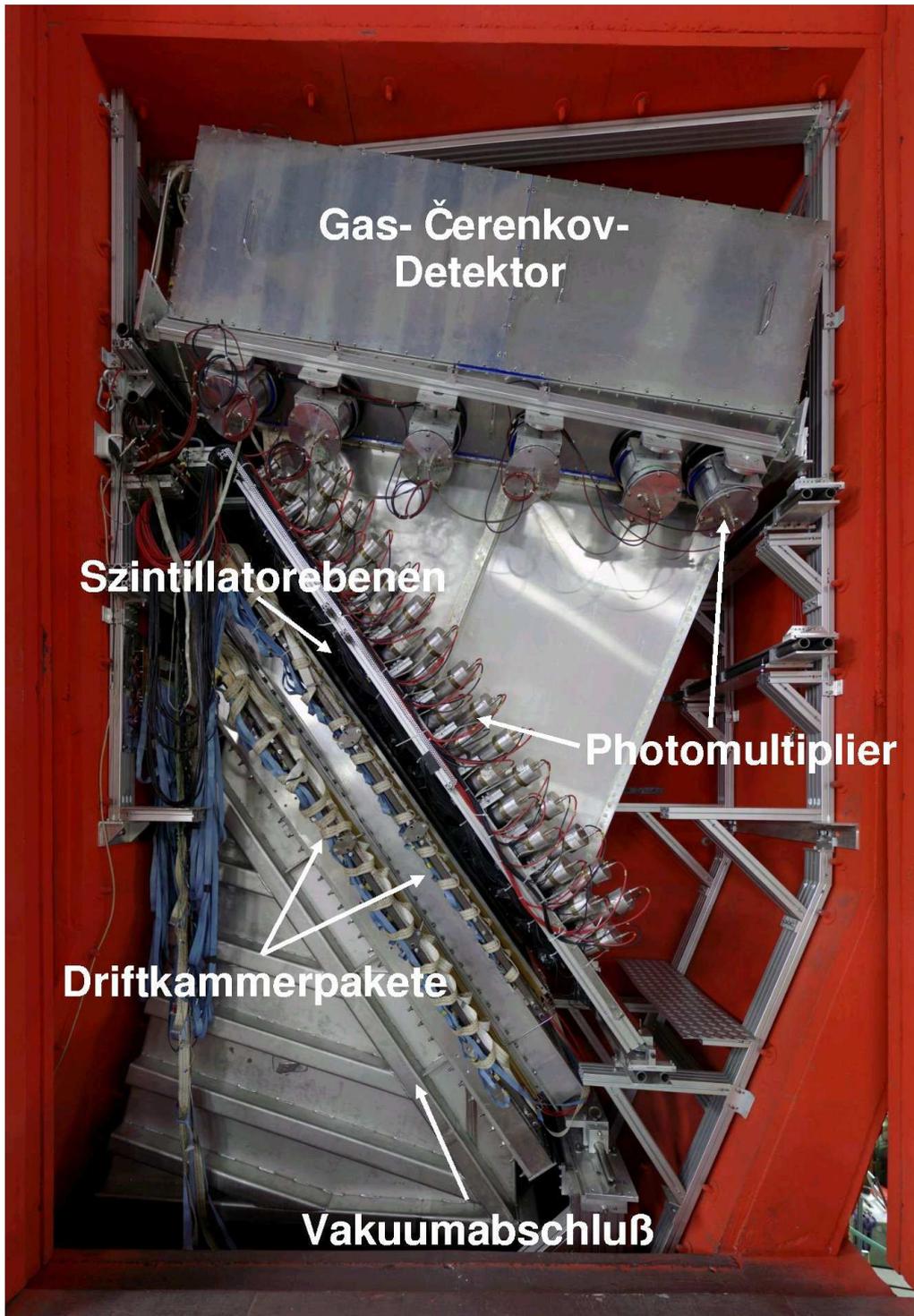


Abb. 3.3: Foto der Detektorsysteme im Abschirmhaus von Spektrometer A.

sind zur Reduktion des Untergrundrauschens an beiden Seiten der Szintillatortsegmente koinzident auslesbare Photomultiplier⁵ angebracht.

Bei dem in dieser Arbeit vorgestellten Prozeß müssen in Spektrometer A sehr niederenergetische Protonen nachgewiesen werden. Ihr spezifischer Energieverlust im Szintillatormaterial ist so hoch, daß eine große Anzahl der Protonen in der dE-Ebene stecken bleibt. Aus diesem Grund wird aus dieser Ebene der Durchtrittszeitpunkt bestimmt und nicht wie üblich aus der ToF-Ebene.

Gas-Čerenkovdetektor: Um Elektronen und Positronen von ebenfalls minimalionisierenden Pionen unterscheiden zu können, verfügen alle drei Spektrometer über einen Gas-Čerenkovdetektor. Der Brechungsindex für das Gas wurde so gewählt, daß Elektronen bzw. Positronen Čerenkovlicht erzeugen, Pionen und schwerere Teilchen dagegen nicht.

- Aus Platzgründen kann in Spektrometer A entweder der Čerenkovdetektor oder das Fokalebene polarimeter eingebaut werden. Da letzteres für das Experiment benötigt wurde, konnten in Spektrometer A minimalionisierende Teilchen mit dem Čerenkovdetektor nicht identifiziert werden.
- Das gestreute Elektron wurde in Spektrometer B nachgewiesen. Dort ist zwar ein Čerenkovdetektor eingebaut, die Akzeptanz des Čerenkovdetektors deckt aber nicht die gesamte Akzeptanz in Spektrometer B ab. Folglich kann er nur als Vetodetektor für Elektronen verwendet werden und ist daher für die Datenanalyse dieses Experimentes von untergeordneter Bedeutung.

Polarimeter: Der Polarisationsgrad des Rückstoßprotons läßt sich mit dem Fokalebene-Polarimeter in Spektrometer A bestimmen. Dieses arbeitet nach dem Prinzip, daß die Spin-Bahn Wechselwirkung des Protons bei der Streuung am Kern in der Größenordnung des Zentralpotentials selbst ist. Durch die Wechselwirkung wird das Proton weiter abgelenkt, als dies bei einem reinen Zentralpotential ohne LS-Kopplung der Fall wäre. Eine Gruppe polarisierter Protonen wird daher eine Asymmetrie in der Streuwinkelverteilung aufweisen, die proportional zur Polarisation der Protonen ist.

Diese Winkelverteilung wird mit einer horizontalen Driftkammer (HDC) vermessen. Als Analysator wird Kohlenstoff verwendet. Das Fokalebene polarimeter wird in [Pos00], [Pos02a] und [Pos02b] im Detail beschrieben.

Im Rahmen dieser Arbeit wurde das Polarimeter zur Bestimmung des Polarisationsgrades des einlaufenden Elektronstrahls benutzt (siehe Kapitel 4.1).

⁵Sekundärelektronenvervielfacher

3.3 Datenerfassung

Jedes der Spektrometer verfügt über einen eigenen Frontendrechner zur Auslese der Rohdaten von den Detektoren. Diese Daten sind über VME-Bus, CAMAC oder Fastbus-Schnittstellen in Form von Analog-Digital-Konvertern (ADC), Zeit-Digital-Konvertern (TDC) oder einfach nur als reines Digitalsignal zugänglich. Zu den ausgelesenen Werten gehören beispielsweise die Signale der Photomultiplier, die Driftzeiten der Elektronen in den VDCs oder die Koinzidenzzeit zwischen zwei Spektrometern.

Alle Daten eines Ereignisses werden zu einem Paket zusammengefaßt und über Ethernet an einen zentralen Rechner geschickt, wo sie der sogenannte „Eventbuilder“ mit den zugehörigen Ereignisdaten der anderen Spektrometer kombiniert. Anschließend werden sie auf Festplatte, CD-ROM oder einem anderen Medium archiviert und für die spätere Datenanalyse zugänglich gemacht.

Bei dem Experiment wurde das Datenerfassungssystem MECDAS⁶ [Kry96] verwendet, welches inzwischen durch das weitaus leistungsfähigere System Aqua [Dis01] ersetzt wurde.

3.4 Target

Im Zentrum der Streuexperimente befindet sich das Target. Sein Ort wird definiert durch den einlaufenden Elektronenstrahl und die gemeinsame Drehachse der drei Spektrometer.

Der Experimentator hat verschiedene Möglichkeiten bezüglich Positionierung und Material des Targets. Festkörpertargets werden in kleine Rähmchen eingeklemmt und an einem höhenverstellbaren Gestell untergebracht. Mittels einer Motorsteuerung kann das jeweils benötigte Rähmchen in den Elektronenstrahl gefahren werden. Für Experimente an Wasserstoff oder Deuterium muß das Targetmaterial verflüssigt werden, um eine möglichst große Dichte und damit hohe Lumineszenz zu erreichen. Das Targetsystem [Ewa96] hierfür besteht aus zwei getrennten Materialkreisläufen (Kühl- und Targetkreislauf), die über einen Wärmetauscher miteinander verbunden sind. Mittels einer speziell konstruierten Targetzelle wird das Targetmaterial in den Elektronenstrahl gebracht. Die Zelle ist 49,5 mm lang, 11,5 mm breit und hat Wände aus einer 10 μm dicken Havarfolie. Um eine lokale Erwärmung der Targetwände und Dichteschwankungen zu verhindern, wird der Elektronenstrahl über die Targetfläche gewedelt. Zusätzlich sorgt ein Ventilator für ein ständiges Umwälzen des Materials im Targetkreislauf. Abbildung 3.4 zeigt eine schematische Zeichnung dieses Kreislaufes.

⁶Mainz Experiment Control and Data Acquisition System

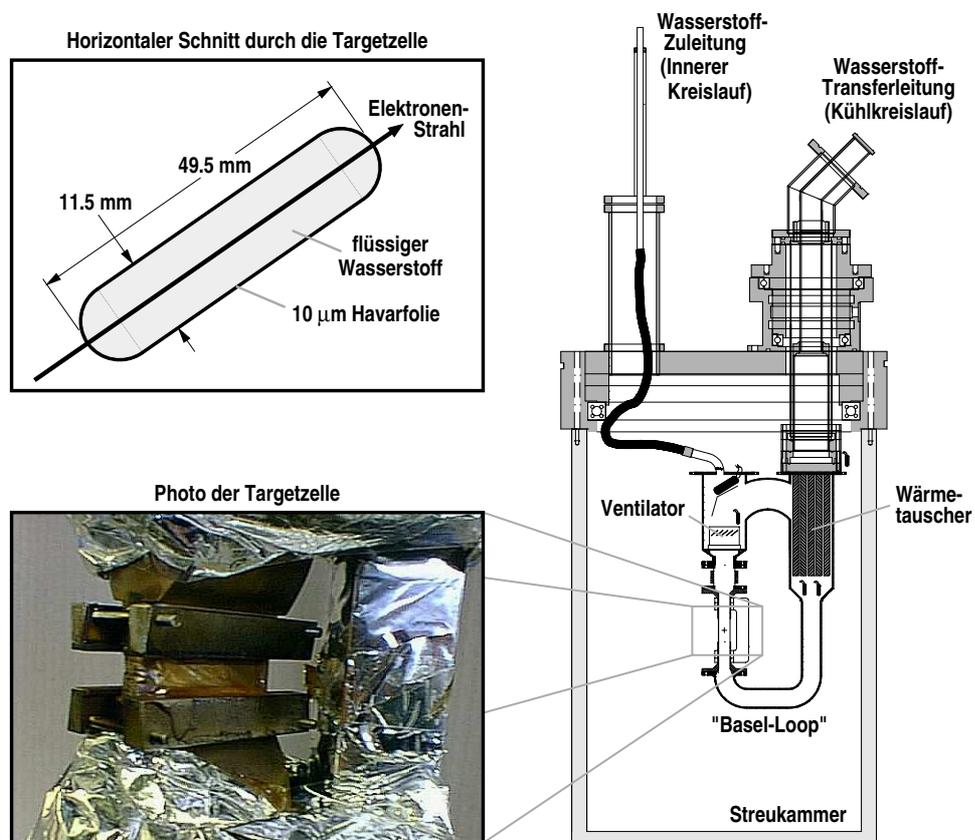


Abb. 3.4: Schematische Darstellung des Targetkreislaufes und Foto der Targetzelle bei der Verwendung von flüssigem Wasserstoff oder Deuterium. Bild aus [Pos00].

Kapitel 4

Kalibrationskinematik

Bei dem in dieser Arbeit vorgestellten Experiment werden für die Bestimmung der Helizitätsasymmetrie $A_{LT'}$ polarisierte Elektronen benötigt. Der Polarisationsgrad des Elektronenstrahls ist folglich eine zwingend erforderliche Meßgröße, welche mit der ersten Kinematik während der Strahlzeit ermittelt werden sollte. Der dabei benutzte Reaktionstyp ist zudem sehr gut geeignet zur Untersuchung systematischer Fehler sowie der Kalibration der Detektorsysteme.

In den Abschnitten dieses Kapitels wird die Kalibrationskinematik im Detail beschrieben und dabei ein besonderes Augenmerk auf die systematischen Unsicherheiten der einzelnen Detektorsysteme gelegt.

4.1 Kinematik

Der Polarisationsübertrag bei der elastischen Streuung eines polarisierten Elektrons an einem freien Proton kann mit Hilfe der Sachs-Formfaktoren genau berechnet werden. Das Fokalebene polarimeter in Spektrometer A erlaubt die Messung der Polarisation des Rückstoßprotons und ermöglicht somit die Bestimmung des Polarisationsgrades des einlaufenden Elektronenstrahls.

In dem Experiment vom Typ $p(\vec{e}, e'\vec{p})$ wird das elastisch gestreute Elektron in Spektrometer B und das Rückstoßproton koinzident in Spektrometer A nachgewiesen. Abbildung 4.1 zeigt eine schematische Darstellung der verwendeten Kinematik¹.

Abgesehen von wenigen Ereignissen, die hauptsächlich auf quasielastische Streuungen in der Havarfolie der Targetzellenwände zurückzuführen sind, ist die Reaktion untergrundfrei. Das sehr gute Ortsauflösungsvermögen von Spektrometer B am Reaktionsort ermöglicht einen Schnitt² auf die Targetwände, wodurch dieser Untergrund nahezu vollständig unterdrückt wird. Abbildung 4.2 zeigt das bereinigte Koinzidenzspektrum für die elastische Streuung von Elektronen an Wasserstoff. Die Halbwertsbreite des Koinzidenzzeitpiks beträgt etwa 620 ps.

¹Es handelt sich um Kinematik „ST4ff“ aus [Pos00], bei der ein Optimum bezüglich der Analysierstärke des Kohlenstoffanalysators im FPP und der Spektrometerakzeptanzen vorliegt.

²In der Datenanalyse werden kinematische Variablen und Meßgrößen durch „Schnitte“ in ihrem Wertebereich eingeschränkt. Bei dem Schnitt auf die Targetwände beispielsweise werden nur Ereignisse aus dem Inneren der Targetzelle akzeptiert.

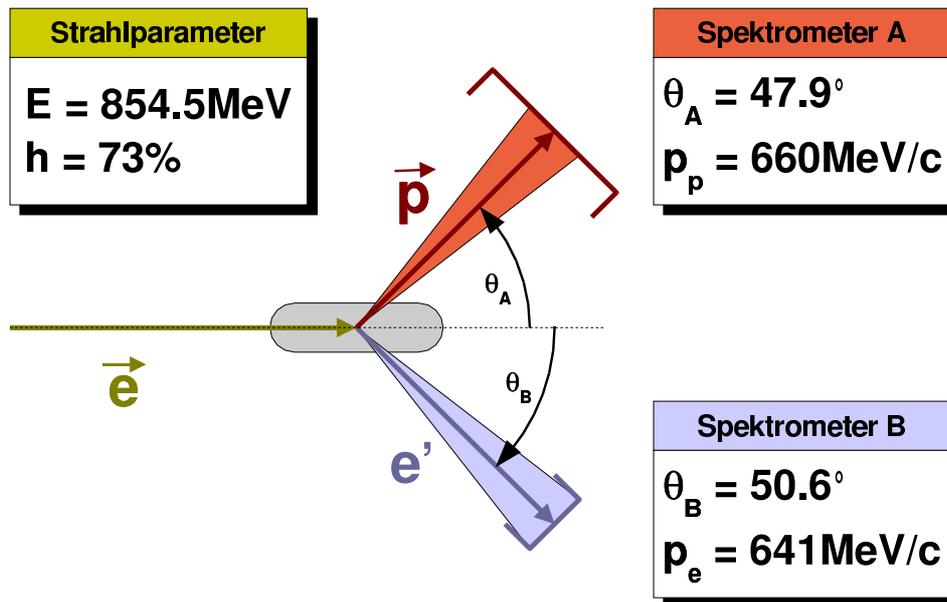


Abb. 4.1: Schematische Darstellung der Kalibrationskinematik. Das Rückstoßproton wird in Spektrometer A und das gestreute Elektron in Spektrometer B nachgewiesen. Der Polarisationsgrad h des einlaufenden Elektronenstrahls betrug im Mittel über alle Kinematiken 73%.

In dem Experiment wurde die Strahlpolarisation zu $(75 \pm 2,7)\%$ bestimmt. Die Meßzeit betrug 7,5 Stunden bei einem mittleren Strom von $1 \mu\text{A}$. Es war geplant, den von der polarisierten Quelle des Beschleunigers verwendeten Kristall für alle Kinematiken zu verwenden, da die Messung des Polarisationsgrades sehr zeitaufwendig ist. Unglücklicherweise hat die Quantenausbeute des Kristalls schon nach wenigen Stunden so stark nachgelassen, daß er gegen einen anderen ausgetauscht werden mußte. Als auch dieser Kristall kurze Zeit später nicht mehr verwendbar war, wurde entschieden, das Experiment mit den aus früheren Messungen bekannten Polarisationsgraden der Kristalle fortzusetzen ohne diese Werte nochmals überprüfen zu können. Für die gesamte Strahlzeit wird daher von einer mittleren Strahlpolarisation von $(73 \pm 5)\%$ ausgegangen.

Desweiteren ist diese Kinematik sehr gut geeignet, die Detektorsysteme zu kalibrieren. Der Grund hierfür ist, daß die elastische Streuung vom Typ $p(e, e'p)$ eine kinematisch überbestimmte Reaktion ist: Aus dem gestreuten Elektron können das Rückstoßproton und umgekehrt mit dem detektierten Proton Impuls und Winkel des Elektrons vollständig rekonstruiert werden. Abbildung 4.3 zeigt die „Missing Energy“ des fertig optimierten Spektrometer B im Vergleich mit den Ergebnissen aus der Monte-Carlo-Simulation *Simul++* [Dis01]. Der Zentralimpuls konnte absolut auf $< 300 \text{ keV}/c$ bestimmt werden, also besser als $\Delta p/p \leq 5 \times 10^{-4}$ (siehe auch Kapitel 5.5).

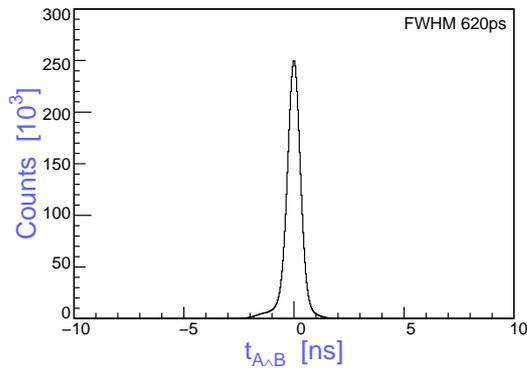


Abb. 4.2: Koinzidenzspektrum für die Reaktion $p(\vec{e}, e' \vec{p})$.

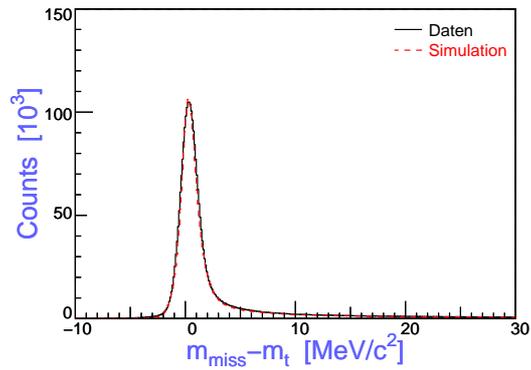


Abb. 4.3: „Missing Energy“ der elastischen $p(\vec{e}, e' \vec{p})$ -Ereignisse.

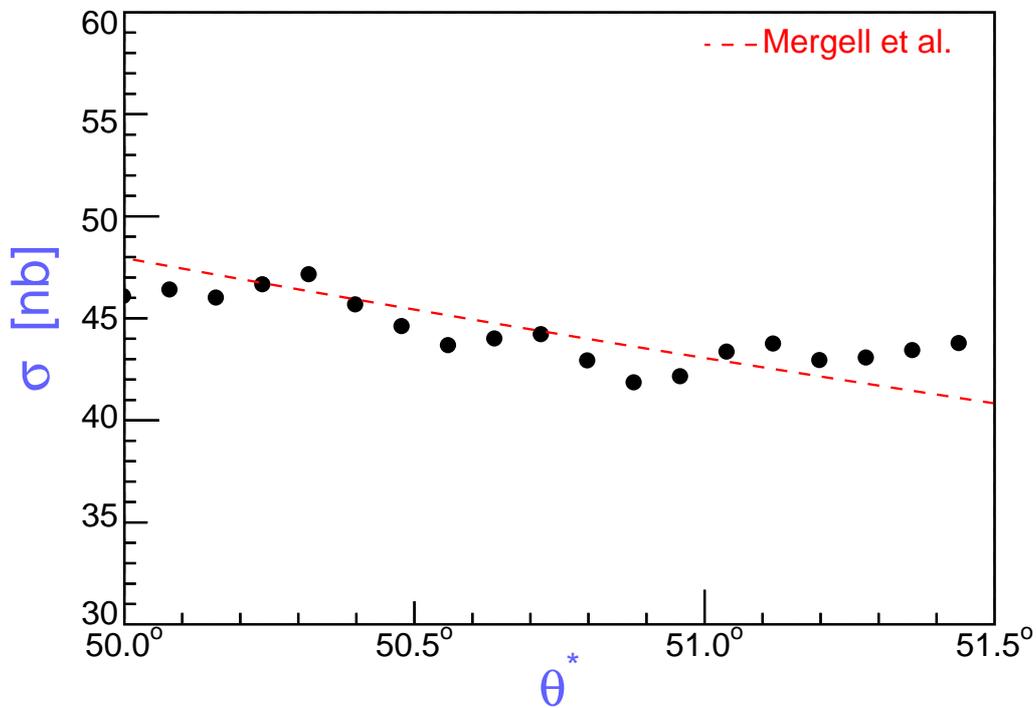


Abb. 4.4: Absoluter Wirkungsquerschnitt für die elastische Streuung von Elektronen am Proton als Funktion des Elektron-Streuwinkels. Eingezeichnet ist die theoretische Vorhersage nach der Parametrisierung von [Me96]. Die statistischen Fehler sind zu klein um dargestellt zu werden.

Der ermittelte elastische Wirkungsquerschnitt ist in Abbildung 4.4 dargestellt. Zum Vergleich ist die theoretische Vorhersage nach der Parametrisierung von [Me96] mit eingetragen. Wie deutlich zu sehen ist, stimmt der gemessene Wirkungsquerschnitt im Mittel sehr gut mit dem Modell überein. Luminositätsbe-

rechnung und globale Detektoreffizienzen sind also wohlverstanden und bekannt. Auffällig ist allerdings ein deutliches streuwinkelabhängiges „Wellenmuster“ im Wirkungsquerschnitt, welches nicht durch die statistischen Fehler erklärt werden kann. Als mögliche Ursachen kommen in Frage:

- lokale Szintillatorineffizienzen
- Fehler im Bahnrückrechnungsverfahren
- Fehler bei der Spurrekonstruktion

In den nun folgenden Abschnitten werden diese Punkte näher untersucht, da der systematische Fehler in dieser Kinematik offensichtlich von einem dieser Effekte dominiert wird. Es ist allerdings zu beachten, daß zur π^0 -Produktion auch andere systematische Fehler beitragen (siehe Kapitel 5) und das hier beschriebene „Wellenmuster“ dort nur eine untergeordnete Bedeutung hat.

4.2 Szintillatorineffizienzen

Beim Durchtritt eines geladenen Teilchens durch das Szintillatormaterial wird ein kleiner Lichtblitz erzeugt, dessen Signal von den Photomultipliern verstärkt und anschließend über einen Diskriminator auf die Logikeinheit gegeben wird (siehe Kapitel 3). War der ursprüngliche Lichtblitz zu klein und wurde vom Diskriminator verworfen oder trat das Teilchen an den Stoßstellen zwischen zwei Szintillatorsegmenten durch die dE- oder ToF-Ebene, so wird kein Triggersignal erzeugt und das Teilchen nicht nachgewiesen. In diesem Fall spricht man von Szintillatorineffizienzen.

Ineffizienzen in einem Fokalebenendetektor können zu „Löchern“ in den Spektren rekonstruierter Targetkoordinaten führen, die sich wie im vorliegenden Fall in einem wellenartigen Verlauf äußern würden. Da das Proton in Spektrometer A koinzident mit dem Elektron in Spektrometer B nachgewiesen wurde, kann eine Ineffizienz des einen Spektrometers auf das andere projiziert werden. Es ist also zu klären, welches der beiden Spektrometer für die „Löcher“ verantwortlich ist.

4.2.1 Spektrometer A

Wird das Proton aufgrund einer Ineffizienz des Szintillators nicht von Spektrometer A nachgewiesen, so kann in der Regel davon ausgegangen werden, daß zumindest das zugehörige Elektron von Spektrometer B gesehen wird. Ist dies der Fall, dann müßte die Summe der Streuwinkelverteilungen aus den koinzidenten Ereignissen und der „Single“-Ereignisse aus Spektrometer B ein wellenfreies Spektrum ergeben.

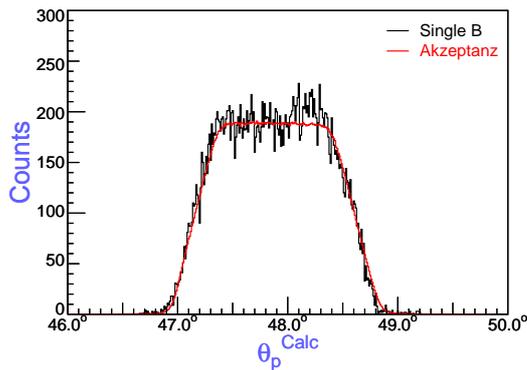


Abb. 4.5: Winkelverteilung des aus den Singles in Spektrometer B rekonstruierten Protons. Eingezeichnet ist die simulierte Verteilung in Spektrometer A.

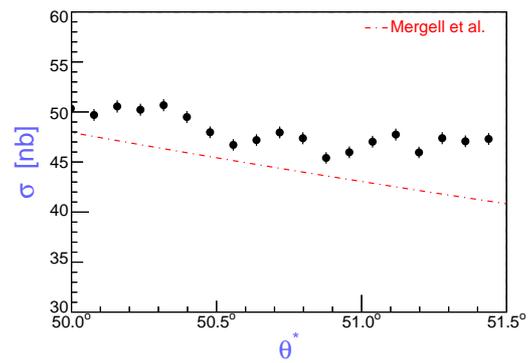


Abb. 4.6: Wirkungsquerschnitt gebildet aus den Koinzidenzen von Spektrometer A und B sowie den Ereignissen nur von Spektrometer B.

Abbildung 4.5 zeigt die Winkelverteilung des aus dem Elektronarm rekonstruierten Protons. Zu erkennen ist ein nahezu untergrundfreies Spektrum, obwohl es sich nur um Singles aus Spektrometer B handelt. Zur Sicherheit wurden zusätzlich Schnitte auf den Kollimator in Spektrometer A sowie auf den Impuls des Protons gemacht. Der aufsummierte Wirkungsquerschnitt ist in Abbildung 4.6 zu sehen. Die wellenartige Struktur ist nach wie vor deutlich vorhanden, woraus geschlossen werden kann, daß der Szintillator in Spektrometer A keine relevanten lokalen Ineffizienzen aufweist. Die Luminositäts- und Totzeitmessung für den Singlearm ist nicht ganz korrekt, was am zu hohen gemessenen Wirkungsquerschnitt zu sehen ist. Da für das Experiment nur die Totzeitmessung für die koinzidenten Ereignisse von Bedeutung ist, wird auf diese Unstimmigkeit nicht weiter eingegangen.

4.2.2 Spektrometer B

Eine Analyse wie im Fall von Spektrometer A ist für Spektrometer B nicht möglich, weil zum einen die Singles in Spektrometer A durch eine entsprechende Skalierung stärker unterdrückt werden als beim Elektronenarm und zum anderen die Rekonstruktion des Reaktionsorts zu schlecht ist um einen sauberen Schnitt auf die Targetwände machen zu können. Ein weiterer Grund für die kleinere Anzahl Singles in A ist der Umstand, daß der Szintillator in B effizienter ist als sein Gegenstück in Spektrometer A, weil sowohl die Triggerelektronik³ als auch die Hochspannungen der Photomultiplier für Spektrometer A auf niederenergetische Protonen eingestellt war. Bei den übrigen Kinematiken sind die Protonen so langsam, daß sie teilweise im Szintillatormaterial stecken bleiben und dementsprechend ein sehr großes Signal liefern, was auf jeden Fall detektiert wird.

³Während des Experimentes wurde versäumt, den ADC-Messbereich auf Elektronen einzustellen.

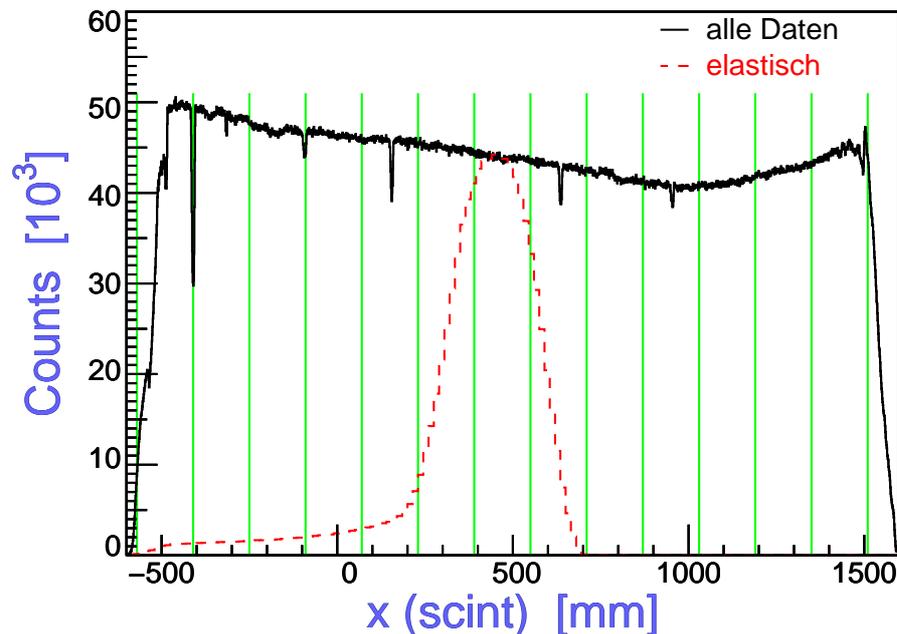


Abb. 4.7: Durchtrittspunkte der Elektronen in der Szintillatorebene von Spektrometer B. Eingetragen ist die Summe aller Teilchenbahnen aller Kinematiken während der Strahlzeit sowie die x-Koordinaten nur aus der Kalibrationskinematik. Die vertikalen Linien entsprechen den Kanten der Paddles der ToF-Ebene; die dE-Paddles sind jeweils um eine halbe Segmentbreite versetzt. Die Daten sind nicht auf die Spektrometerakzeptanzen normiert, d.h. die absolute Höhe pro Bin hat keine Aussagekraft. Die scharfen Einschnitte in regelmäßigen Abständen sind auf die Stoßstellen zwischen den Szintillatorsegmenten zurückzuführen.

Die Stoßstellen zwischen den Szintillatorsegmenten werden daher direkt vermessen, indem anhand der Fokalebenekoordinaten der zugehörige Durchtrittspunkt in der Szintillatorebene berechnet wird. Abbildung 4.7 zeigt die Verteilung der dispersiven Koordinate im Szintillator.

Die Ineffizienzen an den Stoßstellen zwischen den Szintillatorsegmenten sind deutlich erkennbar. Die akzeptierten Ereignisse der elastischen Messung liegen jedoch genau zwischen zwei ineffizienten Stellen, so daß auch für Spektrometer B der Szintillator als Ursache für das Wellenmuster im Wirkungsquerschnitt ausgeschlossen werden kann.

4.3 Bahnrückrechnungsverfahren

Die Driftkammern bestimmen den Durchtrittsort und -winkel eines geladenen Teilchens in der Fokalebene. Mit Hilfe eines Bahnrückrechnungsverfahrens wer-

den sie auf die für die Datenanalyse relevanten Koordinaten am Reaktionsort abgebildet:

$$\begin{pmatrix} x \\ \theta \\ y \\ \phi \end{pmatrix} \mapsto \begin{pmatrix} \Delta p \\ \theta_0 \\ y_0 \\ \phi_0 \end{pmatrix} \quad (4.1)$$

Dabei sind x und θ Ort und Winkel in dispersiver Richtung, y und ϕ in nicht-dispersiver Richtung in der Fokalebene, Δp der Impuls des Teilchens relativ zum Referenzimpuls des Spektrometers, y_0 die projizierte Targetkoordinate in nicht-dispersiver Richtung und θ_0 bzw. ϕ_0 die Winkel in dispersiver und nicht-dispersiver Richtung am Reaktionsort.

Jede der vier Vertexkoordinaten wird aus einem Polynom in den vier Fokalebenekoordinaten parametrisiert:

$$\Delta p = \sum_i c_{1i} x^{m_i} \theta^{n_i} y^{o_i} \phi^{p_i} \quad (4.2)$$

$$\theta_0 = \sum_j c_{2j} x^{m_j} \theta^{n_j} y^{o_j} \phi^{p_j} \quad (4.3)$$

$$y_0 = \sum_k c_{3k} x^{m_k} \theta^{n_k} y^{o_k} \phi^{p_k} \quad (4.4)$$

$$\phi_0 = \sum_l c_{4l} x^{m_l} \theta^{n_l} y^{o_l} \phi^{p_l} \quad (4.5)$$

Welche der Potenzen $m_{i,j,k,l} \cdot p_{i,j,k,l}$ benötigt werden, hängt von der Geometrie des magnetoptischen Systems ab. Beispielsweise sind die Koordinaten in dispersiver und nicht-dispersiver Richtung im Idealfall voneinander entkoppelt und daher entfallen auch die entsprechenden Mischterme. Die Koeffizienten $c_{1i}, c_{2j}, c_{3k}, c_{4l}$ werden in einem Eichverfahren bestimmt, bei dem mittels eines Lochkollimators gezielt Teilchenbahnen herauspräpariert und mit den Gleichungen aus 4.2 bis 4.5 beschrieben werden.

Bei dem Verfahren werden Polynome hoher Ordnung an relativ wenige Stützstellen, definiert durch den Lochkollimator, angepaßt. Es ist folglich nicht auszuschließen, daß die Anpassung zwischen den Stützstellen oder an den Rändern zu unerwünschten Effekten führt. Abbildung 4.8 zeigt die ϕ_0 -Koordinate aufgetragen gegen die θ_0 -Koordinate. Zur Orientierung ist das gleiche Histogramm, erstellt bei einer Eichmessung mit dem Lochkollimator, eingezeichnet. Deutlich erkennbar ist die unregelmäßige Verteilung der Ereignisse in dem zweidimensionalen Histogramm.

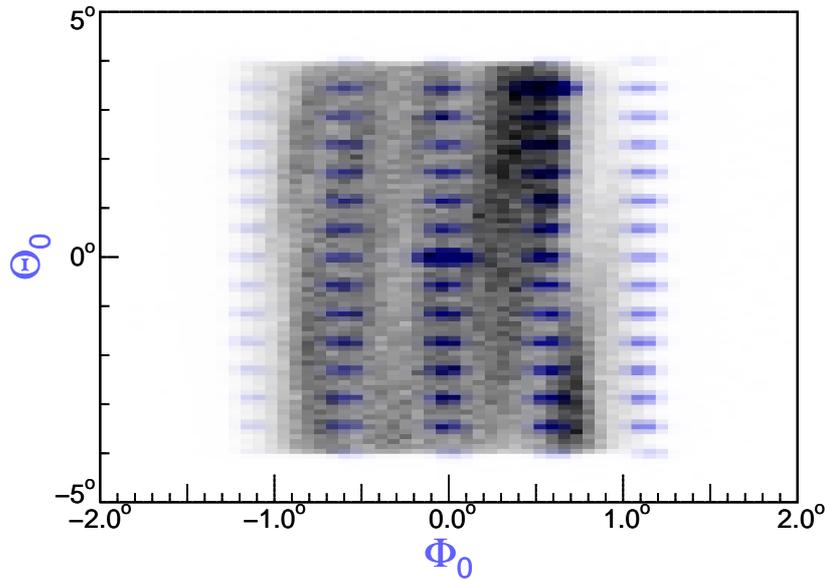


Abb. 4.8: Targetkoordinate ϕ_0 aufgetragen gegen θ_0 . Um die ungleichmäßige Verteilung der Ereignisse hervorzuheben, wurde der Wert jedes Bins zur 4ten Potenz eingezeichnet. Gleichzeitig in blau dargestellt ist das Ergebnis einer Messung mit dem Lochkollimator im Sommer 2000. Es gilt zu beachten, daß die Verteilung der Ereignisse in der Nähe der „Löcher“ ähnliche Grauwerte hat – ein mögliches Indiz für ein Polynom, das den tatsächlichen funktionellen Verlauf nur an den Stützstellen korrekt widerspiegelt.

Um festzustellen, ob der polynomiale Ansatz beim Bahnrückrechnungsverfahren für den wellenförmigen Verlauf des Wirkungsquerschnitts verantwortlich ist, wurden in der Fokalebene isotrop verteilte Koordinaten (x, θ, y, ϕ) gewürfelt. Anschließend wurden diese Koordinaten auf den Targetort zurückgerechnet und dort die gleichen Histogramme wie bei den echten Ereignissen betrachtet. Abbildung 4.9 zeigt das Ergebnis dieser Rechnung für die ϕ_0 -Koordinate aufgetragen gegen die θ_0 -Koordinate. Im Gegensatz zu Abbildung 4.8 sind keine Strukturen oder vergleichbare Wellenmuster erkennbar. Der Algorithmus zur Bahnrückrechnung scheint also nicht für die untersuchten lokalen⁴ systematischen Fehler verantwortlich zu sein.

⁴Einige Spektren zeigen unzulässige „globale“ Abhängigkeiten von eigentlich unabhängigen Targetkoordinaten, wie z.B. ϕ_0 und z_{Target} , die auf falsche Koeffizienten $c_{1i}, c_{2j}, c_{3k}, c_{4l}$ zurückzuführen sind.

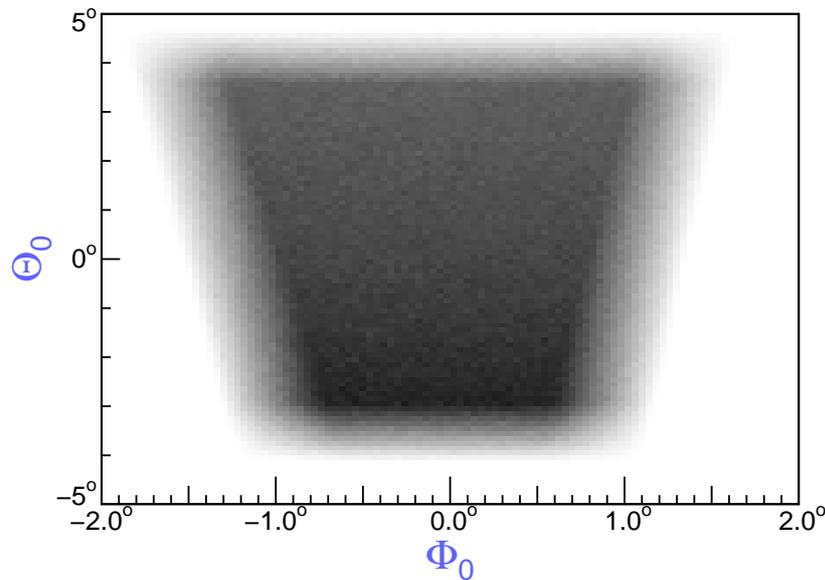


Abb. 4.9: Simulierte Targetkoordinate ϕ_0 aufgetragen gegen θ_0 . Die Koordinaten wurden aus isotrop gewürfelten Fokalebenekoordinaten erzeugt, ohne das Magnetfeld zu berücksichtigen. Die trapezförmige Verteilung ist somit nur ein Softwareartefakt.

4.4 Spurrekonstruktion

Im Juni 1999 wurde das in dieser Arbeit beschriebene Experiment zur π_0 -Elektroproduktion das erste Mal durchgeführt. Wie sich im Nachhinein herausstellte, wurde die vertikale Driftkammer in Spektrometer B nach einer vorausgehenden Reparatur nicht korrekt zusammengesetzt und funktionierte daher so schlecht, daß die Daten nicht auswertbar waren. Im November und Dezember 1999 wurde die Strahlzeit nach einer erneuten Reparatur wiederholt. Die Ergebnisse dieser Arbeit basieren auf den Daten der zweiten Strahlzeit.

Die Fehlfunktion zeigte sich überdeutlich in einem wellenförmigen Verlauf des elastischen Wirkungsquerschnitts. Abbildung 4.10 zeigt die Winkelverteilung des gestreuten Elektrons für die Strahlzeiten im Sommer sowie im Winter 1999.

Ursache für die falsche Rekonstruktion des Elektronstreuwinkels bei der ersten Strahlzeit waren unzureichend gespannte Driftkammerdrähte. Da bei der zweiten Strahlzeit offensichtlich vergleichbare Strukturen im selben Spektrum auftauchen (siehe Abbildung 4.10), liegt die Vermutung nahe, daß die Reparatur nicht ganz erfolgreich war. Ausgehend von einer kurzen Beschreibung der Spurrekonstruktion soll das Problem im folgenden näher untersucht werden.

Beim Durchtritt eines geladenen Teilchens durch die Driftkammer werden Elektron-Ion-Paare erzeugt, die sich zu den Anodendrähten bzw. zur Kathodenfolie bewegen. Gemessen wird die Zeit zwischen dem Impuls an einem Anodendraht

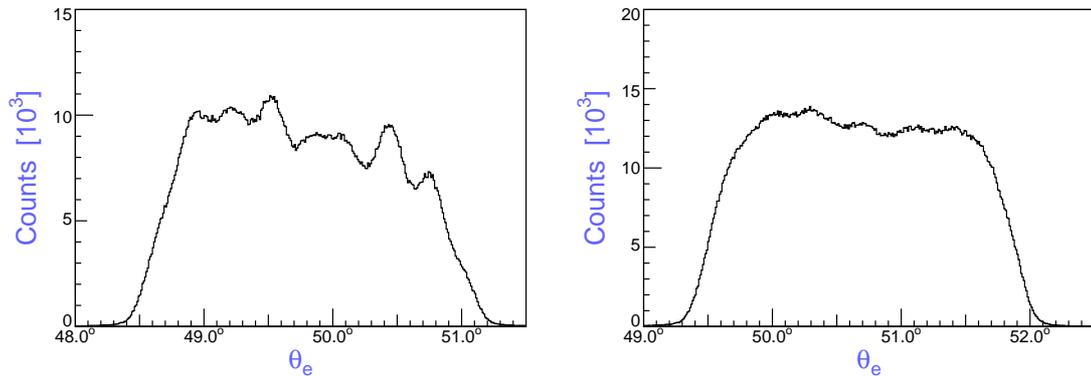
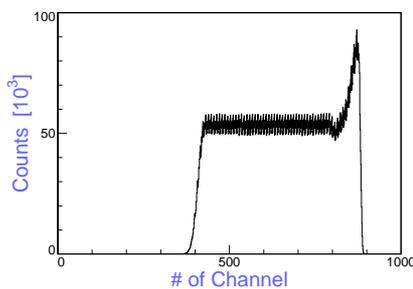


Abb. 4.10: Winkelverteilung des Elektronstreuwinkels für die Strahlzeiten im Juni 1999 (links) sowie nach der Reparatur im November und Dezember 1999 (rechts). Das rechte Spektrum enthält bereits die beschriebenen Korrekturen.

(Start) und dem Signal an den Szintillationszählern (Stop). Das linke Histogramm in Abbildung 4.11 zeigt ein typisches Driftzeitspektrum. Kleine Kanalnummern entsprechen großen Zeiten und umgekehrt große Kanalnummern kurzen Zeiten. Die gemessenen Driftzeiten sind immer positiv, da nicht entschieden werden kann, ob das Signal von oberhalb oder unterhalb des Drahtes kommt.



```

abcn.b.det.vdc.x1[122]: 479
abcn.b.det.vdc.x1[123]: 771
abcn.b.det.vdc.x1[124]: 840
abcn.b.det.vdc.x1[125]: 823
abcn.b.det.vdc.x1[126]: 375

```

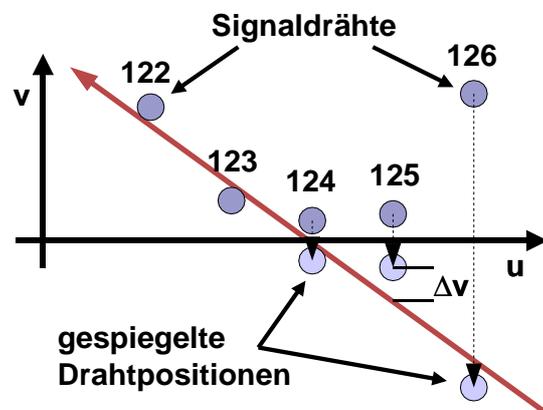


Abb. 4.11: Links oben: Typisches Driftzeitspektrum; Links unten: exemplarische Rohzeiten für einen Teilchendurchtritt; Rechts: Mit Hilfe einer χ^2 -Minimierung wird eine Gerade an die (u,v) -Koordinatenpaare angepaßt. Das beste χ^2 legt das Vorzeichen für die v -Koordinate fest. Signaldraht 126 beispielsweise hat ein gespiegeltes Vorzeichen.

Pro Teilchen feuern im Mittel vier bis fünf Drähte. Abbildung 4.11 zeigt exemplarisch die Rohdaten eines Durchtritts durch die X1-Ebene von Spektrometer B. Zusammen mit der bekannten Driftgeschwindigkeit der Elektronen im Gas wird

aus den Driftzeiten der Abstand zum jeweiligen Draht bestimmt. Mit Hilfe einer χ^2 -Minimierung an die Teilchenbahn kann anschließend ermittelt werden, auf welcher Seite des Drahtes es vorbeigeflogen ist (siehe rechte Zeichnung in 4.11). In Kombination mit der Spur aus den anderen drei Ebenen einer Driftkammer werden anschließend die Fokalebenekoordinaten x , θ , y und ϕ berechnet.

Der Algorithmus geht davon aus, daß keiner der Driftkammerdrähte einen vertikalen Offset besitzt. Befindet sich ein Draht an einer anderen Position als angenommen, so wird eine falsche Driftstrecke ermittelt und die rekonstruierte Teilchenbahn ist nicht korrekt. Nicht ausreichend gespannte Driftkammerdrähte beispielsweise führen zu einem variablen vertikalen Offset entlang des Drahtes und damit zu falschen Fokalebenekoordinaten.

Ziel der Untersuchung ist nun, die tatsächliche Lage eines durchhängenden Drahtes zu bestimmen, um sie anschließend bei der Spurrekonstruktion berücksichtigen zu können. Angenommen, von fünf gefeuerten Drähten hat ein Draht eine falsche Position. Die rekonstruierte Spur wird dann ein relativ großes Δv für den „schlechten“ und ein kleines Δv für die „guten“ Drähte aufweisen. Gemittelt über viele Teilchendurchtritte ist $\Delta \bar{v}_i$ ein Maß für die Abweichung des i -ten Drahtes von der vermuteten Drahtposition v_i .

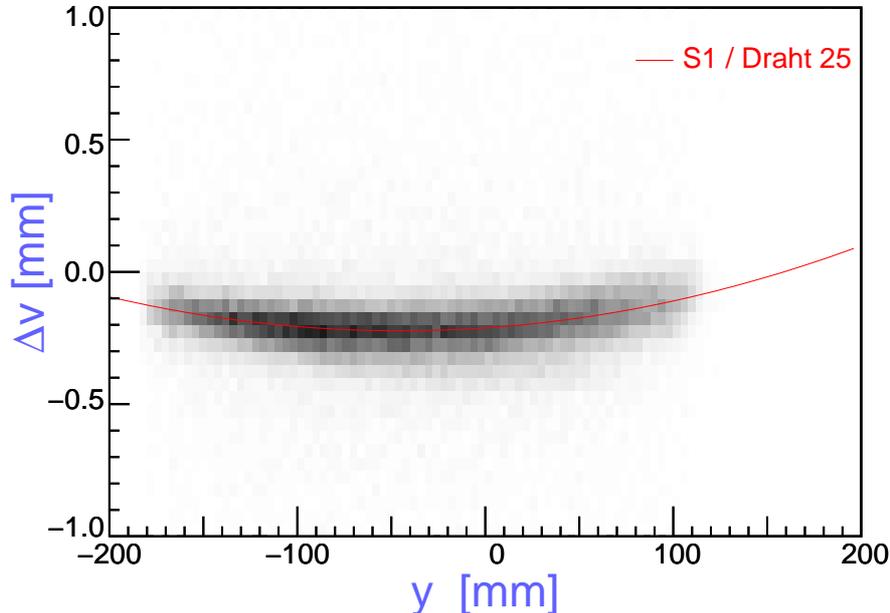


Abb. 4.12: Abweichung eines Drahtes der s1-Ebene von der vermuteten Position. Der Draht weist eine deutliche Krümmung entlang der y -Fokalebenekoordinaten auf. In Rot eingezeichnet ist die χ^2 -Anpassung eines Polynoms 2ter Ordnung an die Krümmung, welcher zur Korrektur der Spurrekonstruktion verwendet wird.

Abbildung 4.12 zeigt die Abweichung Δv für einen Draht aus der s1-Ebene in Abhängigkeit von der Fokalebenekoordinate y . Daran angepaßt wird ein Polynom 2ter Ordnung, welches die offensichtliche Krümmung des Signaldrahtes beschreibt. Bei der Spurrekonstruktion wird nun in einem ersten Durchlauf die ungefähre y -Koordinate mit der ursprünglichen Drahtposition bestimmt sowie die Abweichung $\Delta v(y)$ berechnet und in einem zweiten Durchlauf die korrigierten Fokalebenekoordinaten ermittelt.

Auch wenn dieser modifizierte Algorithmus zur Spurrekonstruktion die Fehler in den Fokalebenekoordinaten deutlich verringert⁵, so ist es damit nicht möglich, den wellenförmigen Verlauf im elastischen Wirkungsquerschnitt vollständig zu beheben. Das Verfahren zur Bestimmung der Abweichung Δv ist lediglich sensitiv auf einzelne falsch positionierte Drähte. Wenn mehrere benachbarte Signaldrähte durchhängen, wird die Abweichung Δv klein sein und trotzdem eine ungenaue Spur rekonstruiert.

4.5 Fazit

Die Spurrekonstruktion in den vertikalen Driftkammern ist die wahrscheinlichste Ursache für die wellenförmigen Strukturen im elastischen Wirkungsquerschnitt. Betrachtet man die ϕ_0 -Winkelverteilungen in den Histogrammen aus Abbildung 4.10, so kommt man leicht zur Schlußfolgerung, daß die Driftkammerdrähte lediglich noch etwas mehr gespannt werden müssen. Die ϕ_0 -Koordinate wird in erster Näherung mit Hilfe der s1- bzw. s2-Ebene, also den schräg gespannten Driftkammerdrähten, bestimmt. Die Spannvorrichtung beeinflußt jedoch immer auch die horizontal gespannten Drähte weshalb zu befürchten ist, daß beim erneuten Spannen der s1/s2-Drähte die Drähte in der x1/x2-Ebene reißen. Eine weitere Reparatur in dieser Form ist also nicht zu empfehlen.

Bei elastischen Streuungen vom Typ $p(e, e'p)$ wird nur ein sehr kleiner Ausschnitt der Fokalebene ausgeleuchtet. Folglich werden lokale Fehlstellen in den Detektoren stärker hervorgehoben, als dies bei einer gleichmäßigen Ausnutzung der Detektorakzeptanz der Fall ist. Da bei den weiteren in dieser Arbeit untersuchten Reaktionen vom Typ $p(e, e'p)\pi^0$ die volle Fokalebeneakzeptanz ausgenutzt wird, kann davon ausgegangen werden, daß die lokalen Fehlstellen im Mittel nur eine untergeordnete Rolle spielen.

Die Untersuchung der wellenförmigen Strukturen im elastischen Wirkungsquerschnitt hat insgesamt gezeigt, daß die Detektorsysteme von Spektrometer A und B keine global signifikanten Ineffizienzen aufweisen.

⁵Der wahrscheinlichste Fehler in der x-Koordinate wird beispielsweise von $80\mu m$ auf $70\mu m$ reduziert.

Kapitel 5

Neutrale Pion-Elektroproduktion

Nachdem in den vorangegangenen Kapiteln die formale Parametrisierung und der apparative Teil des Experimentes zur π^0 -Elektroproduktion eingeführt wurde, wird nun das eigentliche Experiment beschrieben. Ausgehend von einer Vorstellung der verschiedenen Kinematiken werden in den Abschnitten dieses Kapitels die Methoden zur Ereignisidentifikation und Datenanalyse beschrieben, weitere systematische Fehlerquellen aufgezeigt und schließlich die Ergebnisse präsentiert.

5.1 Kinematiken

Das Experiment ist in sieben Kinematiken unterteilt, die wiederum in zwei Gruppen einzuordnen sind. Die erste Gruppe besteht aus sechs Einstellungen, die zusammengenommen den Energiebereich bis $\Delta W = 40\text{MeV}$ über der Pion-Produktionsschwelle abdecken bei einer gleichzeitig „gestückelten“ ϕ_π^* -Akzeptanz und eine einzelne Kinematik, bei der aufgrund des Lorentzboosts eine vollständige ϕ_π^* -Akzeptanz bis $\Delta W = 4\text{MeV}$ gegeben ist. Die Abbildung 5.1 zeigt die Akzeptanzbereiche der Kinematiken als Funktion von ΔW . Die kinematischen Parameter des Experiments sind in Tabelle 5.1 zusammengefasst.

Kinematik	p_p (MeV/c)	p_e (MeV/c)	θ_B	θ_A	α_{OOP}	ϕ_π^*	Meßzeit (Stunden)
0-6	259	662	17,2°	22,1°	0°	180°	15,5
0-7	241	662	16,9°	34,4°	0°	180°	15,5
0-8	243	662	16,9°	52,4°	0°	0°	16
0-9	263	662	17,2°	58,1°	0°	0°	15,5
4-1	263	652	15,4°	40,1°	7.65°	90°	41
4-2	243	652	16,5°	43,4°	3.96°	90°	215
4-3	230	652	16,8°	44,6°	0°	90°	55

Tab. 5.1: Kinematiken zur Pion-Elektroproduktion und die zugehörigen Parameter. Die unterschiedlichen Farben repräsentieren die Kinematiken aus Abbildung 5.1.

5.2 Ereignisidentifikation

Zur Identifikation der $p(e,e'p)\pi^0$ -Ereignisse werden eine Koinzidenz zwischen dem gestreuten Elektron und dem Rückstoßproton verlangt und gleichzeitig Anzeichen

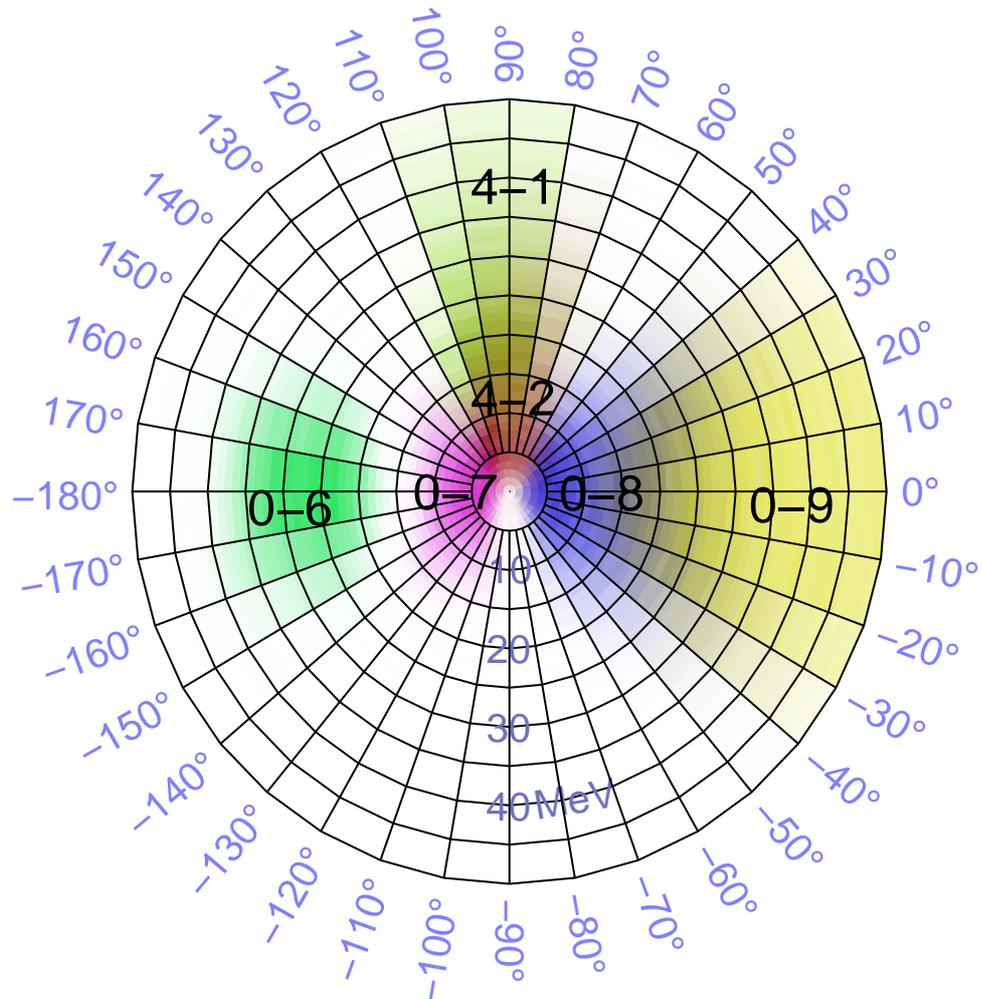


Abb. 5.1: Simulierte ϕ_π^* -Akzeptanz als Funktion der Schwerpunktsenergie ΔW , aufgetragen in einem zweidimensionalen Koordinatensystem. Der Polarwinkel entspricht der ϕ_π^* -Koordinate und der Abstand vom Ursprung der Energie ΔW bis 50 MeV. Die verschiedenen Kinematiken sind farblich voneinander getrennt eingezeichnet und können anhand der Tabelle 5.1 identifiziert werden. Der Übersichtlichkeit wegen ist die Kinematik 4-3 nicht mit eingetragen; sie deckt den zentralen Energiebereich bis $\Delta W = 4$ MeV ab.

für ein neutrales Pion in der Energie- und Massenbilanz gefordert. Da in dem untersuchten kinematischen Bereich Teilchen aus unerwünschten Nebenreaktionen in den Spektrometern nachgewiesen werden, ist zusätzlich für die Selektion der „echten“ Ereignisse eine Untergrundseparation erforderlich.

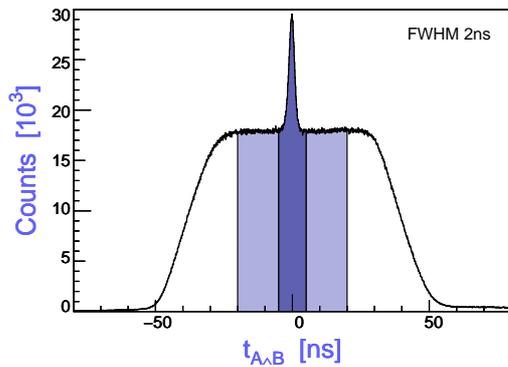


Abb. 5.2: Koinzidenzzeitpektrum mit Schnitten um (dunkelblau) und neben (hellblau) den Koinzidenzzeitpik

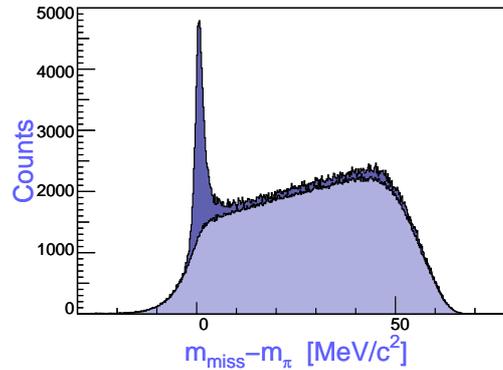


Abb. 5.3: Rekonstruierte Masse abzüglich der Pionmasse aus dem Koinzidenzzeitpik (dunkelblau) und dem skalierten Untergrund (hellblau)

Abbildung 5.2 zeigt ein Koinzidenzzeitpektrum am Beispiel der Kinematik 4-3. Erkennbar ist ein scharfer Koinzidenzzeitpik zwischen dem gestreuten Elektron und dem Rückstoßproton, begleitet von einem breiten Untergrund an zufälligen Ereignissen. Unter der Annahme, daß die zufälligen Ereignisse gleichverteilt sind, kann durch Schnitte neben und um den Koinzidenzzeitpik der Untergrund an Zufallskoinzidenzen von den übrigen Ereignissen getrennt werden. Mittels einer Subtraktion der beiden Ereignismengen werden anschließend die „echten“ Ereignisse für die weitere Datenanalyse extrahiert.

Die Untergrundsubtraktion wird zur Identifikation des π^0 anhand der fehlenden Pionmasse angewendet. Abbildung 5.3 zeigt die Abweichung der rekonstruierten Masse von der Masse des π^0 . Eingezeichnet sind in dunkelblau die Ereignisse aus dem Schnitt auf den Koinzidenzzeitpik und in hellblau die Untergrundereignisse. Das aus der Subtraktion resultierende Spektrum der fehlenden Masse ist in Abbildung 5.4 zu sehen.

5.3 Datenanalyse

Nach der Identifikation der $p(e,e'p)\pi^0$ -Ereignisse soll nun im nächsten Schritt aus diesen Teilchen- bzw. Reaktionszahlen der dreifach differentielle Wirkungsquerschnitt nach Gleichung 2.9 bestimmt werden. Bezogen auf ein ausreichend

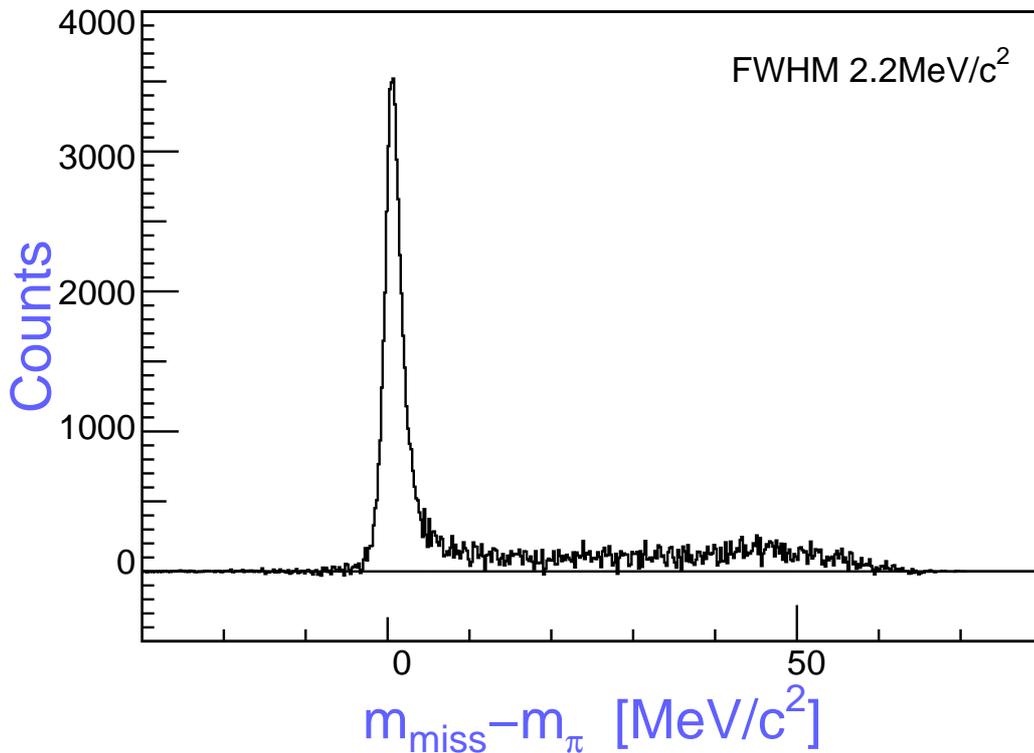


Abb. 5.4: Untergrundbereinigte rekonstruierte Masse abzüglich der Pionmasse („missing mass“). Die Halbwertsbreite beträgt $2,2\text{MeV}/c^2$.

schmales „Bin“, für das sich Mittelwert und integraler Wirkungsquerschnitt in erster Näherung nicht unterscheiden, gilt folgender Zusammenhang:

$$\frac{d^3\sigma(\theta_\pi^*, \phi_\pi^*)}{d\Omega'dE'd\Omega_\pi^*} = \frac{\text{Anzahl der Reaktionen}}{\text{Luminosität} \times \text{akzeptierter Phasenraum}} \quad (5.1)$$

Die Luminosität \mathcal{L} hat die Dimension $1/(\text{Fläche} \times \text{Zeit})$ und ist zusammengesetzt aus der Teilchenzahl-Dichte des Targetmaterials, der Länge des Targets, dem Strom des einlaufenden Elektronenstrahls und der effektiven Meßzeit¹.

Der akzeptierte Phasenraum wird mit einer Monte-Carlo-Simulation² innerhalb der vorgegebenen kinematischen Parameter bestimmt. Dabei werden Energieverluste, Strahlungskorrekturen sowie Impuls- und Winkelaufösung der Spektrometer berücksichtigt.

¹Während der Verarbeitung eines einzelnen Ereignisses wird die Ausleseelektronik verriegelt und kann in dieser Zeit keine weiteren Ereignisse detektieren. Die totale Meßzeit muß daher um diese sogenannte Totzeit korrigiert werden

²Zum Einsatz kommt das Program Simul++ [Dis01].

Die rechte Seite von Gleichung 2.9 beschreibt den differentiellen Wirkungsquerschnitt zusammengesetzt aus den fünf Anteilen $\sigma_i(\theta_\pi^*), i = T, L, TT, LT, LT'$ in Abhängigkeit vom Azimutwinkel ϕ_π^* . Da die Trennung der einzelnen Terme nur durch eine geeignete Wahl von ϕ_π^* möglich ist, liegt es nahe, die gemessenen Pionenzahlen als Funktion des Azimutwinkels zu histogrammieren und anschließend eine geeignete χ^2 -Anpassung an die Verteilung der Ereignisse durchzuführen. Für den Fall unpolarisierter Elektronen ($\bar{h} = 0$) kann Gleichung 2.9 geschrieben werden als:

$$\begin{aligned} \sigma_T(\theta_\pi^*) + \epsilon_L \sigma_L(\theta_\pi^*) + \epsilon \sigma_{TT}(\theta_\pi^*) \cos 2\phi_\pi^* + \sqrt{2\epsilon_L(1+\epsilon)} \sigma_{LT}(\theta_\pi^*) \cos \phi_\pi^* \\ = \\ A + B \cos 2\phi_\pi^* + C \cos \phi_\pi^* \end{aligned} \quad (5.2)$$

Ersetzt man nun den linken Teil der Gleichung 5.1 durch y und in 5.2 den Azimutwinkel ϕ_π^* durch x , so erhält man eine einfache Gleichung, bei der die Koeffizienten A, B und C leicht mit Hilfe einer χ^2 -Anpassung bestimmt werden können³:

$$y = A + B \cos 2x + C \cos x \quad (5.3)$$

Bisher nicht beachtet wurde der Umstand, daß die Koeffizienten A, B und C nicht konstant, sondern ihrerseits mit den kinematischen Größen ϵ und θ_π^* korreliert sind. Zudem wurden die Abhängigkeiten des differentiellen Wirkungsquerschnitts von der Schwerpunktsenergie W und dem Viererimpulsübertrag Q^2 noch nicht berücksichtigt. Mit einer geeigneten Binierung in den relevanten Größen können diese Abhängigkeiten untersucht werden:

- Die Schwerpunktsenergie W wird in Schritten von 1 MeV bis zu einer Energie von 40 MeV über der Pion-Produktionsschwelle variiert.
- Der Viererimpulsübertrag Q^2 wird in einem engen Bereich um $0,05 \text{ (GeV}/c)^2$ akzeptiert. Eine Differenzierung ist hier folglich unnötig. Dies betrifft ebenso den Polarisationsgrad ϵ , da dieser scharf mit Q^2 korreliert ist und den gleichen schmalen Akzeptanzbereich aufweist.
- Der Schwerpunktswinkel θ_π^* weist unterschiedliche Akzeptanzen in den verschiedenen Kinematiken auf. Mittels einer Binierung bzw. durch einen Schnitt um $\theta_\pi^* = 90^\circ$ wird bei Kinematik 4-3 der Wirkungsquerschnitt differentiell nach θ_π^* untersucht. Für die übrigen Kinematiken wird außer dem Schnitt um $\theta_\pi^* = 90^\circ$ alternativ ein modellabhängiger Transport auf die Nominalkinematik durchgeführt (siehe Abschnitt 5.4).

³Die Minimierung wird mit Hilfe des Matrixformalismus [Neu89] durchgeführt. Die Meßwert- und Normalmatrix dazu sind werden im Anhang A.1 gezeigt.

Tabelle 5.2 zeigt den akzeptierten Bereich der verschiedenen kinematischen Variablen und die jeweilige Binierung.

	akzeptierter Bereich	Kinematik	Bins
W/MeV	1073 – 1113	0-6..4-2	40
	1073 – 1077	4-3	4
$Q^2/(\text{GeV}/c)^2$	0,044 – 0,057	alle	1
ϵ	0,923 – 0,937		
ϕ_π^*	$(-180^\circ) - (180^\circ)$	0-6..4-2	36
		4-3	9
θ_π^*	Maid Transport	0-6..4-2	18
	$75^\circ - 105^\circ$		1
	$\cos 180^\circ - \cos 0^\circ$	4-3	10
	$75^\circ - 105^\circ$		1

Tab. 5.2: Binierung der kinematischen Variablen

Bei der χ^2 -Minimierung von Gleichung 5.3 wurde die Helizitätsinformation nicht berücksichtigt, was einem mittleren Polarisationsgrad des Elektronenstrahls von $\bar{h} = 0$ entspricht. Für $\bar{h} \neq 0$ trägt zusätzlich der Interferenzterm $\sigma_{LT'}$ zum Wirkungsquerschnitt bei. Dieser kann relativ einfach durch eine Subtraktion des unpolarisierten Anteils vom gesamten Wirkungsquerschnitt extrahiert werden:

$$\sigma_{LT'}(\theta_\pi^*) \sin \phi_\pi^* = \frac{1}{h\sqrt{2\epsilon_L(1-\epsilon)}} \left(\frac{d^3\sigma(\theta_\pi^*, \phi_\pi^*)}{d\Omega' dE' d\Omega_\pi^*} - \frac{d^3\sigma(\theta_\pi^*, \phi_\pi^*)}{d\Omega' dE' d\Omega_\pi^*} \Big|_{\bar{h}=0} \right) \quad (5.4)$$

Dabei ist $\sigma_{LT'}$ nahe der Pion-Produktionsschwelle im Vergleich mit den Wirkungsquerschnitten σ_0 und σ_{LT} sehr klein. Dementsprechend werden viele Ereignisse benötigt, um den relativen statistischen Fehler möglichst gering zu halten. Ein zweites Problem stellen die systematischen Unsicherheiten dar, da diese bei kleinen Wirkungsquerschnitten das physikalische Ergebnis stark verfälschen können. Eine geeignetere Methode ist daher die Bildung der Helizitätsasymmetrie $A_{LT'}(\phi_\pi^* = 90^\circ)$, weil sich dort die faktorisierbaren systematischen Fehler gegenseitig wegheben:

$$A_{LT'}(\phi_\pi^* = 90^\circ) = \frac{\sqrt{2\epsilon_L(1+\epsilon)}\sigma_{LT'}(\theta_\pi^*)}{\sigma_T(\theta_\pi^*) + \epsilon\sigma_L(\theta_\pi^*) - \epsilon\sigma_{TT}(\theta_\pi^*)} \quad (5.5)$$

$$= \frac{\sigma(h=1) - \sigma(h=-1)}{\sigma(h=1) + \sigma(h=-1)} \quad (5.6)$$

Während des Experimentes wird die Helizität im Sekundentakt mittels eines Zufallsgenerators auf ± 1 festgelegt. Für beide Helizitäten liegen somit die gleiche

Phasenraumakzeptanz, Luminosität und Detektoreffizienz vor. Der tatsächliche Polarisationsgrad des Elektronenstrahls wird durch die Gewichtung der nachgewiesenen Ereignisse mit $h_{\text{eff}} \cdot \sin \phi_{\pi}^*$ berücksichtigt und eine vorhandene Untergrundasymmetrie durch einen Untergrundabzug, wie in Abschnitt 5.2 beschrieben, beseitigt⁴.

5.4 MAID-Transport

Die χ^2 -Anpassung von Gleichung 5.3 setzt eine Histogrammierung der Ereignisse als Funktion von ϕ_{π}^* bei einem festen Schwerpunktswinkel θ_{π}^* voraus. Dies kann bei Kinematik 4-3 leicht durch einen Schnitt um $\theta_{\pi}^* = 90^\circ$ erreicht werden, wie in Abbildung 5.5 zu sehen ist.

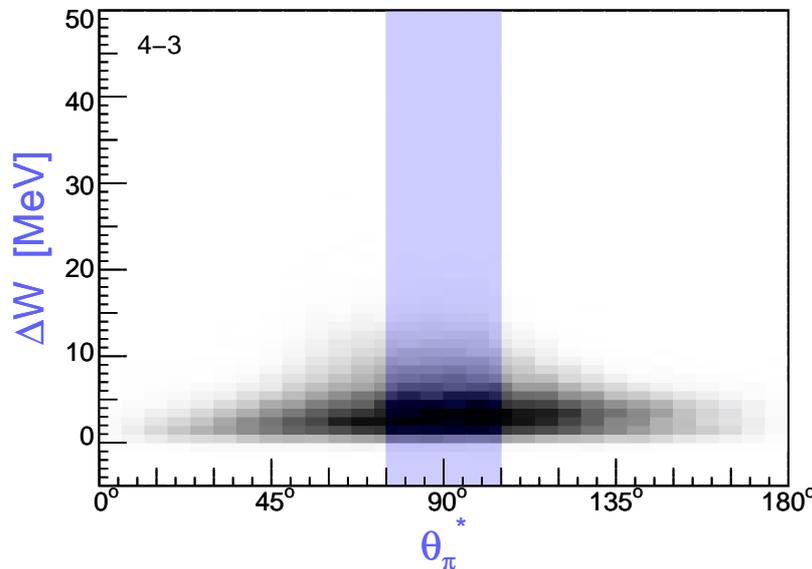


Abb. 5.5: Simulierte θ_{π}^* -Akzeptanz von Kinematik 4-3. Blau markiert ist der Bereich für einen Schnitt um $\theta_{\pi}^* = 90^\circ$.

Aufgrund der unterschiedlichen Akzeptanzen der Kinematiken 0-6 bis 4-2 ist der Schnitt dort nur eingeschränkt möglich (siehe Abbildung 5.6). Insbesondere bei den Kinematiken 4-1 und 4-2 würden vergleichsweise viele Ereignisse verworfen und damit der statistische Fehler vergrößert werden.

Abhilfe schafft eine „Korrektur“ der gemessenen Wirkungsquerschnitte mit den Modellvorhersagen aus MAID2000, bei der die Daten auf die Nominalkinematik bei $\theta_{\pi}^* = 90^\circ$ projiziert werden. Der Transport der Daten erfolgt binweise.

⁴Bei der Datenanalyse kann davon ausgegangen werden, daß eine Asymmetrie der Untergrundereignisse gleichverteilt bezüglich der Ereignisse neben und um den Koinzidenzeitpik ist.

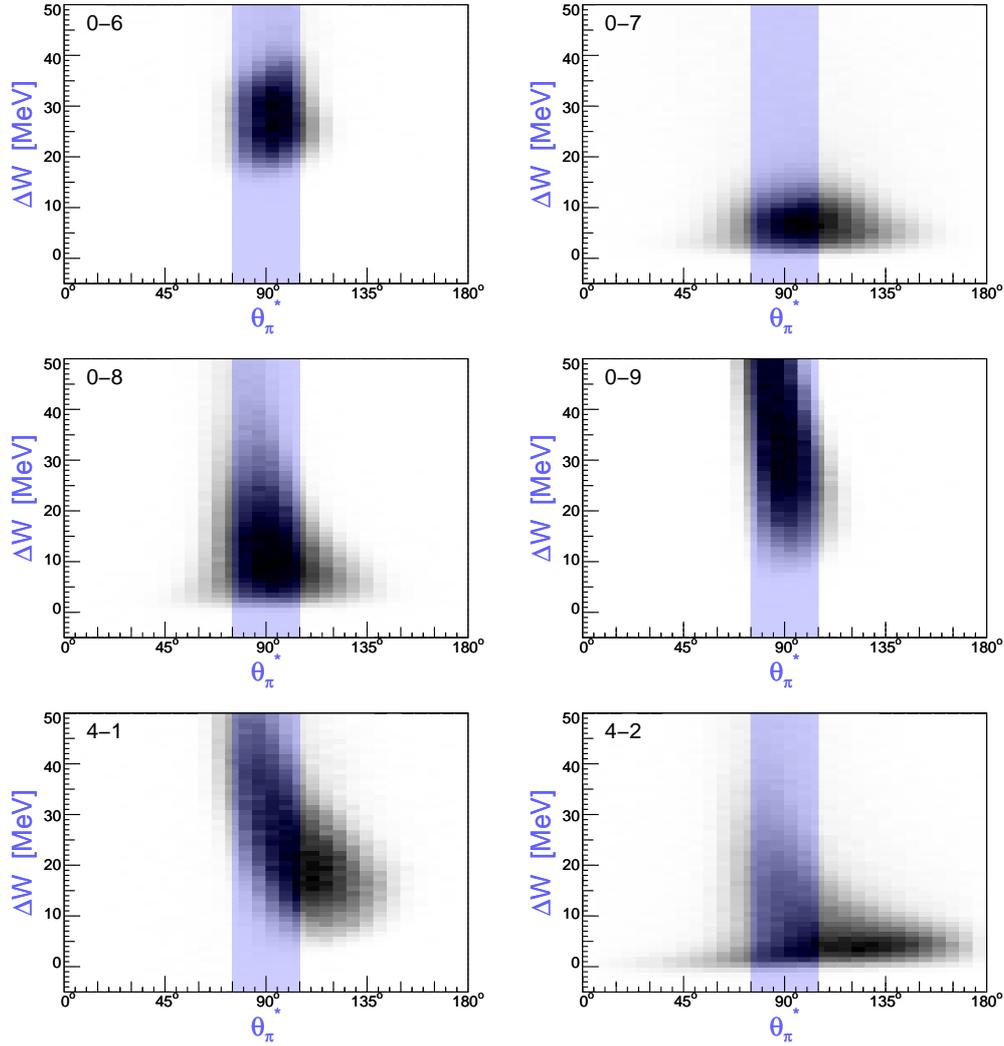


Abb. 5.6: Akzeptanzen der Kinematiken 0-6 bis 4-2. Blau markiert ist der Bereich für den Schnitt um $\theta_\pi^* = 90^\circ$, so wie er bei Kinematik 4-3 angewendet wurde.

$$\frac{d^3\sigma(90^\circ, \phi_\pi^*)}{d\Omega' dE' d\Omega_\pi^*} = \frac{d^3\sigma(\theta_\pi^*, \phi_\pi^*)}{d\Omega' dE' d\Omega_\pi^*} \frac{\text{MAID}(90^\circ, \phi_\pi^*)}{\text{MAID}(\theta_\pi^*, \phi_\pi^*)} \quad (5.7)$$

Da nur relative Änderungen an den Wirkungsquerschnitten vorgenommen werden, ist es prinzipiell nicht notwendig, den absoluten Wirkungsquerschnitt in MAID2000 genau zu kennen. Wie allerdings aus Abbildung 5.7 ersichtlich ist, sind die Korrekturen relativ groß, so daß eine nähere Untersuchung der modellabhängigen Fehler durch den Transport der Daten mit MAID2000 erforderlich ist.

Zunächst ist festzustellen, daß der Transport selbstkonsistent ist (siehe Abbildung 5.8). Eine relative Änderung des Wirkungsquerschnitts führt zu einer absoluten

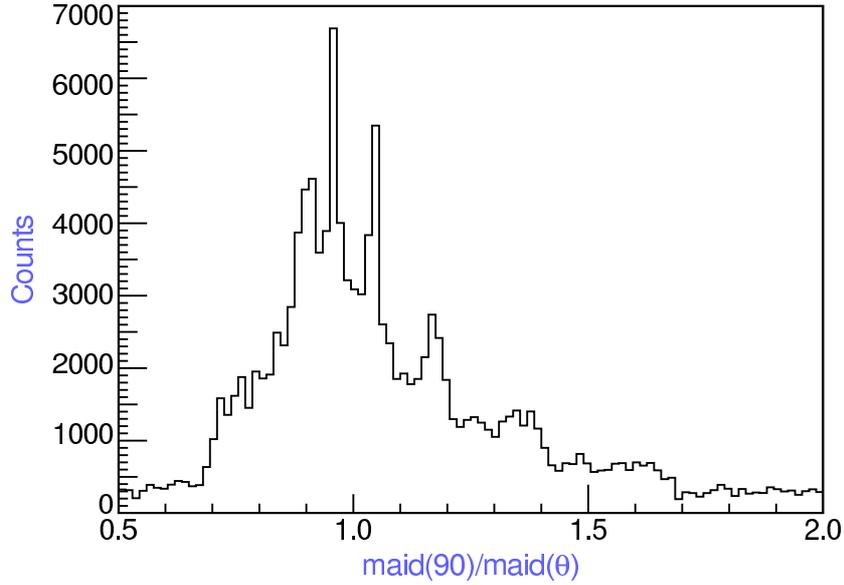


Abb. 5.7: Häufigkeitsverteilung des MAID-Transportfaktors pro Bin. Ein Wert von 1.0 bedeutet, daß der gemessene Wirkungsquerschnitte nicht transportiert werden muß.

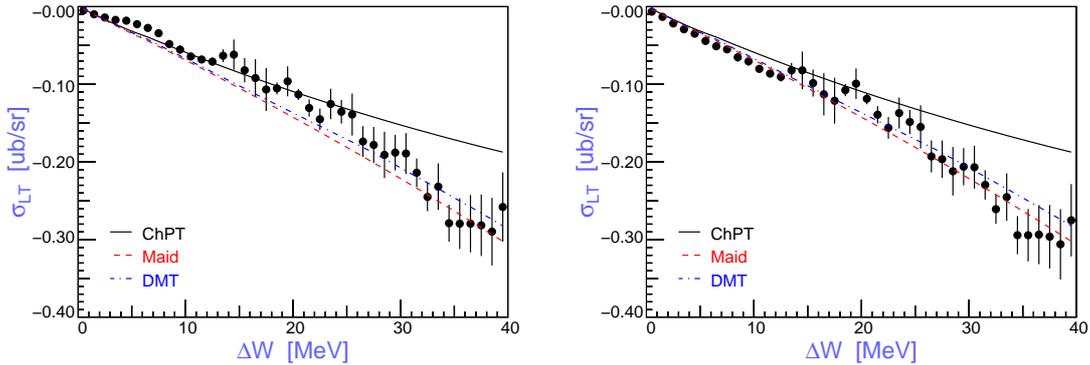


Abb. 5.8: Longitudinal-Transversal Interferenzterm als Funktion der Schwerpunktsenergie ΔW . Das linke Histogramm zeigt die ermittelten Wirkungsquerschnitte ohne Korrektur und ohne Schnitt auf θ_π^* und das rechte die Daten nach dem Transport mit MAID2000 auf die Nominalkinematik.

Übereinstimmung zwischen Modell und Daten. Dies ist nicht selbstverständlich und zeigt, daß der Modellfehler in der gleichen Größenordnung wie der systematische Fehler liegt. Um die Modellabhängigkeit weiter zu untersuchen, wurden die Multipolamplituden bis $l = 1$ um einen willkürlichen⁵ Faktor geändert und die Auswirkungen auf den Transportfaktor untersucht.

Die Variation der Multipolamplituden wurde derart implementiert, daß der ent-

⁵Die Multipolamplituden wurden jeweils um 10% erhöht. Für die dominanten Amplituden wie beispielsweise M_{1+} ist dies sicherlich ein übertrieben großer Wert.

sprechende Anteil an den Helizitätsamplituden H_1 bis H_6 variiert wurde. Tabelle 5.3 listet die benötigten Koeffizienten für die Zerlegung.

	E_{0+}	M_{1-}	E_{1+}	M_{1+}
$H_1(\theta)$	0	0	$-\frac{3}{\sqrt{2}} \sin \theta \cos \frac{\theta}{2}$	$\frac{3}{\sqrt{2}} \sin \theta \cos \frac{\theta}{2}$
$H_2(\theta)$	$-\sqrt{2} \cos \frac{\theta}{2}$	$\sqrt{2} \cos \frac{\theta}{2}$	$\frac{3}{\sqrt{2}} \cos \frac{\theta}{2} (1 - 3 \cos \theta)$	$\frac{1}{\sqrt{2}} \cos \frac{\theta}{2} (1 - 3 \cos \theta)$
$H_3(\theta)$	0	0	$\frac{3}{\sqrt{2}} \sin \theta \sin \frac{\theta}{2}$	$-\frac{3}{\sqrt{2}} \sin \theta \sin \frac{\theta}{2}$
$H_4(\theta)$	$\sqrt{2} \sin \frac{\theta}{2}$	$\sqrt{2} \sin \frac{\theta}{2}$	$\frac{3}{\sqrt{2}} \sin \frac{\theta}{2} (1 + 3 \cos \theta)$	$\frac{1}{\sqrt{2}} \sin \frac{\theta}{2} (1 + 3 \cos \theta)$

	L_{0+}	L_{1-}	L_{1+}
$H_5(\theta)$	$\cos \frac{\theta}{2}$	$\cos \frac{\theta}{2}$	$-2 \cos \frac{\theta}{2} (1 - 3 \cos \theta)$
$H_6(\theta)$	$-\sin \frac{\theta}{2}$	$\sin \frac{\theta}{2}$	$-2 \sin \frac{\theta}{2} (1 + 3 \cos \theta)$

Tab. 5.3: Koeffizienten für die Zerlegung der Helizitätsamplituden in ihre Multipolanteile bis $l = 1$.

Abbildung 5.9 zeigt die Auswirkungen der Variation der Multipolamplituden. Wie nicht anders erwartet, wird der Transport dominiert von der Genauigkeit der Multipole E_{0+} , L_{0+} und $M_{1\pm}$. Die höheren Amplituden spielen nur eine untergeordnete Rolle. Da eine Änderung um 10% den Wirkungsquerschnitt im Mittel nur um etwa 5% ändert, wird der absolute Fehler durch den MAID-Transport ebenfalls kleiner 5% angenommen.

Im Anhang (Abschnitt A.2) sind die ermittelten Wirkungsquerschnitte sowohl für den MAID-Transport als auch für den Schnitt um $\theta_\pi^* = 90^\circ$ aufgeführt. Beide Methoden liefern innerhalb des Gesamtfehlers das gleiche Ergebnis. Auf der einen Seite hat der Transport die bessere Statistik, dafür aber einen modellabhängigen Fehler und auf der anderen Seite führt der Schnitt in θ_π^* zu einer größeren statistischen Unsicherheit.

5.5 Fehlerbetrachtungen

Abgesehen von den statistischen Fehlern, die bei Experimenten mit Reaktionswahrscheinlichkeiten unvermeidbar auftreten und Unsicherheiten bei der Modellierung des physikalischen Vorganges, wie zum Beispiel Strahlungskorrekturen, sind Unzulänglichkeiten im apparativen Aufbau sowie der softwareseitigen Abbildung desselben eine dominante Fehlerquelle.

In Kapitel 4 wurden bereits einige der systematischen Fehler des Aufbaus diskutiert mit dem Resultat, daß sie für das Experiment von untergeordneter Bedeutung sind. Ebenso können die Fehler bei der Bestimmung der Luminosität vernachlässigt werden, da der Strahlstrom hinreichend genau bekannt ist, Dichteschwankungen im Targetmaterial nahezu ausgeschlossen werden, die Targetlänge

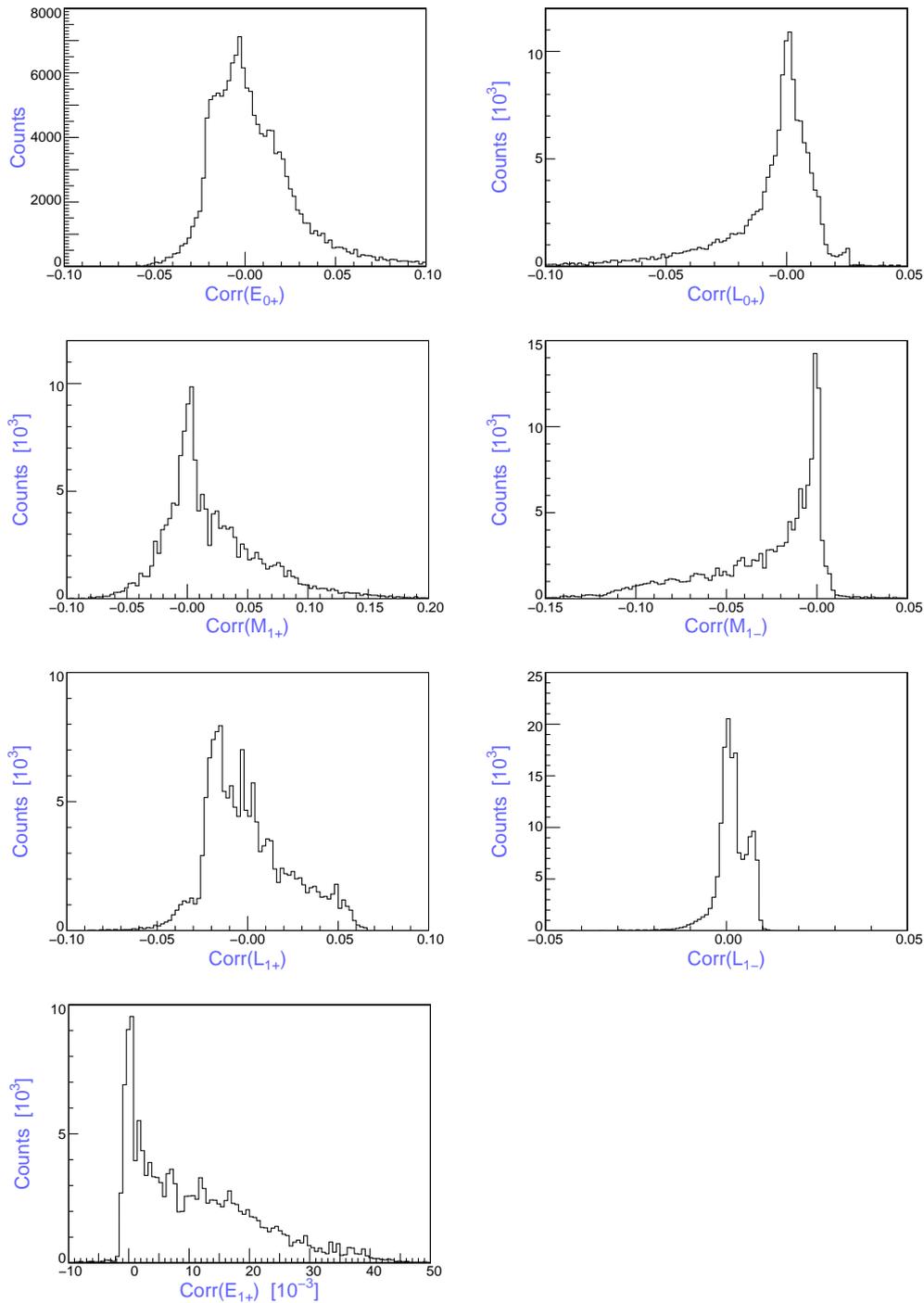


Abb. 5.9: Auswirkungen der Variation der Multipolamplituden bis $l = 1$ um 10% auf den MAID-Transportfaktor. Die Zahlen entsprechen relativen Änderungen $\left(\frac{\text{Faktor}(\text{neu}) - \text{Faktor}(\text{alt})}{\text{Faktor}(\text{alt})}\right)$ des Transportfaktors bezüglich der Nominalkinematik.

stabil ist und die Totzeit gemessen wird. Als Obergrenze für den systematischen Fehler werden hier 2% angenommen. Allerdings hat der Polarisationsgrad des Elektronenstrahls eine relativ große Unsicherheit, da er während des Experimentes nicht ermittelt werden konnte.

Betrachtet man Gleichung 5.1, so fällt auf, daß die bisher untersuchten Fehler entweder die Luminosität oder, aufgrund von Detektorineffizienzen, die Anzahl der Reaktionen beeinflussen.

In den Abschnitten dieses Kapitels werden nun diejenigen Fehlerquellen vorgestellt, die die Simulation bzw. Akzeptanz des Phasenraumes beeinflussen.

5.5.1 Zentralimpuls von Spektrometer B

Bei der Pion-Elektroproduktion wird der Viererimpuls des virtuellen Photons aus dem Elektronarm berechnet. Da die absolute Energie des Photons die Lage der Pion-Produktionsschwelle bestimmt, ist die genaue Kenntnis des Zentralimpulses von Spektrometer B von entscheidender Bedeutung. Der Fehler wurde mit der Kalibrationskinematik absolut auf $< 300 \text{ keV}/c$ festgelegt.

Eine Variation des Impulses von Spektrometer B führt zu dem Problem, daß in der Simulation Ereignisse unter die Produktionsschwelle verschoben werden und so bei der Berechnung des Wirkungsquerschnitts auf den falschen Phasenraum normiert wird. Der Fehler betrifft hauptsächlich die niederenergetischen Bins, da sich dort aufgrund des Schwellenverhaltens der Wirkungsquerschnitt am schnellsten ändert.

Die relativen Fehler $\frac{\delta\sigma}{\sigma}$ für die ersten vier ΔW -Bins sind: 43% ($\Delta W = 0,5 \text{ MeV}$), 21% ($1,5 \text{ MeV}$), 2% ($2,5 \text{ MeV}$), 1% ($3,5 \text{ MeV}$). Für $\Delta W > 4 \text{ MeV}$ sinkt der Fehler auf unter 1%. Die Ungenauigkeit von $300 \text{ keV}/c$ betrifft nur den Absolutimpuls des Spektrometers. Die relativen Abstände der W -Bins untereinander sind auf $\leq 10^{-4}$ genau bekannt.

5.5.2 Position der Targetzelle

Kinematik 4-3 zeichnet sich durch seine volle Akzeptanz im Azimutwinkel ϕ_π^* aus. Dies hat zwar den großen Vorteil, daß mit einer einzigen Einstellung der gesamte Winkelbereich abgedeckt wird, gleichzeitig steigen aber auch die Anforderungen an die Simulation des Phasenraumes, da Ereignisse an den Akzeptanzrändern stärker zum Wirkungsquerschnitt beitragen als bei den übrigen kombinierten Kinematiken. Diese Problematik soll anhand von Abbildung 5.10 verdeutlicht werden.

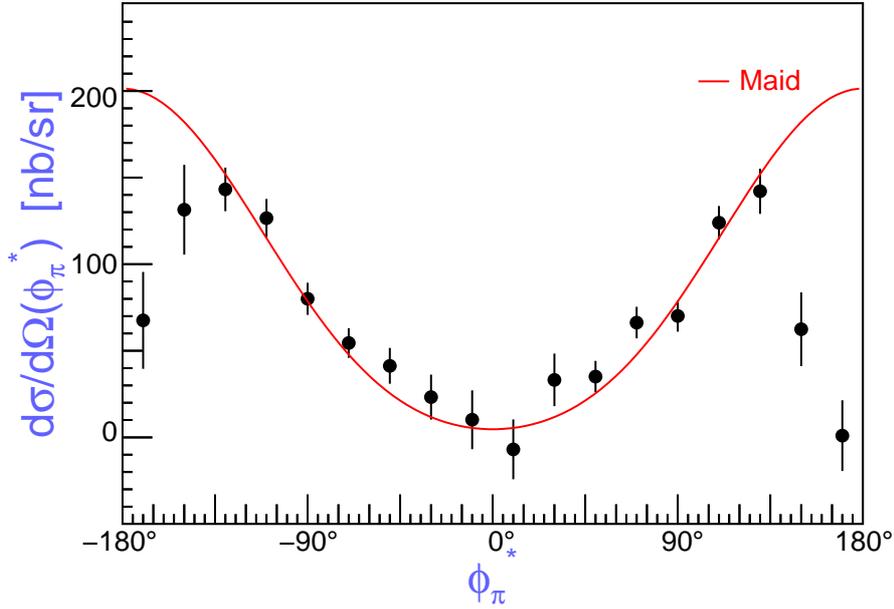


Abb. 5.10: Differentieller Wirkungsquerschnitt als Funktion des Azimutwinkels ϕ_π^* für das Bin $\Delta W = 3,5$ MeV. Eingezeichnet in rot ist die theoretische Vorhersage aus MAID2000. Bei diesem ΔW -Bin ist der untersuchte Effekt am größten.

An den Datenpunkten wird eine χ^2 -Minimierung durchgeführt, um die Wirkungsquerschnitte σ_0 und σ_{LT} zu extrahieren. Es fällt auf, daß die Punkte bei $|\phi_\pi^*| > 135^\circ$ nicht durch eine $\cos \phi_\pi^*$ -Verteilung nach Gleichung 2.9 beschrieben werden können. Die χ^2 -Anpassung an alle Datenpunkte würde folglich einen im Vergleich mit MAID viel zu kleinen Wirkungsquerschnitt liefern. Läßt man dagegen die „schlechten“ Punkte weg, so steigt beispielsweise für das Bin $\Delta W = 3,5$ MeV der ermittelte Wirkungsquerschnitt σ_0 um 31%. Es ist also zu klären, welchen Grund die zu kleinen Wirkungsquerschnitte haben und ob es eine Rechtfertigung gibt, die ϕ_π^* -Akzeptanz bei der χ^2 -Anpassung auf $\pm 135^\circ$ einzuschränken.

Der Schwerpunktwinkel ϕ_π^* ist gekoppelt mit dem Protonstreuwinkel und dementsprechend gibt es eine Korrelation zwischen ϕ_π^* und der Koordinate ϕ_0 für Spektrometer A. Abbildung 5.11 verdeutlicht diesen Zusammenhang. Die problematischen Datenpunkte liegen demnach an den Akzeptanzrändern von Spektrometer A bei kleinen ϕ_0 an der linken Kollimatorbacke.

Die ϕ_0 -Koordinate ihrerseits wird primär auf die y -Koordinate in der Fokalebene abgebildet und hängt in bezug auf ihre Akzeptanz gleichzeitig von der z -Koordinate am Reaktionsort ab. Es müssen folglich zwei mögliche Fehlerquellen untersucht werden: Ineffizienzen in einem der Fokalebenendetektoren und Position der Targetzelle.

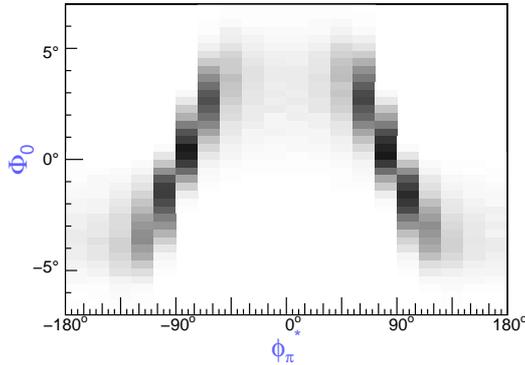


Abb. 5.11: Abhängigkeit der Targetkoordinate ϕ_0 vom Azimutwinkel ϕ_π^* für Spektrometer A.

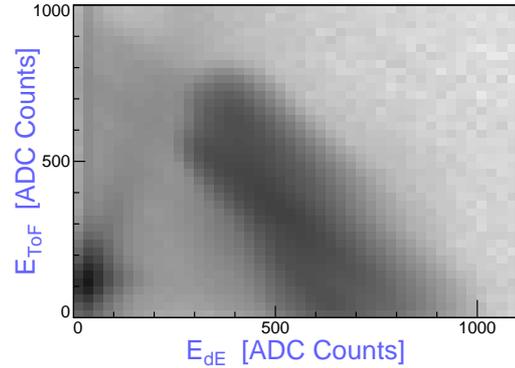


Abb. 5.12: Energiedeposition in der dE-Ebene aufgetragen gegen die ToF-Ebene. Die Daten sind logarithmisch eingezeichnet.

- a) **Fokalebene:** Die Szintillatorsegmente sind entlang der y -Koordinate ausgerichtet und werden über Photomultiplier gleichzeitig beidseitig ausgelesen. Eine zu geringe Verstärkung an einem der beiden Photomultiplier führt zu einer ungleichmäßigen Verteilung der Ereignisse in y -Richtung, da nur diejenigen Ereignisse nachgewiesen werden, die in der Nähe des Photomultipliers ein ausreichend großes Signal liefern. Das Resultat wären zu geringe Ereigniszahlen in ϕ_0 an einer der Kollimatorbacken.

Abbildung 5.12 zeigt die Energiedeposition des nachgewiesenen Teilchens in den Szintillatorebenen von Spektrometer A. Zu erkennen ist bei kleinen dE/ToF-Kanalnummern eine Menge von Ereignissen, die als π^+ identifiziert werden können sowie eine breite Verteilung (Protonen) bei großen Kanalnummern. Würden Protonen aufgrund eines zu kleinen Signals in den Photomultipliern nicht nachgewiesen, so müsste die breite Verteilung deutlich bei kleinen dE-Kanalnummern sichtbar sein. Da dies nicht erkennbar ist, kann davon ausgegangen werden, daß die Problematik der ϕ_0 -Verteilung nicht auf Ineffizienzen des Szintillators zurückzuführen ist.

- b) **Zellenposition:** Da die Position der Targetzelle die Menge der möglichen z -Koordinaten festlegt, beeinflusst sie indirekt auch die ϕ_0 -Akzeptanz. Bei den Vorbereitungen für ein Experiment wird das Target so justiert, daß die Zelle symmetrisch um $z_0 = 0$ liegt. Beim anschließenden Abkühlen und Befüllen mit flüssigem Wasserstoff kann jedoch nicht ausgeschlossen werden, daß sich die Targetaufhängung verformt und somit die Lage der Targetzelle verändert. Ein Indiz für eine Verschiebung der Zelle ist der Vergleich mit der leeren Targetzelle, die sich für Kalibrationszwecke auf der Targetleiter befindet und nicht mit dem Targetkreislauf verbunden ist.

Abbildung 5.13 zeigt die Position der abgekühlten Targetzelle im Vergleich mit der Leerzelle auf der Targetleiter. Die Differenz für die untersuchte Kinematik beträgt zumindest 2 mm, weshalb der Fehler der Targetposition in dieser Größenordnung angenommen wird. Da der Kühlkreislauf im Verlauf des Experiments mehrmals erwärmt und wieder abgekühlt wurde und zudem nicht untersucht wurde, ob das Herein- und Herausfahren der Targetzelle aus dem Elektronenstrahl die Position verändert, können die 2 mm nicht allgemeingültig angenommen werden, sondern dienen lediglich als Anhaltspunkt für die Größe der Unsicherheit.

Eine Variation der simulierten Targetposition hat einen deutlichen Einfluß auf die Verteilung der ϕ_0 -Ereignisse, wie an Abbildung 5.14 zu erkennen ist. Dies betrifft vor allem die Wirkungsquerschnitte für $|\phi_\pi^*| > 135^\circ$.

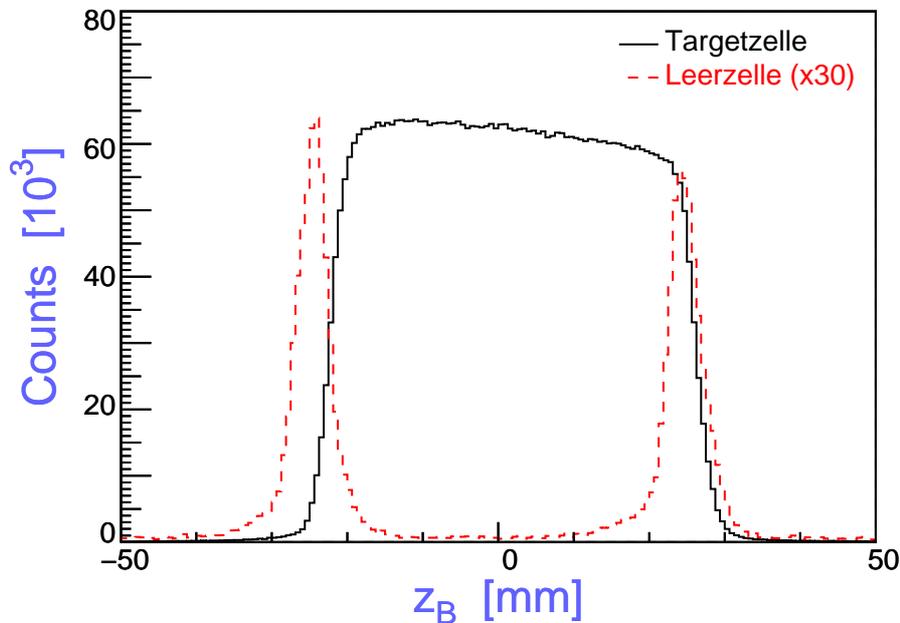


Abb. 5.13: Position der kalten Targetzelle im Vergleich mit der Leerzelle. Die Daten stammen von der Kalibrationskinematik aus Kapitel 4.

Die Variation der z_0 -Koordinate kann die zu kleinen Wirkungsquerschnitte in den betreffenden Winkelbereichen nicht vollständig erklären. Dennoch reicht es aus zu erkennen, daß die systematischen Unsicherheiten zu groß sind um bei der χ^2 -Anpassung nicht beachtet zu werden. Es muß also entweder der Fehler für $|\phi_\pi^*| > 135^\circ$ ausreichend groß angegeben werden, damit die Daten die Minimierung nicht beeinflussen oder man läßt die Datenpunkte komplett weg. Da eine χ^2 -Minimierung von einer Gauß-Verteilung der Ereignisse ausgeht, was im vorliegenden Fall sicherlich nicht gegeben ist, werden die Datenpunkte verworfen.

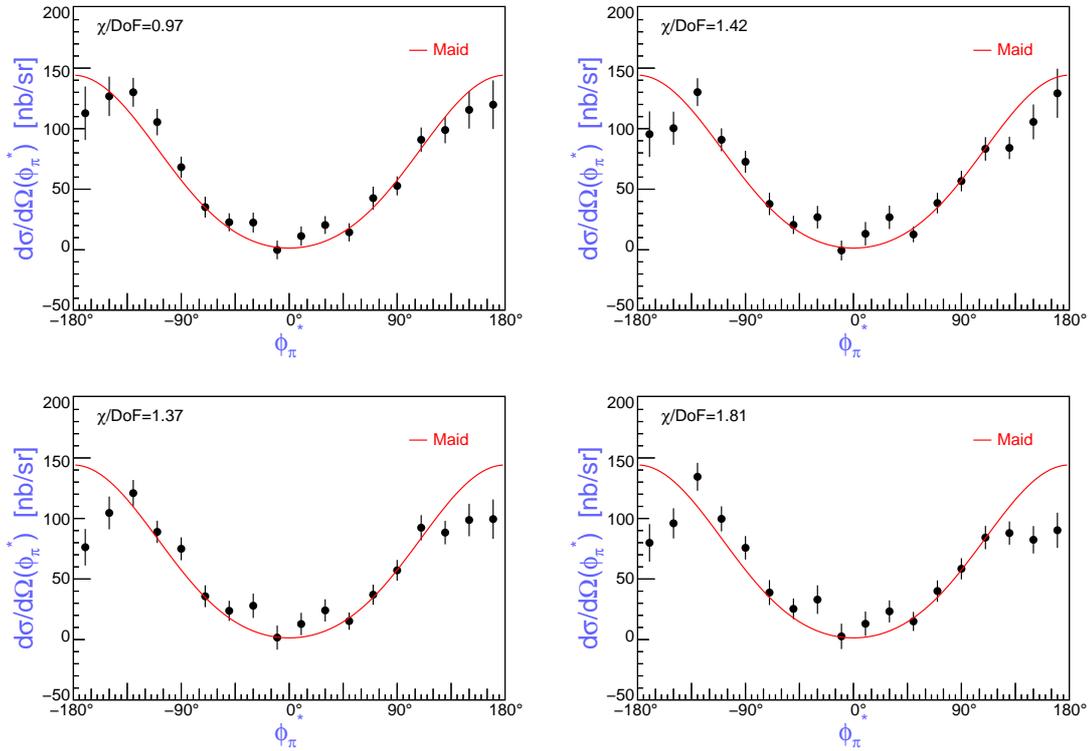


Abb. 5.14: Wirkungsquerschnitt ($\Delta W = 2,5\text{MeV}$) als Funktion des Azimutwinkels ϕ_0 bei einer Variation in $z_0 = +2\text{mm}, +0\text{mm}, -2\text{mm}, -4\text{mm}$ (dargestellt von links oben nach rechts unten). Die Veränderung der Targetposition betrifft vor allem die Bins bei $|\phi_\pi^*| > 135^\circ$.

5.5.3 Form der Targetzelle

Bei den Kinematiken 4-1 und 4-2 wird Spektrometer B zum Nachweis des gestreuten Elektrons aus der Ebene gekippt. Dabei wird vorausgesetzt, daß die Targetzelle eine ausreichend große Akzeptanz in dispersiver Richtung hat, damit das Elektron diese auch verlassen kann. Dies betrifft insbesondere Kinematik 4-1, bei der das Spektrometer um $7,65^\circ$ geneigt wird.

Anhand von Abbildung 5.15 wird deutlich, daß die Stützbacken der Targetzelle das Elektron in seiner Bahn beeinträchtigen und somit dessen Energie und Streuwinkel ändern. Für das gewählte Beispiel ($\theta_e^{LAB} = 15,4^\circ$, $\phi_e^{LAB} = 7,65^\circ$) betrifft dies rein geometrisch alle Elektronen mit dem Reaktionsort $z < 6,6\text{mm}$. Die Akzeptanz des Spektrometers in dispersiver Richtung verschärft das Problem zusätzlich. Umgekehrt kommt hinzu, daß sich beim Abkühlen des Targetkreislaufes die Zelle nicht nur in z -Richtung bewegt (siehe Abschnitt 5.5.2), sondern auch entlang der y -Achse, was den Spielraum in dispersiver Richtung wieder etwas vergrößert. Der tatsächliche Reaktionsort, ab dem die Daten verwendbar

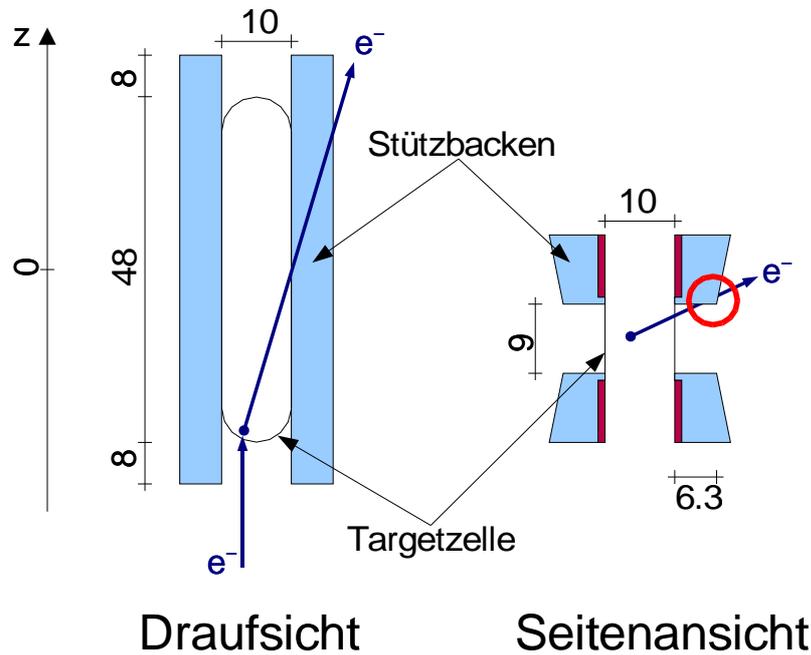


Abb. 5.15: Geometrie der Targetzelle. Eingezeichnet ist in blau die Bahn eines einlaufenden Elektrons, welches exemplarisch unter dem Laborwinkel $\theta_e^{LAB} = 15,4^\circ$, $\phi_e^{LAB} = 7,65^\circ$ gestreut wird. Die Wedleramplitude beträgt horizontal ± 2 mm und vertikal $\pm 0,5$ mm

sind, wird daher experimentell durch Schnitte an verschiedenen z -Positionen bestimmt. Entscheidendes Hilfsmittel dabei ist der Überlapp mit Kinematik 4-2, bei der das Problem nicht beobachtbar ist. Der Schnitt wird so gewählt, daß Kinematik 4-1 die gleichen absoluten Wirkungsquerschnitte liefert wie Kinematik 4-2. Nach dieser Methode sind die Ereignisse mit $z > 0$ mm für die weitere Datenanalyse nutzbar.

5.6 Ergebnisse

Mit Kinematik 4-3 wurden die Wirkungsquerschnitte σ_0 und σ_{LT} für vier Bereiche bis 4 MeV oberhalb der Pion-Produktionsschwelle bestimmt. Abbildung 5.17 zeigt den Verlauf dieser Wirkungsquerschnitte als Funktion des Polarwinkels θ_π^* . Der Transversal-transversal-Interferenzterm σ_{TT} ist in diesen Bereichen innerhalb der statistischen Fehler mit null verträglich. Bei der χ^2 -Anpassung an die Datenpunkte wurde die auf $\pm 135^\circ$ eingeschränkte ϕ_π^* -Akzeptanz verwendet.

Eine Trennung der longitudinalen und transversalen Anteile des Wirkungsquerschnitts ist nicht möglich, da die Polarisation ϵ während des gesamten Experimentes nicht verändert wurde und eine Rosenbluth-Separation mindestens einen

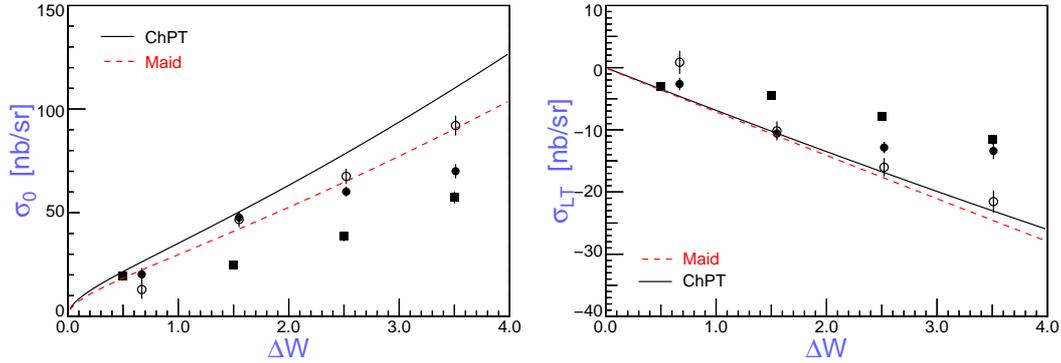


Abb. 5.16: Differentielle Wirkungsquerschnitte als Funktion der Schwerpunktsenergie ΔW . Datenpunkte: gefüllte Kreise entsprechen voller ϕ_π^* -Akzeptanz, offene Kreise $|\phi_\pi^*| \leq 135^\circ$ und die gefüllten Quader den Daten aus [Mer02].

weiteren ϵ -Punkt benötigt. Letztere wurde im Rahmen von [Mer02] durchgeführt und so liegt es nahe, die Wirkungsquerschnitte der Kinematiken mit dem gleichen ϵ miteinander zu vergleichen.

Abbildung 5.16 zeigt die Ergebnisse der beiden Experimente. Deutlich zu erkennen ist der signifikante Unterschied der ermittelten Wirkungsquerschnitte, obwohl die kinematischen Parameter aus dem alten und neuen Experiment identisch sind. Die Abweichung für das niedrigste Bin ($\Delta W = 0,5$ MeV) kann mit der Genauigkeit des Absolutwertes der Schwerpunktsenergie und für das höchste Bin ($\Delta W = 3,5$ MeV) mit den in Abschnitt 5.5.2 beschriebenen Akzeptanzproblemen erklärt werden.

Ungeklärt bleibt dagegen die Diskrepanz der beiden mittleren Energiebereiche. Die Vorgehensweise in der Datenanalyse ist bei altem und neuem Experiment vergleichbar. Als einziger großer Unterschied wird bei den alten Daten ein enger Schnitt in Q^2 gemacht. Der gleiche Schnitt bei den neuen Daten zeigt, daß sich lediglich die Ereigniszahlen reduzieren, was zu einem größeren statistischen Fehler führt.

Das Modell MAID2000 beschreibt die neuen Daten sehr gut. Für eine endgültige Klärung der Diskrepanz muß der Bereich um die Pion-Produktionsschwelle bei einem kleinen Q^2 in einem weiteren Experiment untersucht werden.

Die aus den Kinematiken 0-6 bis 4-2 gewonnenen differentiellen Wirkungsquerschnitte sind in Abbildung 5.18 dargestellt. Auch hier beschreibt MAID2000 sehr gut die differentiellen Wirkungsquerschnitte σ_0 , σ_{LT} und σ_{TT} über den gesamten akzeptierten Energiebereich. Die chirale Störungstheorie hingegen überschätzt erheblich den Beitrag des Transversal-transversal-Anteils zum totalen Wirkungsquerschnitt. Im Gegensatz zur theoretischen Vorhersage ist das gemessene σ_{TT} bis weit oberhalb der Pion-Produktionsschwelle mit null verträglich.

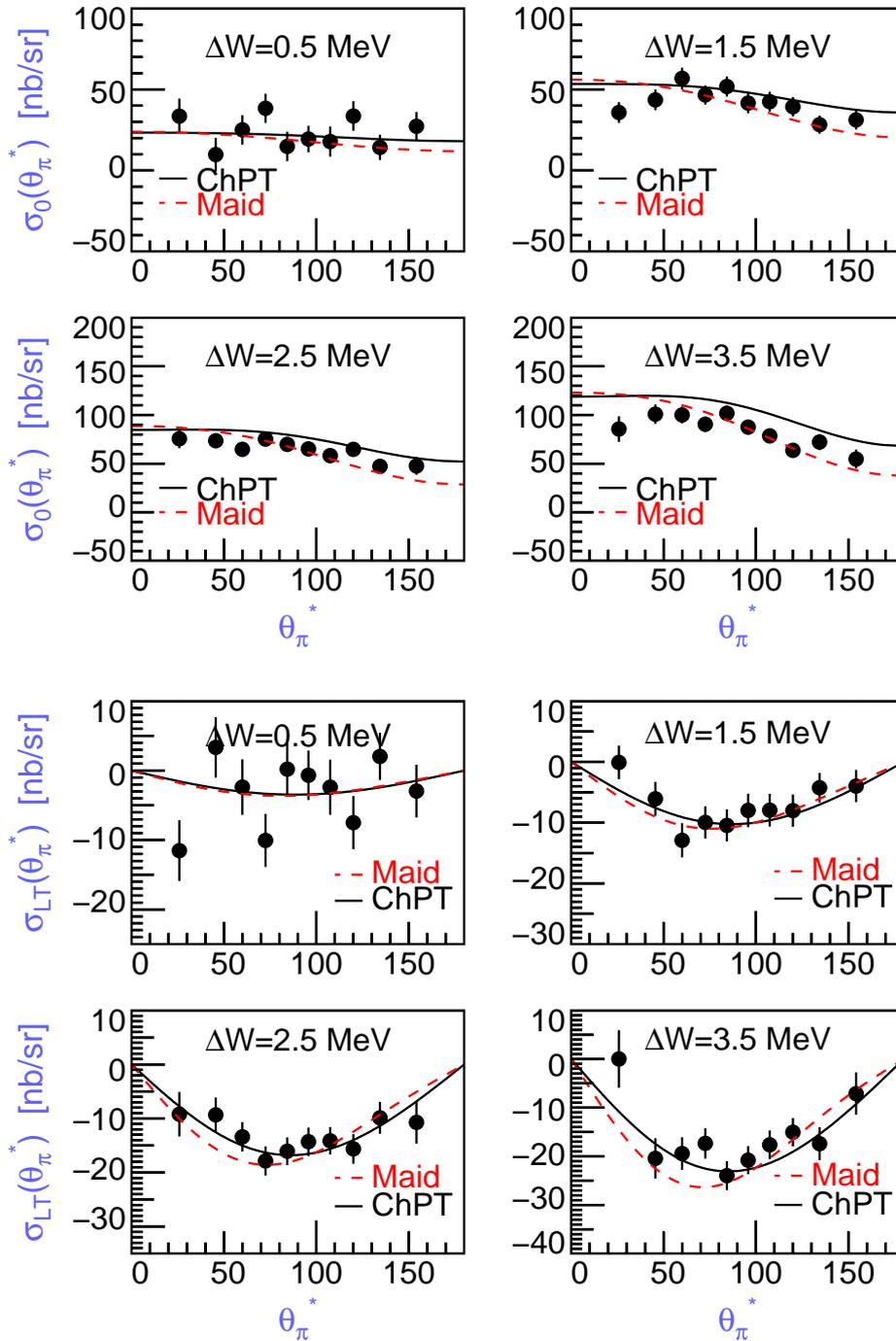


Abb. 5.17: Differentielle Wirkungsquerschnitte für die ersten 4 MeV oberhalb der Pion-Produktionsschwelle. Eingezeichnet sind die Vorhersagen aus dem Modell MAID2000 (gestrichelt) sowie der ChPT (durchgezogen).

Die Wirkungsquerschnitte der hier beschriebenen Kinematiken sind deutlich unempfindlicher gegenüber den Akzeptanzproblemen, wie sie Kinematik 4-3 hat. Dies liegt daran, daß die ϕ_π^* -Akzeptanz jedes ΔW -Bins aus jeweils drei Kinematiken zusammengesetzt ist. Die Akzeptanzränder der Kinematiken decken sich nicht mit den Rändern der ϕ_π^* -Akzeptanz und die χ^2 -Anpassung wird folglich nur schwach von systematischen Fehlern beeinflusst.

Wie an Abbildung 5.19 deutlich zu erkennen, kann $\sigma_{LT'}$ wegen der großen statistischen Fehler insbesondere für hohe Schwerpunktsenergien nicht genau bestimmt werden. Die Asymmetrie $A_{LT'}$, dargestellt in Abbildung 5.20, stellt daher einen alternativen Zugang zur Bestimmung des helizitätsabhängigen Teils des Wirkungsquerschnitts dar. Sie erlaubt außerdem die Größe des π^+N -Aufbruchs zu bestimmen, sofern die helizitätsunabhängigen Anteile ausreichend genau bekannt sind.

Sowohl MAID als auch die chirale Störungstheorie können die gemessene Asymmetrie nicht hinreichend genau beschreiben. Bei der ChPTh ist hauptsächlich der Transversal-transversal-Anteil für die Diskrepanz zu den gemessenen Daten verantwortlich. Setzt man σ_{TT} auf null, dann weist die theoretisch berechnete Asymmetrie deutlich kleinere Unterschiede zum Experiment auf. Zum Vergleich wurden in den Histogrammen zusätzlich die Vorhersagen eines dynamischen Modells (DMT [DMT01]) mit eingetragen.

Die statistischen Fehler der Asymmetrie sind klein genug, um mit einer χ^2 -Minimierung die Größe des Aufbruchs auf 6%⁶ genau zu bestimmen. Dies ist aber erst sinnvoll, wenn die zugehörige Theorie überarbeitet wurde.

⁶Es wurde eine χ^2 -Minimierung durchgeführt, um die Güte der Daten für spätere Analysen zu kennen. Die ermittelte Zahl hat jedoch abgesehen vom statistischen Fehler keinerlei Aussagekraft, da die zugrundeliegende Theorie teilweise große Abweichung von den ermittelten Wirkungsquerschnitten aufweist.

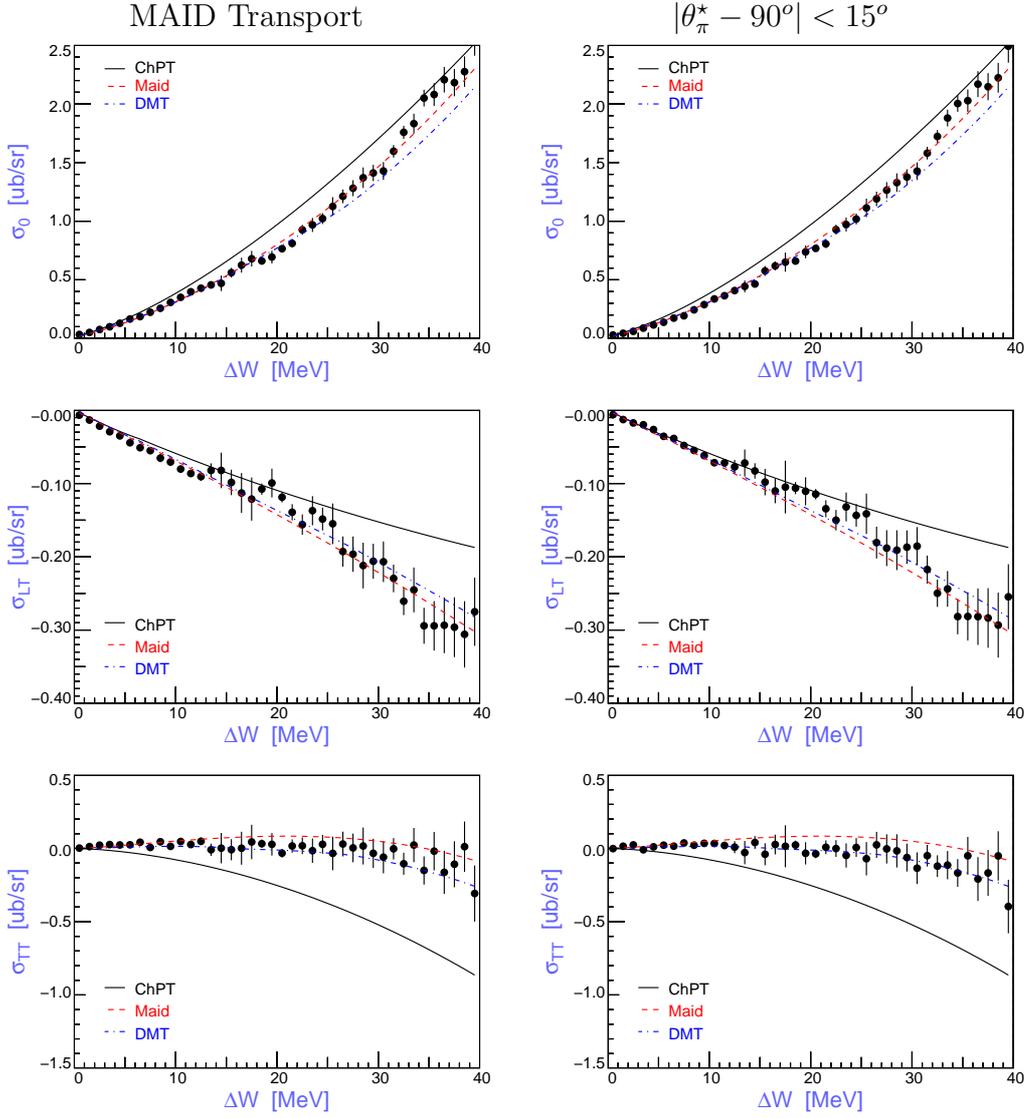


Abb. 5.18: Wirkungsquerschnitte als Funktion der Schwerpunktsenergie ΔW für die Kinematiken 0-6 bis 4-2. Die Spektren in der linken Spalte zeigen die ermittelten Wirkungsquerschnitte nach dem Transport auf die Nominalkinematik. Die rechte Spalte zeigt die Wirkungsquerschnitte bei einem Schnitt in θ_π^* .

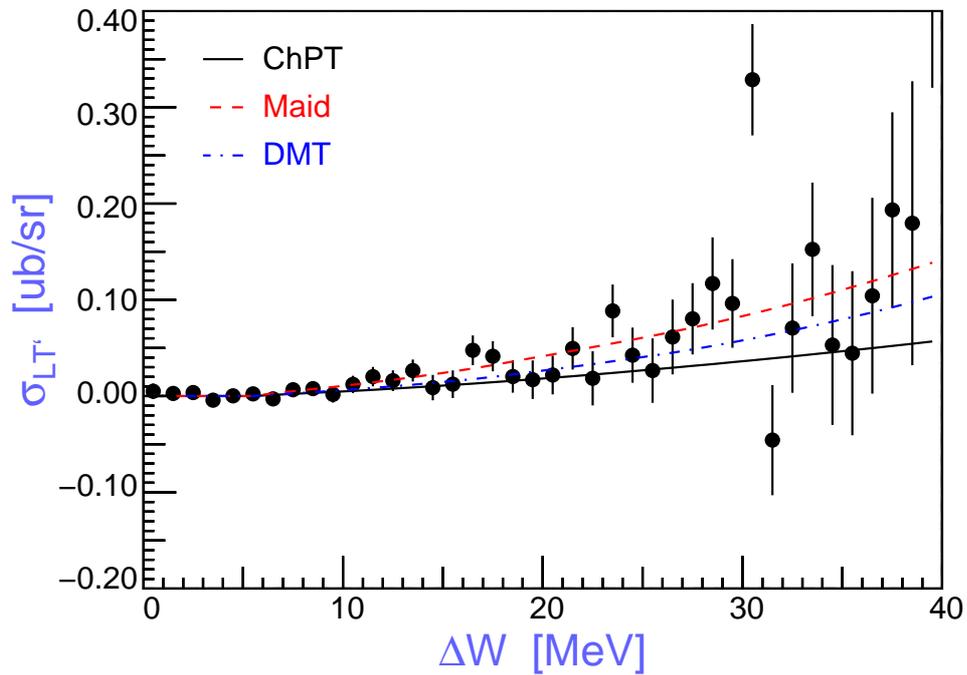


Abb. 5.19: Wirkungsquerschnitt $\sigma_{LT'}$ als Funktion der Schwerpunktsenergie ΔW .

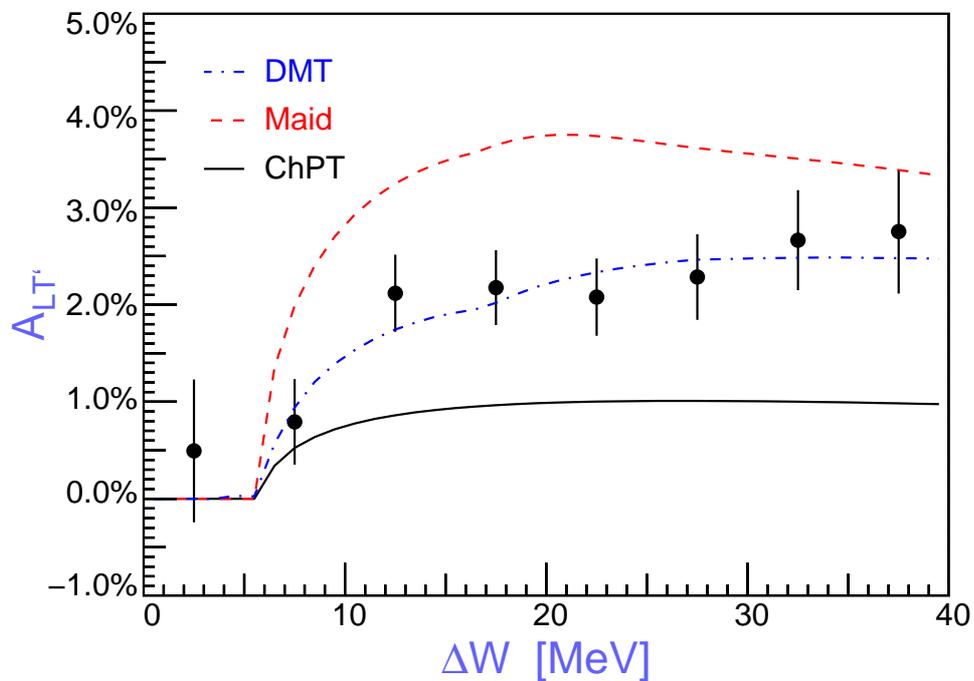


Abb. 5.20: Helizitätsasymmetrie $A_{LT'}$ als Funktion der Schwerpunktsenergie ΔW .

Kapitel 6

Steuerung

Für die Durchführung von Experimenten an der Drei-Spektrometer-Anlage werden bei der Datenerfassung, der Datenanalyse und zur Steuerung der Anlage Rechnersysteme eingesetzt.

Im Rahmen dieser Arbeit wurde das bestehende Steuerungssystem *ECS*¹ durch ein neues System *Mezzo* vollständig ersetzt. Parallel dazu erfolgte eine Trennung der bis dahin zwischen Datenerfassung und Steuerung gemeinsam genutzten Elektronik. Die Ersetzung wurde schrittweise durchgeführt, da während der Umstellungsphase Experimente stattfanden und somit die Funktionsfähigkeit beider Systeme gewährleistet bleiben mußte.

In den Abschnitten dieses Kapitels werden zunächst die grundlegenden Anforderungen an das Steuerungssystem dargelegt. Anschließend wird das bisherige System vorgestellt und seine Problemstellen aufgezeigt, gefolgt von einer detaillierten Beschreibung des neuen Steuerungssystems.

6.1 Grundlegende Begriffe in der Regeltechnik

Um eine physikalische Größe, wie z.B. Druck, Temperatur oder Spannung, auf einem gewünschten Wert zu halten, kann sie entweder gesteuert oder geregelt werden. Die Struktur des technischen Prozesses legt dabei fest, um welchen der beiden Typen es sich handelt:

Steuerung: Das Kennzeichen einer Steuerung ist ein sogenannter *offener Wirkungsablauf*. Hierbei wird der sich einstellende Wert nicht überprüft, wodurch sich Störungen durch äußere Einflüsse nicht auf den Einstellvorgang auswirken. Fast alle Größen der Spektrometeranlage fallen in diese Kategorie, da bei den meisten Geräten nur ein *Sollwert* vorgegeben wird.

Regelung: Eine Regelung ist ein *geschlossener Wirkungsablauf*. Die zu regelnde Größe (*Regelgröße, Istwert*) wird fortlaufend gemessen und mit dem vorgegebenen Wert (*Führungsgröße, Sollwert*) verglichen. Die Differenz (*Regelabweichung*) zwischen diesen beiden Größen wird dazu verwendet, in einem Verstellvorgang die *Regelgröße* mit der *Führungsgröße* in Übereinstimmung zu bringen. Ein Beispiel für eine Regelung bei der Spektrometeranlage ist die Feldkompensation an Spektrometer C.

¹Experiment Control System

Das Übertragungsverhalten der meisten Regler läßt sich auf drei Grundtypen zurückführen: den *Proportional*-, den *Integral*- und den *Differentialregler*:

a) Proportional (P-)Regler

Man spricht von einem *P-Regler*, wenn die Ausgangsgröße $u(t)$ proportional zur Regelabweichung $e(t)$ ist. Es gilt:

$$u(t) = K_P e(t) \quad (6.1)$$

Eine Erhöhung der Reglerverstärkung K_P führt zu einer Verringerung der Dämpfung bei einem *P-Regler*. Als Folge daraus darf die Verstärkung nicht zu groß werden, da der Kreis sonst instabil werden kann und zu schwingen beginnt.

b) Integral (I-)Regler

Die Wirkungsweise des *I-Reglers* läßt sich wie folgt beschreiben:

$$u(t) = K_I \int_{t_0}^{t_1} e(t) dt \quad (6.2)$$

Die Besonderheit eines *I-Reglers* ist, daß er auch ein Stellsignal $u(t)$ produzieren kann, wenn die augenblickliche Regelabweichung $e(t) = 0$ ist. Dies liegt daran, daß der Wert $u(t)$ eine Funktion der in der Vergangenheit aufgetretenen Regelabweichungen ist.

c) Differenzierender (D-)Regler

Für den *D-Regler* gilt folgender funktioneller Zusammenhang:

$$u(t) = K_D \frac{de(t)}{dt} \quad (6.3)$$

D-Regler werden eingesetzt, um die Dämpfung eines Regelkreises zu verbessern und damit Schwingungen entgegenzuwirken.

Häufig findet man Kombinationen aus den aufgeführten Reglertypen, da nur so eine Störgröße schnell kompensiert werden kann, ohne das System instabil werden zu lassen. Die Kompensation für den Kurzschluß in Dipol 1 von Spektrometer C beispielsweise enthält einen *PI-Regler*.

6.2 Anforderungen an das System

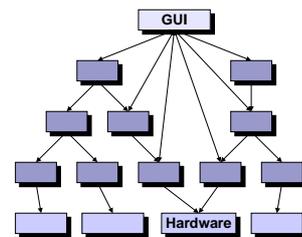
Es ist die Aufgabe des Steuerungssystems, die statischen Komponenten eines Experiments zu konfigurieren und zu überwachen. Eine Größe ist statisch, wenn sie sich während der Datennahme nicht oder nur sehr langsam ändert. Beispiele hierfür sind die Hochspannungen für die Driftkammern, die Targetposition oder

das Feld eines Magneten. Diese Definition impliziert, daß die Größen nicht bei jedem Ereignis von der Datenerfassung ausgelesen werden; man spricht daher auch von der *Slow-Control*.

Bedingt durch die räumliche Trennung und die Vielzahl der zu steuernden Komponenten ist es erforderlich, mehrere Computer einzusetzen. Diese sollen dicht bei der jeweiligen Hardware aufgestellt werden oder, wo dies aus Platzgründen nicht möglich ist, mittels eines geeigneten Bussystems mit ihr kommunizieren. Zum Zeitpunkt dieser Arbeit umfaßt die Drei-Spektrometer-Anlage etwa 2100 Komponenten, die überwacht, eingestellt oder geregelt werden müssen.

Das Steuerungssystem muß in einer geeigneten Weise die Struktur der Spektrometeranlage widerspiegeln. Durch die vorhandene physikalische Einteilung in mehrere Gerätegruppen wie beispielsweise Magnete und deren weitergehende Zerlegung in Dipole und Quadrupole liegt es nahe, eine Baumstruktur zu wählen. Eine solcher Baum kann auf mehrere Arten implementiert werden.

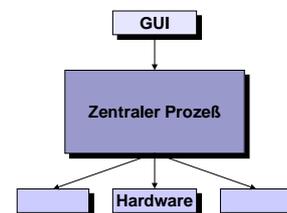
- a) Jeder Knoten und jedes Blatt des Baumes ist ein eigenständiges Programm, das mittels eines gemeinsamen Protokolls mit den übrigen Komponenten des Systems kommuniziert. Dies entspricht der Struktur des *ECS*.



Vorteile: Einzelkomponenten können leicht komplexe Regel- und Steuerungsfunktionen übernehmen und sind nicht auf eine übergeordnete Verwaltung angewiesen.

Nachteile: Beim Ausfall eines Programmes kann das gesamte System instabil werden. Der Wartungsaufwand bei Änderung grundlegender Dinge ist sehr hoch, da viele unterschiedliche Programme existieren.

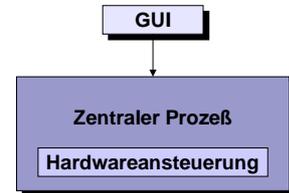
- b) Die Blätter des Baumes sind einzig für die Ansteuerung der Geräte zuständig, während die Knoten in einem gemeinsamen zentralen Prozeß verwaltet werden.



Vorteile: Das System verfügt über eine hohe Stabilität aufgrund der wenigen beteiligten Prozesse. Die Kommunikationswege sind kurz und erlauben rasche Antworten auf externe Anfragen. Zu jedem Zeitpunkt ist der Zustand des Gesamtsystems, also aller beteiligten Geräte, bekannt.

Nachteile: Die Gerätetreiber benötigen eine übergeordnete Verwaltung. Individuelle Steuer- und Regelprogramme sind nur schwer in das System zu integrieren.

- c) Der gesamte Baum wird in einem einzigen monolithischen Prozeß implementiert.



Vorteile: Es gibt nur einen einzigen „*Single Point of Failure*“. Zudem wird Interprozesskommunikation lediglich für den Datenaustausch mit den Benutzerschnittstellen benötigt.

Nachteile: Dezentrale Hardware muß durch Bereitstellung zusätzlicher Bussysteme an den zentralen Prozeß gekoppelt werden. Individuelle Steuer- und Regelprogramme sind schwer in das System zu integrieren.

In allen Fällen handelt es sich aus softwaretechnischer Sicht um ein verteiltes System, bei dem mehrere unabhängige Programme über ein gemeinsames Protokoll miteinander reden müssen. Da die Hardware der Spektrometeranlage sehr vielfältig und inhomogen ist, müssen die unterschiedlichen Geräte auf eine gemeinsame Softwareschnittstelle abgebildet werden.

Für das neue Steuerungssystem Mezzo wurde die Struktur b) gewählt. Sie stellt einen guten Kompromiß zwischen der Flexibilität vieler (dezentraler) Prozesse bei geringen Hardwarekosten durch zusätzliche Bussysteme und der Stabilität eines einzigen Programmes dar. Zusätzlich kann auf einfache Weise dafür gesorgt werden, daß Mezzo nach einem Ausfall eines Frontend-Rechners problemlos und selbständig weiterläuft.

Ausgehend von diesen groben strukturellen Festlegungen werden in den folgenden Abschnitten zunächst das alte und anschließend das neue Steuerungssystem im Detail beschrieben.

6.3 Ausgangspunkt: Das alte System (ECS)

Das ECS war das zu Beginn dieser Arbeit an der Drei-Spektrometer-Anlage verwendete Steuerungssystem. Es wurde parallel zum Aufbau und der Inbetriebnahme der Spektrometer entwickelt. Daher war eine der Grundvoraussetzungen an das ECS seine frühe Verfügbarkeit, um die neuen Detektorsysteme testen zu können. Realisiert werden konnte dies durch eine große Anzahl unabhängiger Programme, die für Testzwecke einzeln ansprechbar waren und später im Verbund die Spektrometeranlage abbildeten.

Die Grundlage der Interprozesskommunikation zwischen den Komponenten des ECS ist MUPIX² [Kra95], ein am Institut entwickeltes System zum Austausch von Nachrichten innerhalb eines LANs³. MUPIX basiert auf dem verbreiteten UDP⁴ und erweitert es um die Möglichkeit, Anwenderdaten innerhalb eines Ethernetpaketes zu verschicken.

Bei UDP handelt es sich im Gegensatz zu dem ebenfalls geläufigen TCP/IP⁵ um ein ungerichtetes Protokoll. Dies bedeutet, daß es keine feste Verbindung zwischen dem Sender und dem gewünschten Empfänger eines Paketes gibt. Um die sichere Übertragung einer Nachricht zu garantieren, wurde in MUPIX ein „*at-most-once*“ Verfahren implementiert, das dafür sorgt, daß die Nachricht mindestens einmal ihren Empfänger erreicht. Doppelte Pakete werden ausgefiltert. TCP/IP konnte nicht verwendet werden, weil beim ECS so viele Programme miteinander kommunizieren müssen, daß eine zu große Anzahl Portnummern benötigt wird.

Um die Wartbarkeit und Transparenz des Systems zu verbessern, können die Adressaten statt mit ihrer IP-Nummer auch über logische Namen angesprochen werden. Ermöglicht wird dies durch einen zusätzlichen Prozeß (genannt *namserv*), bei dem sich jeder Teilnehmer anmeldet und seinen Namen mitteilt, unter dem er bekannt sein möchte.

Die Spektrometeranlage wird durch eine hierarchische Anordnung von Geräte-Handlern und Schnittstellenumsetzern für die Hardware dargestellt. Handler sind das Gegenstück zu den physikalischen Geräten und werden für deren Steuerung und Überwachung verwendet. Innerhalb eines Handlers werden die Geräte in ihre funktionellen Untereinheiten zerlegt. Der Zustand einer solchen Einheit wird durch den Wert ihrer Zustandsvariablen beschrieben. Beispiele hierfür sind Strom und Spannung der Netzteile für die Driftkammern oder Druck und Temperatur des Targets. Auf der untersten Hierarchieebene kommunizieren die Schnittstellenumsetzer direkt mit der Hardware, beispielsweise über ein Bussystem wie VME oder die serielle Schnittstelle. Die Handler sind selbständig für die Kontrolle und Überwachung der Geräte zuständig. Liegt der Wert einer Zustandsvariablen außerhalb eines vorgegebenen zulässigen Bereichs, so wird eine Status- bzw. Fehlermeldung generiert und an die Benutzerschnittstellen weitergeleitet.

Um ein komplexes Gerät wie einen Magneten steuern zu können, werden viele Geräte-Handler eingesetzt. Damit der Experimentator den Überblick nicht verliert, gibt es einen zentralen Prozeß (genannt *ServAd*) [Mar93], der das Starten und Stoppen ganzer Teilbäume innerhalb der Hierarchie verwaltet. Unterstützt wird er von sogenannten Prozessmanagern, die auf jedem Frontend laufen und dort die benötigten Geräte-Handler starten.

²MUltiProcessor Interprocess Communication System

³Local Area Network

⁴User Datagram Protocol

⁵Transmission Control Protocol/Internet Protocol

In regelmäßigen Abständen schicken die Geräte-Handler Zustandsmeldungen an einen weiteren zentralen Prozeß, den *opcon*⁶. Dieser sammelt diese Meldungen und verteilt sie an alle Klienten, die über den Zustand eines Gerätes informiert werden wollen. Wichtige Klienten sind beispielsweise die Benutzeroberfläche und der Prozeß zum Protokollieren der Steuerungsdaten.

Initialisierungs- und Konfigurationsdaten des Steuerungssystems werden in einer Datenbank [Kun96] abgelegt. Diese wurde in Form einer Client/Server-Architektur entwickelt, damit jede Komponente des ECS darauf zugreifen kann. Sie basiert auf einfachen Textdateien innerhalb des Unix-Dateisystems. Änderungen an der Konfiguration eines Geräte-Handlers erfordern einen Neustart des entsprechenden Gerätebaumes.

Das ECS stellt zwei Programme zur Interaktion mit den Gerätehandlern zur Verfügung. Dies ist zum einen die grafische Benutzeroberfläche *Xopdisp* [Ste93] und zum anderen das textorientierte Programm *adjust*. Beide erlauben dem Experimentator, direkt den Wert der Zustandsvariablen der Untereinheiten jedes Geräts zu verändern und auszulesen.

Abbildung 6.1 zeigt einen Überblick über die beschriebenen Prozesse und ihre Verbindung untereinander am Beispiel der Magnetfeldauslese von Spektrometer B.

Das Steuerungssystem Mezzo ersetzt das vorgestellte ECS. Letzteres war durch häufige Ausfälle während Strahlzeiten negativ aufgefallen und mußte dementsprechend entweder verbessert oder ausgetauscht werden.

Ursache für die Instabilität ist die starke Abhängigkeit der vielen Einzelprozesse untereinander. Während eines Experimentes kommt es aufgrund der Strahlung in der Experimentierhalle zu unvermeidbaren Fehlfunktionen der dortigen Rechner-systeme. Als Folge dieser Fehlfunktion können einzelne Gerätetreiber oder auch das gesamte Betriebssystem auf dem Frontend terminieren. Werden aus dem ECS Komponenten entfernt, ohne daß sie sich wie vorgesehen abmelden können, funktionieren die Namensauflösung (*namserv*) und der Gerätemanager (*servad*) nicht mehr korrekt. Zusätzlich erlaubt der Manager nicht, Teile des Gerätebaumes erneut zu starten, wenn die betroffenen Handler auch von anderen Ästen benötigt werden. Alle diese Komplikationen führen dazu, daß das gesamte Steuerungssystem erneut initialisiert werden muß, auch wenn nur ein kleiner Teil davon fehlt. Da dieser Vorgang sehr zeitintensiv ist, kommt es zu den genannten großen Ausfallzeiten bis zu mehreren Stunden.

Im Rahmen der Erneuerung der Datenerfassung werden die veralteten Frontend-rechner auf Basis des M68k durch neue VME-Bus-CPU's ersetzt (Intel). Gleichzeitig sollte Linux statt einem modifizierten SunOS als Betriebssystem verwendet werden. Unterschiede bei der Implementation des UDP-Protokolls verhinderten

⁶OPerator CONtrol

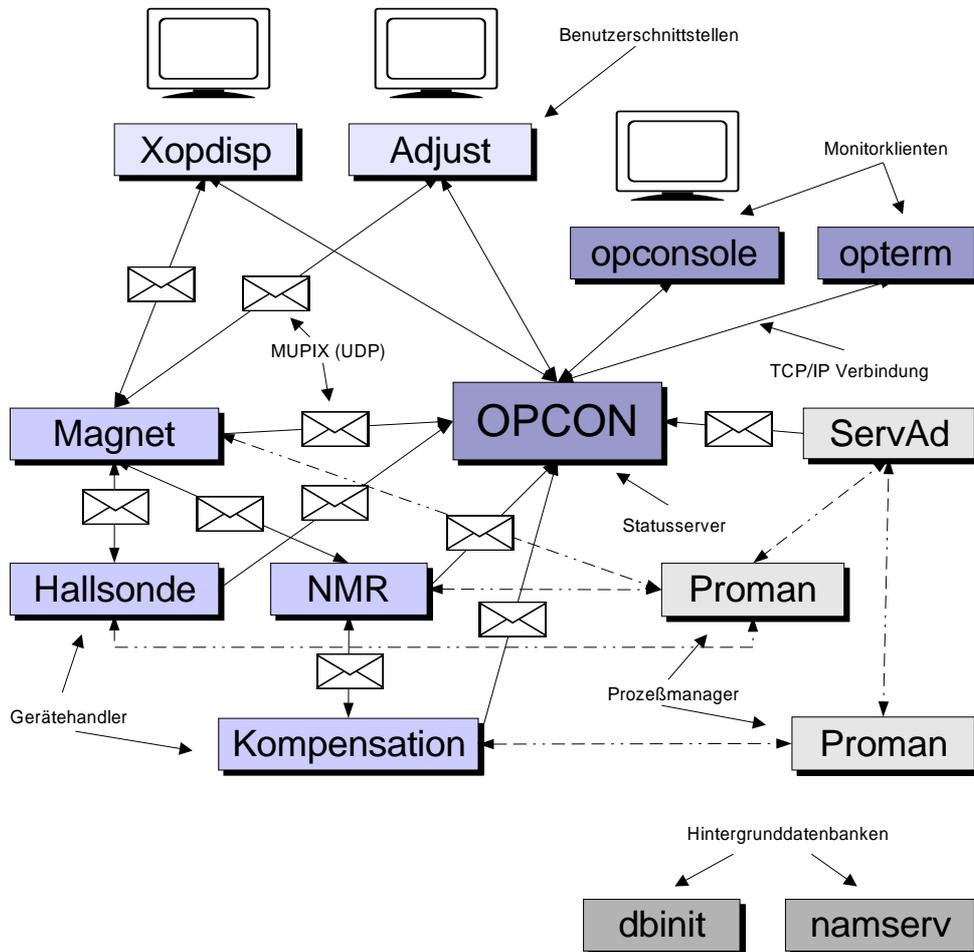


Abb. 6.1: Kommunikationswege der Komponenten im ECS. Die durchgezogenen Linien entsprechen MUPIX-Datenpaketen und die gestrichelten Steuerbefehle des Prozessmanagers.

jedoch die direkte Portierung von MUPIX nach Linux. Da außerdem zahlreiche Veränderungen (Patches) an einzelnen Programmteilen vorgenommen wurden, ohne in das Gesamtsystem integriert zu werden, lag kein vollständig kompilierbarer Quellcode mehr vor. Aus diesem Grund wurde auf die Überarbeitung des ECS verzichtet und ein neues Steuerungssystem ohne die bekannten Problemstellen entwickelt.

6.4 Das neue Steuerungssystem (Mezzo)

6.4.1 Gerätetreiber

Gerätetreiber bilden die Schnittstelle zwischen dem Steuerungssystem und der Hardware. Sie sind im eigentlichen Sinne Anwenderprogramme, da sie entweder über ein Bussystem oder über Linux-Kernelmodule auf die physikalisch vorhandenen Geräte zugreifen.

Aus Zeit- und Wartungsgründen ist es nicht praktikabel, jedes Gerät individuell in Mezzo einzubinden. Stattdessen werden die Gemeinsamkeiten aller Geräte auf wenige verschiedene Funktionseinheiten bzw. Kanäle reduziert und in *Eigenschaftslisten* (siehe 6.4.3) verpackt. Diese Kanäle entsprechen ADCs, DACs, Digital In, Digital Out oder universell verwendbaren Textkanälen. Tabelle 6.1 zeigt am Beispiel des Analog-Digital-Wandlers eine derartige Liste.

Eigenschaft	Typ	Bemerkung
ActualValue	double	letzter gelesener Wert
AlarmFlags	integer	signalisiert Alarmzustand
AlarmHigh	double	obere Alarmschwelle
AlarmLow	double	untere Alarmschwelle
AlarmState	integer	Alarmüberwachung ein/aus
AlarmType	integer	Momentary/Latching (siehe B.3)
Class	integer	Kennzahl für die Klasse
Epsilon	double	Filtergröße
Error	string	letzte Fehlermeldung
Init	integer	Klasse ist initialisiert
Polling	double	Frequenz für regelmäßige Auslese
Range	integer	Meßbereichsnummer
RangeList	range[]	Liste der verfügbaren Meßbereiche
State	integer	Kanal ist ein/aus

Tab. 6.1: Eigenschaftsfelder eines Analog-Digital-Wandlers

Die Gerätetreiber verwenden die *Eigenschaftslisten* lediglich für die gemeinsame Kommunikation mit dem zentralen Prozeß. Hardwarezugriffe sind an dieser Stelle nicht möglich, da Mezzo intern ereignisorientiert arbeitet, die Zugriffe aber unter Umständen mehrere Minuten dauern können. Eine Blockade des Systems über eine so lange Zeit ist inakzeptabel.

Das Problem der Blockade kann elegant durch die Verwendung mehrerer Programmfäden (Threads) gelöst werden: Für jedes angeschlossene Gerät wird ein Faden erzeugt, der auf die Hardware zugreift und sich dafür beliebig viel Zeit lassen kann. Solange der Zugriff dauert, ist zwar der zusätzliche Faden von außen nicht ansprechbar, die Kommunikation des Gerätetreibers mit den übrigen

Komponenten des Steuerungssystems funktioniert aber nach wie vor. Über eine gepufferte Warteschlange empfängt und verschickt der Faden Befehle asynchron nach außen.

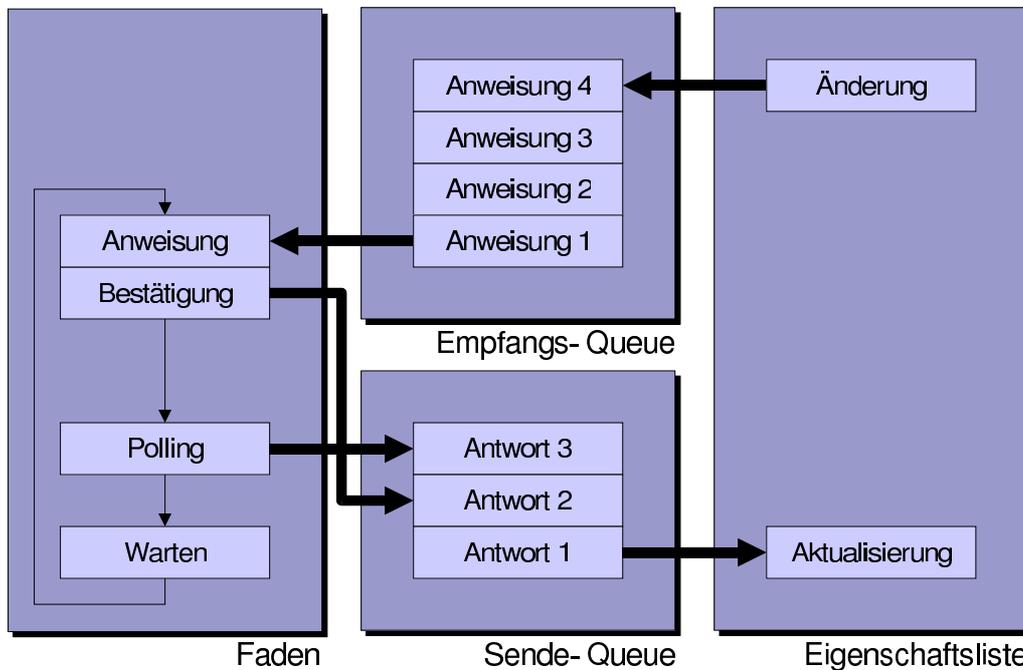


Abb. 6.2: Übersicht über die Sende- und Empfangspuffer der Gerätetreiber

Abbildung 6.2 verdeutlicht diesen Ablauf. Wird eine Eigenschaft in der Liste verändert, so fügt das System eine Anweisung in die Empfangsqueue des Fadens für den Hardwarezugriff. Dort wird periodisch mit der festgelegten Abfragefrequenz die Queue abgearbeitet und die Rückantwort in die Sendequelle eingefügt. Anschließend erfolgt die regelmäßige Auslese der Hardware. Die Eigenschaftsliste aktualisiert ihre Elemente anhand der Antworten und sendet die geänderten Werte an den zentralen Prozeß.

Die periodische Auslese der Hardware wird im Gerätetreiber angestoßen. Denkbar wäre auch, daß der zentrale Prozeß Anfragen an den Treiber schickt⁷. Diese Methode hat aber den Nachteil, daß für jeden auszulesenden Kanal zusätzliche Datenpakete verschickt werden müssen. Da der zentrale Prozeß ohnehin ein *ping*⁸ an den Gerätetreiber schickt, sind diese Pakete überflüssig.

Vom Funktionsumfang her sind die Gerätetreiber so einfach wie möglich gehalten. Alle Werte werden so, wie sie von der Hardware ausgelesen wurden, direkt an den

⁷Im ECS wurden periodische Abfragen von den übergeordneten Gerätetreibern gestellt.

⁸Im Abstand von zehn Sekunden wird ein leeres Datenpaket an den Gerätetreiber geschickt um zu testen, ob er noch darauf reagiert.

zentralen Prozeß geschickt. Erst dort finden die notwendigen Konvertierungen in physikalisch sinnvolle Größen statt.

Interbus-S

Im Laufe dieser Arbeit wurden die Frontendrechner der Spektrometer durch moderne PC-basierte Computer ersetzt. Diese Modernisierung wurde zum Zweck der Leistungssteigerung der Datenerfassung durchgeführt, hat aber auch einen entscheidenden Einfluß auf die Steuerung. Bis dahin wurden wichtige Komponenten der Spektrometeranlage (Driftkammern, Kollimatoren) über VME-Bus-ADCs, DACs und Digital-IO-Module angesteuert. Da eine Trennung gemeinsam genutzter Hardware zwischen Datenerfassung und Steuerung erfolgen sollte, mußte diese Hardware durch ein anderes Bussystem ersetzt werden. Aufgrund der Erfahrung der Elektronikabteilung des Instituts mit dem Interbus-S, wurde die Steuerungshardware schrittweise auf dieses System umgestellt.

Interbus [Jan98] ist ein Sensor-/Aktorbus-System (EN 50254) aus der Gruppe der Feldbusse mit Ringstruktur. Im Vergleich mit anderen offenen Feldbussystemen ist der Interbus mit weltweit über 37% Marktanteil der Marktführer. Primäres Einsatzgebiet des echtzeitfähigen Bussystems ist die Übertragung von Prozeßdaten. Es verbindet alle Interbus-Teilnehmer über eine sogenannte Anschaltbaugruppe mit dem Rechnersystem.

Topologisch gesehen ist es ein Ringsystem, bei dem jeder Teilnehmer zwei Hin- und Rückleitungen besitzt. Es arbeitet als ein über alle Busteilnehmer verteiltes Schieberegister, bei dem jedes Gerät Datenspeicher enthält, die über die Ringverbindung des Bussystems zu dem Schieberegister zusammengesetzt werden. In einem Datenzyklus werden die Anforderungsdaten vom Master zu den Teilnehmern und gleichzeitig die Auslesedaten zurück zum Master geschickt.

Für die Spektrometeranlage wurden mehrere Interbusmodule entwickelt: Für die Ansteuerung und Auslese der Driftkammerhochspannung, der Gasflüsse und des Vakuumsystems gibt es universell einsetzbare Multi-IO-Module. Ebenso werden die Kollimatoren und die Kompensationselektronik für die Magnetfeldauslese über den Interbus gesteuert. Schließlich wurden für die Vorverstärkernetzteile und die Schwelleneinstellung der Driftkammern spezielle Module gebaut (siehe B.4.20).

6.4.2 Zentraler Prozeß

Der zentrale Prozeß ist für die Verwaltung des gesamten Steuerungssystems zuständig: Er sammelt die ausgelesenen Werte aller Gerätetreiber, konvertiert sie in

physikalisch sinnvolle Größen und schickt sie anschließend an die Benutzeroberfläche und die Datenerfassung, wo sie in den regulären Datenstrom⁹ eingespeist werden. Zudem sorgt er dafür, daß Sollwerte und Konfigurationsparameter in einer permanenten Datenbank abgelegt werden.

Wie bereits in Kapitel 6.4.1 erwähnt, verfügen die von den Gerätetreibern verwalteten Hardwarekanäle nur über ein Minimum an Funktionalität. Die fehlenden Eigenschaften werden stattdessen von sogenannten *Aliases* im zentralen Prozeß bereitgestellt. Ein Alias ist eine *Eigenschaftsliste* mit einem frei wählbaren Namen, die fest mit einem Kanal verknüpft wird. Anhand des Namens kann der Experimentator den anonymen Hardwarekanal einer bestimmten Gerätegruppe zuordnen. Um die baumartige Struktur der Spektrometeranlage beschreiben zu können, sollte der gewählte Name einem Bezeichner der Programmiersprache C entsprechen. Beispielsweise `b.vdc.gas.argon` für den Gasfluß der Driftkammer von Spektrometer B. Obwohl der zentrale Prozeß selbst nicht auf die Syntax des Namens achtet, ist die regelgerechte Schreibweise für die Benutzeroberfläche und die Anbindung an die Analysesoftware wichtig¹⁰.

Tabelle 6.2 zeigt die *Eigenschaftsfelder* eines Alias. Wie deutlich zu sehen ist, gibt es Felder für jede Art von Hardwarekanälen. Je nachdem, mit welchem Typ von Kanal der Alias verbunden ist, werden einige der Felder nicht genutzt. Dies entspricht zwar nicht einem reinen objektorientierten Konzept, bei dem ein angepaßtes Aliasobjekt erzeugt werden würde, hat aber entscheidende Vorteile beim Zugriff auf die Datenbank, bei der Benutzeroberfläche und der Verwaltung von Fehlerzuständen:

- Es ist nicht erforderlich, daß eine relationale Datenbank eingesetzt wird. Stattdessen wird für jeden Alias nur eine einzige Tabelle angelegt, in die alle Änderungen an den Sollwerten sowie der Konfiguration eingetragen werden. Unterstützt wird dies durch die ereignisorientierte Grundstruktur des Steuerungssystems, bei der das System ohnehin nur auf Modifikationen an den Eigenschaftslisten reagiert.
- Die Benutzeroberfläche muß sich nur um die Darstellung eines einzigen Typs von Eigenschaftslisten kümmern. Ob konkret der Sollwert eines DACs oder der aktuelle Auslesewert eines ADCs angezeigt werden soll, ist leicht feststellbar, da die Eigenschaftsfelder leer sind, wenn sie nicht benutzt werden.
- Da ein Alias fest mit einem Kanal verbunden wird, wäre es unvorteilhaft, wenn aufgrund einer Fehlbedienung beispielsweise ein ADC mit dem Alias für einen DAC verknüpft wird. Der Verzicht auf spezialisierte Aliase erhöht somit die Stabilität des Gesamtsystems und reduziert den Programmieraufwand.

⁹Der reguläre Datenstrom wird von der Datenerfassung AQUA erzeugt und enthält die Experimentdaten der Spektrometer und eventuell zusätzlicher Detektoren.

¹⁰Die Benutzeroberfläche kann nur bei Einhaltung der korrekten Schreibweise die Baumstruktur rekonstruieren und anschließend visualisieren.

Eigenschaft	Bemerkung	ADC	DAC	DI	DO
ActualValue	letzter gelesener Wert	✓		✓	
AlarmHigh	obere Alarmschwelle	✓			
AlarmLow	untere Alarmschwelle	✓			
AlarmState	Alarmüberwachung ein/aus	✓			
AlarmType	Momentary/Latching	✓			
AutoTrimLevel	automatischer Abgleich	✓	✓	✓	✓
BitCount	Anzahl gültiger Bits			✓	✓
Channel	Name des zugehörigen Kanals	✓	✓	✓	✓
Class	Kennzahl für die Klasse	✓	✓	✓	✓
Delta	erlaubte Regelabweichung	✓			
DemandValue	Sollwert		✓		✓
Digits	Anzahl Nachkommastellen	✓	✓		
Epsilon	Filtergröße	✓	✓		
Error	letzte Fehlermeldung	✓	✓	✓	✓
Expression	Umrechnungsformel für Rohdaten	✓	✓		
FatalFlags	Maske für fatale Fehlerzustände	✓	✓	✓	✓
Flags	aktueller Fehlerzustand	✓	✓	✓	✓
Label	Liste für vorgegebene Sollwerte		✓		✓
LimitHigh	oberes Limit	✓	✓		
LimitLow	unteres Limit	✓	✓		
Link	Verknüpfung mit anderem Alias	✓	✓		
Mask	Bitmuster für Fehlerzustand			✓	✓
Polling	Frequenz für regelmäßige Auslese	✓	✓	✓	✓
Range	Meßbereichsnummer	✓	✓		
SeriousFlags	Maske für ernste Fehlerzustände	✓	✓	✓	✓
SetPoint	zuletzt ausgelesene Stellgröße		✓		✓
StartBit	erstes Bit eines DIO-Kanals			✓	✓
State	Kanal ist ein/aus	✓	✓	✓	✓
Unit	Maßeinheit	✓	✓		

Tab. 6.2: Eigenschaftsfelder eines Alias und für welchen Kanaltyp sie verwendet werden.

Bit	Flagge	Bemerkung
0	AF_NOTCONNECTED	nicht mit Gerätetreiber verbunden
1	AF_DISABLED	Kanal ist abgeschaltet
2	AF_NOTINITIALIZED	Kanal ist nicht initialisiert
3	AF_DBFAILED	Datenbankfehler
4	AF_ACCESSFAILED	Hardwarefehler
5	AF_LIMITLOW	untere physikalische Grenze unterschritten
6	AF_LIMITHIGH	obere physikalische Grenze überschritten
7	AF_ALARMLOW	untere Alarmschwelle
8	AF_ALARMHIGH	obere Alarmschwelle
9	AF_COMPAREFAILED	Vergleich fehlgeschlagen
10	AF_SETPOINT	Abweichung bei Stellwert
11	AF_ACTUALVALUE	Abweichung bei Istwert

Tab. 6.3: Der augenblickliche Zustand eines Alias wird durch ein Bitmuster repräsentiert. Ein gesetztes Bit weist auf einen Fehler hin.

Eine der wichtigsten Aufgaben des Steuerungssystems ist die Überwachung der Spektrometeranlage. Es muß geprüft werden, ob die benötigten Gerätetreiber funktionieren, die Datenbank korrekt arbeitet und die ausgelesenen Werte innerhalb vorgegebener Grenzen liegen. Dazu wird für jeden Alias ein Eigenschaftsfeld mit Flaggen zur Verfügung gestellt, bei dem eine gesetzte Flagge einen Fehlerzustand signalisiert. Tabelle 6.3 listet die bereitgestellten Flaggen auf. Sie werden bei jeder Änderung der Eigenschaftsliste aktualisiert. Eine besondere Bedeutung haben die Flaggen AF_NOTINITIALIZED, AF_ALARMLOW und AF_ALARMHIGH. Im Gegensatz zu den übrigen Flaggen werden diese Zustände von den Hardwarekanälen bzw. Gerätetreibern selbst überwacht. Dies ist notwendig, da einige Hardwarekomponenten der Spektrometeranlage selbständig die Überschreitung von Alarmschwellen verwalten.

Die Entscheidung, ob eine gesetzte Flagge einen ernsten oder fatalen Fehlerzustand signalisiert, erfolgt mit Hilfe der beiden Felder *SeriousFlags* und *FatalFlags*. Dort wird jeweils ein Bitmuster abgelegt, welches bei jeder Modifikation der aktuellen Flaggen mit diesen verglichen werden soll¹¹. Am Ergebnis dieser Verknüpfung ist zu erkennen, welcher Fehlertyp vorliegt.

Nach dem Ausfall eines Frontends und dessen Neustart sorgt der zentrale Prozeß selbstständig für einen Wiederaufbau der Kommunikation mit dem entsprechenden Gerätetreiber und aktualisiert nötigenfalls die Sollwerte der Hardwarekanäle. Aus diesem Grund ist der Verbindungsaufbau zwischen einem Alias und seinem zugehörigen Kanal ein relativ komplexer Vorgang, der in mehreren Schritten abläuft.

¹¹Dieser Vergleich findet beispielsweise in der Benutzeroberfläche statt.

In regelmäßigen Zeitabständen überprüft der zentrale Prozeß jeden Alias, ob er aktiviert aber noch nicht mit dem Gerätetreiber verbunden ist, d.h. die Flagge *AF_DISABLED* ist null und gleichzeitig die Flagge *AF_NOTCONNECTED* eins. Wenn dies der Fall ist, wird anhand des Kanalnamens der entsprechende Gerätetreiber ermittelt und eine Verbindung mit ihm hergestellt. Der Name eines Kanals hat immer die Form *portnummer@ipnummer:bezeichner*.

Unmittelbar nach dem Verbindungsaufbau fordert der zentrale Prozess eine Liste aller Eigenschaftslisten des Gerätetreibers an und erzeugt eine lokale Kopie davon. Zu jedem physikalisch vorhandenen Kanal des Gerätetreibers gibt es nun ein virtuelles Duplikat im zentralen Prozess. Jede Änderung im Duplikat führt zu einer Aktualisierung des Hardwarekanals und umgekehrt. Der Vorteil liegt darin, daß der zentrale Prozeß nun nicht mehr Anfragen an den Kanal über das langsame Ethernet stellen muß sondern direkt darauf zugreifen kann. Während die Kopie erstellt wird, bekommt die Verbindung den Status *SYNCHRONIZING* und danach *ONLINE*. Ändert sich der Wert eines Eigenschaftsfeldes im Gerätetreiber sorgt er automatisch für einen Abgleich bei der Kopie.

Level	Bemerkung
0	Nichts korrigieren
1	Kanal und Polling aktivieren
2	Konfigurationsdaten abgleichen
3	Sollwert einstellen

Abb. 6.3: Einstellmöglichkeiten für den automatischen Abgleich zwischen Datenbank und Hardwarekanal nach einem Verbindungsaufbau

Im nächsten Schritt werden die in der Datenbank gespeicherten Konfigurationsdaten und Sollwerte für den Alias mit den tatsächlichen Werten vom Kanal verglichen. Abweichungen können vom zentralen Prozess automatisch korrigiert werden. Zu diesem Zweck gibt es das Eigenschaftsfeld *AutoTrimLevel*, welches den erlaubten Umfang der Korrekturen vorgibt. Tabelle 6.3 zeigt die möglichen Einstellungen und ihre Bedeutung.

In der Regel müssen die ausgelesenen Rohdaten noch in eine physikalisch sinnvolle Größe konvertiert werden. Die Daten liegen meist in Form eines Spannungswertes vor, der über ein Polynom in den gewünschten Wert umgerechnet wird. Jeder Alias verfügt über das Feld *Expression*, in das ein frei definierbarer Ausdruck eingetragen und zur Laufzeit des Programmes ausgewertet wird.

Abbildung 6.4 verdeutlicht die erläuterte Struktur des zentralen Prozesses und die Verbindung zu den Hardwarekanälen und den Benutzerschnittstellen.

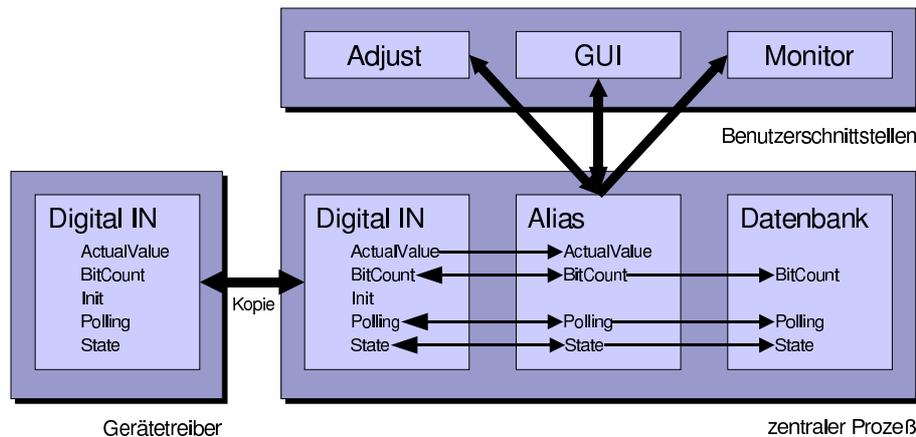


Abb. 6.4: Struktur des zentralen Prozesses am Beispiel eines „Digital-IN“

6.4.3 Kommunikation

Kommunikation und Datenaustausch aller beteiligten Komponenten erfolgt auf der Basis des TCP-Protokolls. Es ermöglicht eine zuverlässige Verbindung zwischen zwei Endpunkten und garantiert, daß alle verschickten Datenpakete in der richtigen Reihenfolge ankommen. Im Gegensatz zu MUIPIX (ECS) ist demnach kein zusätzlicher Programmieraufwand für die Zustellung von Paketen notwendig und wegen der geringen Anzahl beteiligter Prozesse kann auf das ungerichtete UDP-Protokoll gänzlich verzichtet werden.

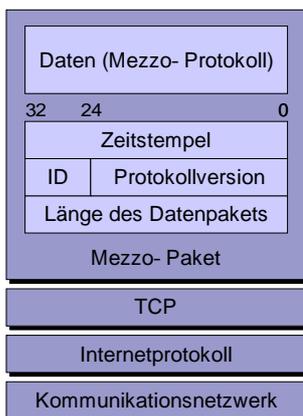


Abb. 6.5: Datenpaket

Auf den TCP-Paketen setzt das Mezzo-interne Protokoll auf. Dabei gibt es zwei grundsätzliche Übertragungsmodi: ein binärer und ein textorientierter (ASCII) Modus.

Beide Modi sind inhaltlich identisch, der binäre Modus besitzt zusätzlich einen eigenen Header und versendet die Daten in einem maschinenunabhängigen Format¹². Abbildung 6.5 zeigt die Struktur eines solchen Paketes. Der Header besteht aus drei 32bit-Worten. Zuerst kommt die Länge des gesamten Datenpaketes, danach eine Kennung bestehend aus einer 8bit-Identifikationsnummer und der Versionsnummer für das interne Protokoll (24bit) und schließlich ein Zeitstempel. Anhand der Identifikationsnummer kann unterschieden werden, ob es sich um ein Mezzopakete oder um ein Paket von der Datenerfassung handelt. Auf diese Weise können

¹²Die Kodierung in das maschinenunabhängige Format erfolgt nach RFC1014 (XDR External Data Representation)

die Steuerungsdaten in den regulären Datenstrom eingespeist und archiviert werden.

Der binäre Übertragungsmodus ist schneller als der Textmodus, weil zeitaufwendige Konvertierungen nach ASCII und zurück entfallen. Einige Komponenten wie die Benutzeroberfläche können aus Implementationsgründen nur den Textmodus verwenden (siehe 6.4.5). In der Regel erfolgt die Kommunikation jedoch binär.

In regelmäßigen Abständen verschickt der zentrale Prozeß ein *ping* an alle Klienten, um feststellen zu können, ob sie noch auf Anfragen reagieren. Dieses *ping* ist ein datenloses Mezzopaket, auf das die Gegenseite ebenfalls mit einem datenlosen Paket antwortet. Abbildung 6.6 zeigt die dabei typischen Antwortzeiten eines Steuerungsrechners.

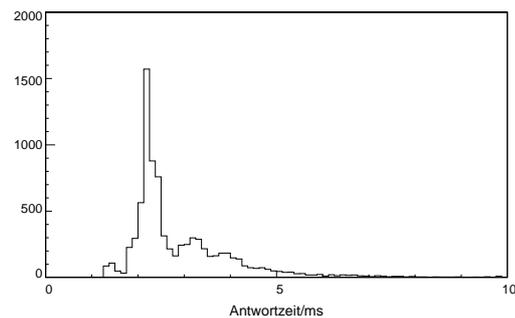


Abb. 6.6: Typische Antwortzeiten eines Klienten (a1icontrol4) während einer Strahlzeit (aufgenommen am 18.12.2002)

Unabhängig von Syntax und Semantik des Protokolls werden auf der untersten Übertragungsebene primitive Datentypen verschickt. Bei diesen Typen handelt es sich beispielsweise um Gleitkommazahlen oder Bitmuster.

Um die Programmierung zu erleichtern und übersichtlicher zu gestalten, werden diese verschiedenen Datentypen in *C++*-Klassen verpackt, die die Konvertierung in den entsprechenden Übertragungsmodus automatisch übernehmen. Abgeleitet werden sie von einer abstrakten Basisklasse *Unknown*, deren Methoden die Kindklassen implementieren. Abgesehen von der Umwandlung der Daten stellt sie auch Funktionen zur Verfügung, um Werte zuzuweisen und untereinander zu vergleichen. Übergeordnete Verwaltungsstrukturen können auf diese Weise mit jedem beliebigen Datentyp umgehen, sofern sie nur auf die Funktionalität dieser Basisklassen zurückgreifen.

Bei den verschickten Daten ist zunächst nicht erkennbar, ob es sich beispielsweise um die ausgelesene Hochspannung für einen Photomultiplier oder den Sollwert für den Strom in einem Dipolmagneten handelt. Man kann dieses Problem entweder dadurch lösen, daß man für jeden Wert einen gesonderten Befehl zum Einstellen bzw. Auslesen reserviert, oder man kennzeichnet ihn mit einem eindeutigen Namen. Letztere Variante hat zwar den größeren Verwaltungsaufwand als die Befehlsvariante, kann dafür aber schneller um zusätzliche Werte erweitert werden.

Aus diesem Grund werden Größen, die die Eigenschaft eines Objektes beschreiben, immer mit einem Namen gekennzeichnet und alle Eigenschaften eines Ob-

jekt es wiederum zu einer sogenannten *Eigenschaftsliste* zusammengefaßt. Wird ein Wert stets mit dem Namen der Eigenschaft und dem Namen der zugehörigen Eigenschaftsliste verschickt, so ist seine Herkunft eindeutig feststellbar.

Mezzo verwaltet alle Objekte in solchen Eigenschaftslisten, unabhängig ob es sich bei besagtem Objekt um einen ADC-Kanal oder die Schnittstelle zur Datenbank handelt. Das Übertragungsprotokoll wird dadurch sehr einfach, da abgesehen von Steuerbefehlen nur die Möglichkeit zur Manipulation einer beliebigen Eigenschaft implementiert werden muß. Gültig ist sie für alle Eigenschaftslisten des Steuerungssystems (siehe Anhang B.3).

Abbildung 6.7 zeigt die Grundstruktur des Protokolls. Jedes Paket kann aus einer beliebigen Zahl von Anweisungen bestehen. Es beginnt mit einem Integerwert, der die Anzahl der folgenden Befehle festlegt. Danach kommt die Anweisungsliste, zusammengesetzt aus Pärchen von Befehlskennung (Integer) und Argumenten (variabel). Die vorhandenen Steuerbefehle werden im Anhang B.1 erläutert.

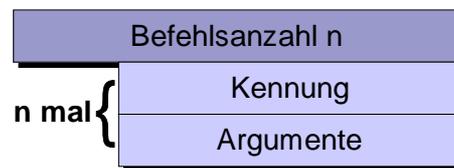


Abb. 6.7: Mezzo-Protokoll

Eine Hierarchie von Klassen sorgt dafür, daß das Übertragungsprotokoll und der Zugriff auf die Eigenschaftslisten verborgen bleibt. Ermöglicht wird dies durch eine *ereignisorientierte interne Struktur des Steuerungssystems*: Die Eigenschaftslisten müssen nicht aktiv auf Manipulationen ihrer Eigenschaftsfelder achten, beispielsweise durch einen Pollingmechanismus, sondern bekommen automatisch jede Änderung mitgeteilt. Konkret bedeutet dies, daß eine Funktion einer Klasse aufgerufen wird, sobald ein Eigenschaftsfeld modifiziert wurde.

6.4.4 Ablaufsteuerung

Der zentrale Prozeß steuert und überwacht die Zugriffe auf einzelne „unabhängige“ Kanäle. Diese Zugriffe zeichnen sich dadurch aus, daß sie allesamt *atomar* sind, also mit einer einzigen Aktion erledigt werden können. Es ist auf diesem Weg nicht vorgesehen, komplexe Steuer- oder Regelungsabläufe durchzuführen, bei denen mehrere Zugriffe auf das gleiche Gerät erforderlich sind.

Das Hilfsprogramm *Adjust* erlaubt es, einzelne Aktionen wie das Setzen und Auslesen von Sollwerten und Stellgrößen von der Kommandozeile (*shell*) aus zu starten. Auf diesem Weg können die meisten Einstellvorgänge automatisch abgearbeitet werden mit dem Vorteil, daß die *shell* ein bekanntes Werkzeug ist, welches vom Experimentator ohne Zusatzwissen bedient werden kann. Nachteil dieser Methode ist, daß die ereignisorientierte Grundstruktur des Steuerungssystems nicht auf

die Kommandozeilenumgebung übertragen werden kann. Um beispielsweise zu erkennen, wann sich eine Größe ändert, muß das entsprechende Eigenschaftsfeld regelmäßig abgefragt werden. Dies führt zu einem enormen Aufwand, da jedesmal ein Programm gestartet wird.

Aus diesem Grund wurde für komplexere Aufgaben ein Programm entwickelt, das eine eigene Umgebung für Steuer- und Regelungsaufgaben bereitstellt. Innerhalb dieser Umgebung können beliebige Zugriffe ausgeführt und auf Änderungen eines Eigenschaftsobjektes reagiert werden. Das Programm arbeitet nach dem Schema, daß beim Programmstart das Skript mit den auszuführenden Aktionen zunächst kompiliert und in einen ausführbaren Code umgewandelt wird. Anschließend wird eine Kopie aller Objekte des zentralen Prozesses angefordert und parallel dazu der Code abgearbeitet.

Der Compiler für die Skripte wurde mit den Werkzeugen Flex und Bison¹³ erstellt. Bison zerlegt die Anweisungen im Skript in eine baumartige Struktur, die dann in eine Liste sequentieller und indizierbarer Befehle konvertiert wird. Die Indizierung ermöglicht die Verwendung von Sprungbefehlen im Programmcode, welche für bedingte Anweisungen und Programmschleifen benötigt werden. Die Befehle des auszuführenden Codes bestehen aus einer Kennung für den Befehl sowie einem optionalen Argument. Die folgenden beiden Programmlistings zeigen ein Beispielskript (6.8) und den daraus resultierenden Programmcode (6.9). Eine ausführliche Dokumentation der Skriptsprache steht in Anhang B.5.

```
1 // Variablendeklaration
2 var i;
3
4 // In einer Schleife werden Hochspannungen ausgedruckt
5 for ( i = 0; i < 10; i = i + 1 ) {
6     printf( "HV[%d] = %dV\n", i, a.trigger.hv[i].get );
7 }
```

Abb. 6.8: Beispielskript für die Auslese der Hochspannungen an den Photomultipliern von Spektrometer A

Um zu verhindern, daß mehrere Skripte gleichzeitig auf einen Hardwarekanal zugreifen, stellt der zentrale Prozeß Synchronisationsobjekte zur Verfügung. Diese sogenannten Semaphoren erlauben die Implementation eines Verriegelungsmechanismus, mit dem kritische Programmabläufe vor einem mehrfachen Zugriff geschützt werden können.

Der Befehl *lock* sperrt die angegebene Semaphore. Weitere Aufrufe dieser Funktion blockieren solange, bis das ursprüngliche Programm mit dem Befehl *unlock* die Semaphore freigibt.

¹³Die Programme Flex/Lex (LEXical analyser) und Bison/Yacc (Yet Another Compiler Compiler) sind Standardwerkzeuge für den Kompilerbau.

<pre> 1 push "i" 2 var l 3 push "i" 4 push 0.0000 5 pop 6 label 0: push "i" 7 push 10.0000 8 cml 9 jz l 10 begin 11 push "a" 12 push "trigger" 13 struct 14 push "hv" 15 push "i" 16 array </pre>	<pre> 17 struct 18 push "get" 19 struct 20 push "i" 21 push "HV[%d] = %dV" 22 push 3.0000 23 print 24 end 25 push "i" 26 push "i" 27 push 1.0000 28 add 29 pop 30 jmp 0 31 label 1: </pre>
---	--

Abb. 6.9: Bytecode des kompilierten Beispielskriptes

```

// Semaphore sperren
lock( sem_test );

// Kritischer Programmabschnitt
...

// Semaphore freigeben
unlock( sem_test );

```

Abb. 6.10: Beispielcode für die Verwendung von Semaphoren

6.4.5 Grafische Benutzerschnittstelle

Für das Steuerungssystem wurde eine grafische Benutzerschnittstelle entwickelt und implementiert. Sie ermöglicht dem Experimentator zum einen, auf einfache Weise die Eigenschaftslisten des zentralen Prozesses einzusehen, zu verwalten und neue Aliase anzulegen. Ihre zweite Aufgabe besteht darin, alle für das Experiment wichtigen Größen übersichtlich darzustellen und etwaige Fehlerzustände kenntlich zu machen.

Um zumindest beim Benutzerinterface möglichst plattformunabhängig zu sein, wurde dort die Programmiersprache Java verwendet. Auch wenn die Java-Implementation unter Linux noch einige Mängel bei der Unterstützung multimedialer Anwendungen hat, ist sie für die Benutzung im Rahmen des Steuerungssystems ausreichend ausgereift.

Im Gegensatz zu den übrigen Komponenten kommuniziert die grafische Benutzerschnittstelle über das ASCII-Interface mit dem zentralen Prozeß. Grund hierfür ist der Umstand, daß in Java binäre Daten und Zeiger nur mit großem Aufwand manipuliert werden können. Insbesondere der Header des Mezzo- und Aqua-Protokolls ist problematisch, weil dort die binäre Länge des gesamten Datenpaketes bekannt sein muß. Da vom zentralen Prozeß nur gefilterte Daten weiter-

gereicht werden, ist die Verarbeitungsgeschwindigkeit der ASCII-Übertragung in Java schnell genug.

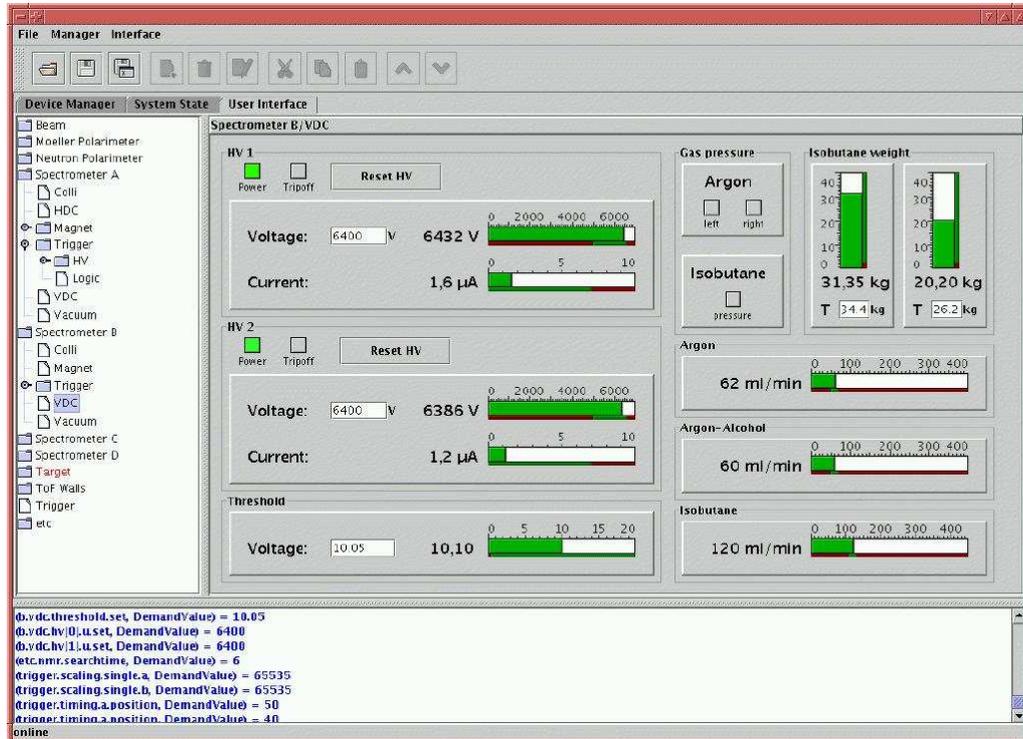


Abb. 6.11: Das Bildschirmfoto zeigt die Schnittstelle für die Driftkammer von Spektrometer B. Die grafische Benutzeroberfläche ist mittels eines eingebauten Editors frei konfigurierbar. Der abgebildete Teil zeigt nur die für den Betrieb der Driftkammer wichtigen Parameter.

Die Abbildungen 6.11 und 6.12 zeigen die grafische Benutzeroberfläche im Einsatz.

6.4.6 Magnetfeldeinstellung der Spektrometer

Die zuverlässige und reproduzierbare Einstellung der Magnete an der Drei-Spektrometer-Anlage ist für Präzisionsexperimente unverzichtbar. Aus diesem Grund wurden bereits bei der Entwicklung des alten Steuerungssystems (ECS) Algorithmen zur Magnetfeldeinstellung [Kra95, Kun96] entwickelt und implementiert. Im Rahmen dieser Arbeit wurden diese Verfahren auf das neue System (Mezzo) übertragen, Fehler beseitigt und der Einstellvorgang optimiert.

Die Magnete werden mit Netzgeräten des Typs MPS8000 der Firma Danfysik betrieben. Nach Vorgabe eines Sollwertes¹⁴ für den Strom stellen sie diesen

¹⁴Der Strom wird prozentual zum Nominalwert des Netzgerätes angegeben.

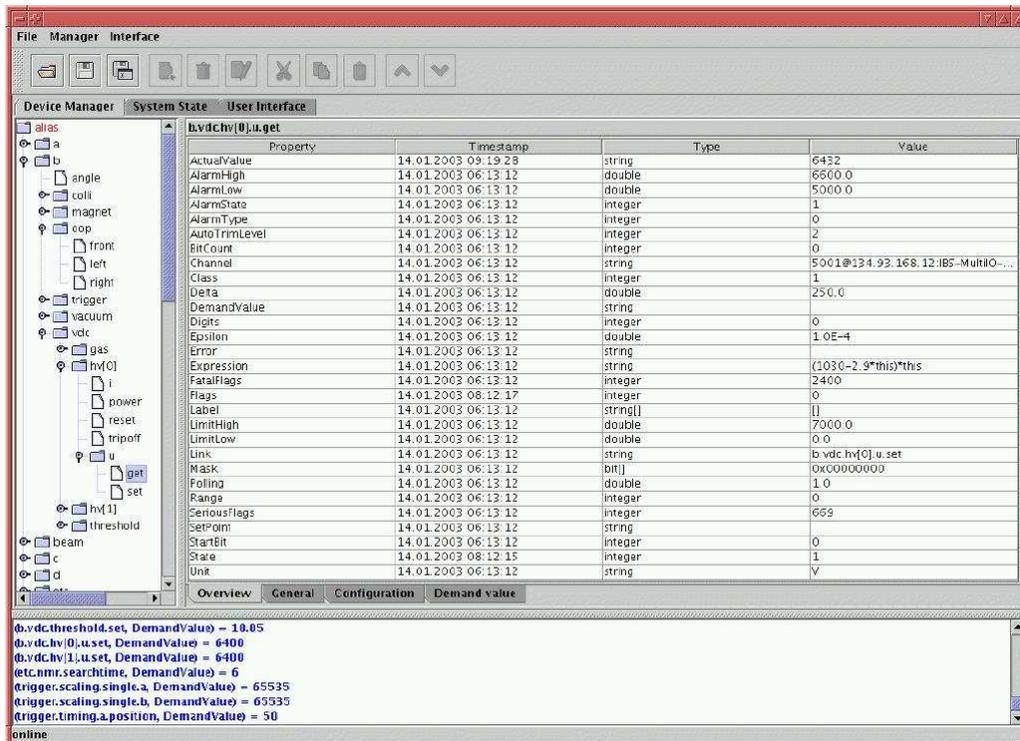


Abb. 6.12: Über den „Device Manager“ werden die Aliase im zentralen Prozeß konfiguriert. Das Bildschirmfoto zeigt die Eigenschaftsliste für die Auslese einer der beiden Hochspannungen der Driftkammer von Spektrometer B.

selbständig ein und korrigieren alle Regelabweichungen durch Messen der Regelgröße.

Für alle Spektrometer gibt es Vorschriften, um vom gewünschten Zentralimpuls ausgehend den erforderlichen Strom jedes Magneten¹⁵ zu ermitteln:

- Umrechnung des gewünschten Zentralimpulses über ein Polynom in den Feldwert auf der Zentralbahn des Spektrometers
- Korrektur des zentralen Feldwertes auf die Position der verwendeten NMR-Sonde¹⁶ bei den Dipol-Magneten. Der so ermittelte Feldwert ist die Führungsgröße bei der Prozedur zur Feldeinstellung.
- Umrechnung des Feldwertes über ein Polynom in einen Strom für das Magnetnetzteil

¹⁵Bei den Spektrometern A und C gilt der Dipol 1 als Referenzmagnet: Dipol 2, der Quadrupol und der Sextupol werden relativ zu Dipol 1 eingestellt

¹⁶Die NMR-Sonden sitzen nicht direkt auf der Zentralbahn, um kein Teilchen auf seiner Flugbahn zu beeinträchtigen.

Um Wirbelströme im Magneten zu vermeiden, darf das Magnetfeld nicht zu schnell variiert werden. Daher werden Änderungen am Netzgerät in Form einer Rampe durchgeführt, bei der der Strom in großen zeitlichen Abständen um kleine Beträge geändert wird. Hinzu kommt, daß aufgrund der Remanenz eines Magneten der Sollwert stets „von oben“¹⁷ eingestellt werden muß.

Falls der Magnet in einem undefinierten Zustand ist, muß er mittels einer sogenannten „bigloop“ in einen bekannten Zustand gebracht werden. Hierbei wird der Magnet für einen längeren Zeitraum in Sättigung gebracht und anschließend das Feld langsam auf null gefahren.

Das Polynom zur Umrechnung eines Feldwertes in den resultierenden Strom eines Magneten ist nicht exakt bekannt. Vielmehr wird es als Startwert für einen *PI-Regler* benutzt, der anhand des mit einer NMR-Sonde gemessenen tatsächlichen Feldes die Regelabweichung zur Führungsgröße bestimmt und kleine Korrekturen am Strom vornimmt. Wenn die Regelabweichung ausreichend klein ist, wird dieser Regler beendet und der Magnet hat seinen endgültigen Feldwert erreicht.

Abbildung 6.13 veranschaulicht die Prozedur zur Magnetfeldeinstellung am Beispiel eines Dipols. Die Felder von Quadrupolen und Sextupolen sind so klein, daß das Umrechnungspolynom von Feldwert zu Strom als ausreichend genau angenommen werden kann.

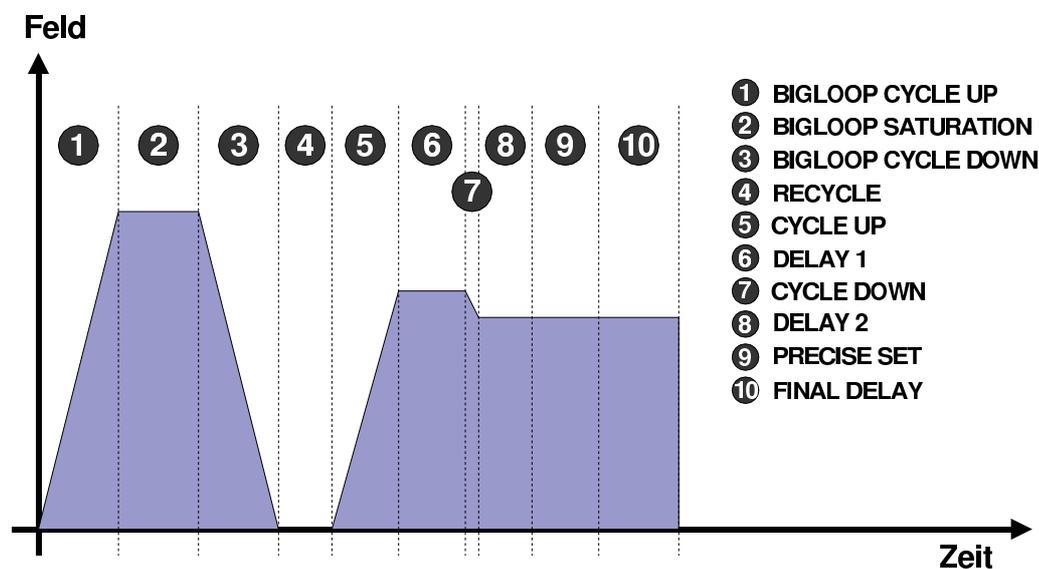


Abb. 6.13: Prozedur zum Einstellen des Feldes eines Magneten

Eine gesonderte Stellung nimmt der Dipol 1 von Spektrometer C ein. Ein Kurzschluß zwischen zwei Windungen mit zeitlich variablem Widerstand führt zu so starken Schwankungen des Magnetfeldes, daß ein zusätzlicher Regelkreis [Jen03]

¹⁷Die Steigung der Rampe ist negativ.

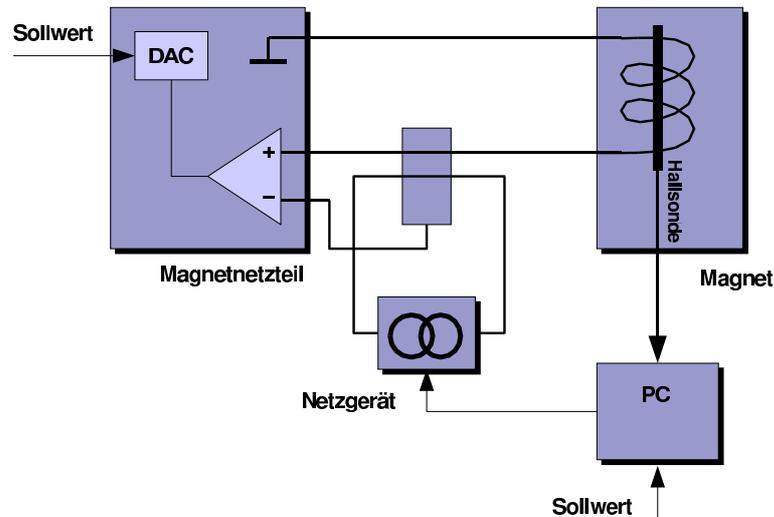


Abb. 6.14: Schematischer Aufbau der Kompensationselektronik für den Kurzschluß in Dipol 1 von Spektrometer C

zur Kompensation der Feldschwankungen nötig ist. Diese Regelung arbeitet nach folgendem Prinzip: Über eine Hallsonde wird das augenblickliche Feld gemessen und mit dem Referenzwert verglichen. Ein *PI-Regler* in einem PC steuert anhand der gemessenen Regelabweichung ein Netzgerät an, das auf den internen Regelkreis des zugehörigen Danfysik-Netztes (MPS8000) eine Stromschwankung proportional zur Feldschwankung aufbringt. Daraufhin korrigiert das MPS8000 den Strom und gleicht damit die Feldschwankungen aus. Die Regelung funktioniert jedoch nur für relativ langsame Feldschwankungen und arbeitet nur mit der Genauigkeit und Auflösung der verwendeten Hallsonde. Abbildung 6.14 zeigt schematisch den Aufbau der Kurzschlußkompensation.

Kapitel 7

Zusammenfassung

Die vorliegende Arbeit beschreibt ein Experiment vom Typ $H(\vec{e}, e'p)\pi^0$ zur Elektroproduktion neutraler Pionen an der Schwelle bei einem Viererimpulsübertrag von $Q^2 = 0.05 (\text{GeV}/c)^2$. Es wurde mit der Drei-Spektrometer-Anlage am Mainzer Mikrotron MAMI durchgeführt. Die Strahlenergie betrug 855 MeV bei einem mittleren Strom von $5 \mu\text{A}$ und einem Polarisationsgrad von $(73 \pm 5)\%$.

Die geladenen Reaktionsprodukte wurden mit hochauflösenden Magnetspektrometern nachgewiesen und das neutrale Pion aus der Impuls- und Energiebilanz der Reaktion rekonstruiert. Die Wirkungsquerschnitte der Strukturfunktionen σ_0 , σ_{LT} , σ_{TT} , $\sigma_{LT'}$ sowie die Helizitätsasymmetrie $A_{LT'}$ wurden bei einer festen Polarisation des virtuellen Photons von $\epsilon = 0.933$ bestimmt. Die Daten decken einen Bereich bis $\Delta W = 40 \text{ MeV}$ oberhalb der Pion-Produktionsschwelle ab.

Vergleiche mit den Modellvorhersagen aus MAID2000 zeigen, daß die helizitätsunabhängigen Wirkungsquerschnitte innerhalb der statistischen und systematischen Fehler von MAID gut beschrieben werden. Die chirale Störungstheorie überschätzt die Größe des Transversal-transversal-Anteils erheblich. Hier wurde im Gegensatz zur Theorie im Experiment ein mit null verträglicher Wert ermittelt.

Im Rahmen des apparativen Teils dieser Arbeit wurde das Steuerungssystem für die Drei-Spektrometer-Anlage von Grund auf neu konzipiert und implementiert. Das neue System basiert auf einer Client/Server-Grundstruktur, die unempfindlich gegenüber dem Ausfall einzelner Teilsysteme ist. Die Kommunikation mit anderen Systemen, insbesondere der Datenerfassung, wurde verbessert, so daß die Steuerungsdaten gemeinsam mit den Ereignissen aus der Datenerfassung archiviert werden.

Anhang A

Datenanalyse

A.1 χ^2 -Minimierung

In Kapitel 5.3 wird eine χ^2 -Minimierung an Gleichung 5.3 mit Hilfe des Matrixformalismus durchgeführt. Die Meßwertematrix $\tilde{\beta}$ und die Normalmatrix $\tilde{\alpha}$ dazu lauten:

$$\tilde{\beta} = \begin{pmatrix} \sum_{n=1}^N \frac{1}{\sigma^2} y_n \\ \sum_{n=1}^N \frac{1}{\sigma^2} y_n \cos x_n \\ \sum_{n=1}^N \frac{1}{\sigma^2} y_n \cos 2x_n \end{pmatrix} \quad (\text{A.1})$$

$$\tilde{\alpha} = \begin{pmatrix} \sum_{n=1}^N \frac{1}{\sigma^2} & \sum_{n=1}^N \frac{1}{\sigma^2} \cos x_n & \sum_{n=1}^N \frac{1}{\sigma^2} \cos 2x_n \\ \sum_{n=1}^N \frac{1}{\sigma^2} \cos x_n & \sum_{n=1}^N \frac{1}{\sigma^2} \cos^2 x_n & \sum_{n=1}^N \frac{1}{\sigma^2} \cos x_n \cos 2x_n \\ \sum_{n=1}^N \frac{1}{\sigma^2} \cos 2x_n & \sum_{n=1}^N \frac{1}{\sigma^2} \cos x_n \cos 2x_n & \sum_{n=1}^N \frac{1}{\sigma^2} \cos^2 2x_n \end{pmatrix} \quad (\text{A.2})$$

σ ist der mittlere quadratisch addierte statistische Fehler von y_n .

A.2 Wirkungsquerschnitte in tabellarischer Form

A.2.1 Kinematiken 0-6 bis 4-2

Dieser Abschnitt listet die ermittelten Wirkungsquerschnitte aus den Kinematiken 0-6 bis 4-2. Die angegebenen Fehler entsprechen den jeweiligen Diagonalelementen der Fehlermatrix aus der χ^2 -Minimierung. Für σ_0 , σ_{LT} und σ_{TT} werden die Ergebnisse nach den drei in Kapitel 5.3 beschriebenen Methoden präsentiert.

Der Interferenzterm $\sigma_{LT'}$ wird mit einem Schnitt um $\theta_\pi^* = 90^\circ$ sowie einer Gewichtung mit $h_{\text{eff}} \cdot \sin \phi_\pi^*$ bestimmt.

ΔW (MeV)	$ \theta_\pi^* - 90^\circ < 15^\circ$		mit MAID2000		ohne MAID2000	
	σ_0 (nb/sr)	$\delta\sigma_0$ (nb/sr)	σ_0 (nb/sr)	$\delta\sigma_0$ (nb/sr)	σ_0 (nb/sr)	$\delta\sigma_0$ (nb/sr)
0,5	29,45	0,98	33,95	0,46	31,45	0,42
1,5	43,65	0,60	50,84	0,32	42,63	0,26
2,5	60,27	1,98	75,90	0,44	57,58	0,33
3,5	87,96	14,17	99,55	0,72	68,39	0,52
4,5	115,20	7,69	129,12	1,08	84,11	0,74
5,5	137,66	10,39	165,76	2,03	104,84	1,29
6,5	172,94	5,55	186,24	4,22	121,50	2,55
7,5	192,93	10,64	223,09	8,58	163,24	6,53
8,5	243,87	7,14	257,03	15,64	212,84	14,55
9,5	288,35	7,91	306,86	15,28	266,96	15,01
10,5	337,02	10,90	350,64	13,50	310,43	11,94
11,5	364,19	14,31	397,75	10,71	353,61	9,50
12,5	407,85	26,22	428,17	16,33	383,93	14,10
13,5	443,91	50,51	456,40	26,41	414,14	22,52
14,5	464,07	25,57	470,02	65,78	426,72	52,60
15,5	575,50	42,64	560,63	41,94	522,28	37,81
16,5	618,29	38,26	624,92	63,89	577,87	57,09
17,5	650,37	81,73	679,47	66,99	647,74	62,67
18,5	659,86	22,31	660,72	20,94	656,13	20,69
19,5	738,05	54,60	693,74	54,46	687,76	53,45
20,5	768,01	22,56	764,87	22,31	761,99	21,92
21,5	805,98	35,45	810,72	34,09	805,53	33,85
22,5	928,01	42,13	926,13	42,06	914,06	41,48
23,5	971,06	59,16	967,98	59,85	954,45	58,70
24,5	1018,29	46,15	1022,31	46,36	1007,24	45,37
25,5	1113,36	79,01	1125,55	79,59	1090,96	77,52
26,5	1190,21	61,82	1211,88	58,54	1173,62	56,20
27,5	1264,32	69,29	1281,33	68,12	1244,35	65,53
28,5	1328,96	79,22	1370,84	87,47	1327,68	83,45
29,5	1376,56	66,42	1411,19	69,44	1374,86	65,98
30,5	1427,07	75,04	1428,44	76,50	1393,12	72,89
31,5	1580,40	53,99	1596,15	53,25	1562,96	51,83
32,5	1722,72	54,70	1758,16	55,59	1720,00	53,93
33,5	1879,75	71,25	1831,93	85,21	1799,96	83,34
34,5	2003,17	70,17	2050,06	71,18	2007,96	69,11
35,5	2027,60	95,33	2080,07	97,32	2026,10	94,24
36,5	2169,02	110,72	2206,81	108,95	2154,73	105,70
37,5	2146,90	115,86	2181,75	116,23	2128,97	112,53
38,5	2223,17	127,14	2275,37	129,77	2215,24	124,74
39,5	2492,24	140,15	2559,05	147,11	2504,09	140,73

ΔW (MeV)	$ \theta_\pi^* - 90^\circ < 15^\circ$		mit MAID2000		ohne MAID2000	
	σ_{LT} (nb/sr)	$\delta\sigma_{LT}$ (nb/sr)	σ_{LT} (nb/sr)	$\delta\sigma_{LT}$ (nb/sr)	σ_{LT} (nb/sr)	$\delta\sigma_{LT}$ (nb/sr)
0,5	-6,11	0,34	-6,75	0,16	-5,19	0,14
1,5	-12,76	0,21	-13,43	0,11	-10,00	0,09
2,5	-17,40	0,66	-21,82	0,15	-14,18	0,12
3,5	-19,85	4,78	-29,38	0,26	-17,30	0,18
4,5	-26,43	2,62	-35,04	0,39	-18,29	0,26
5,5	-35,63	3,56	-44,37	0,73	-22,79	0,47
6,5	-38,56	1,92	-51,17	1,53	-27,60	0,93
7,5	-48,12	3,77	-55,14	3,10	-34,41	2,36
8,5	-54,84	2,51	-65,33	5,69	-48,37	5,29
9,5	-61,97	2,80	-70,66	5,51	-55,44	5,41
10,5	-71,61	3,89	-80,31	4,92	-64,35	4,35
11,5	-71,96	5,15	-86,32	3,86	-68,24	3,42
12,5	-77,27	9,51	-90,63	5,92	-70,98	5,12
13,5	-72,04	18,47	-82,16	9,66	-63,41	8,24
14,5	-82,88	10,51	-82,18	24,24	-61,98	19,38
15,5	-98,14	17,99	-98,54	17,38	-82,07	15,67
16,5	-110,21	16,71	-112,97	27,16	-92,06	24,27
17,5	-104,87	35,96	-121,25	29,48	-106,81	27,58
18,5	-106,31	8,26	-107,64	7,75	-105,33	7,66
19,5	-110,65	19,18	-99,36	19,64	-96,32	19,27
20,5	-114,60	7,23	-118,68	7,15	-113,19	7,02
21,5	-134,47	11,57	-139,26	11,12	-130,54	11,04
22,5	-150,06	13,99	-156,12	13,97	-145,09	13,77
23,5	-132,00	19,68	-137,13	19,91	-125,40	19,52
24,5	-143,49	15,46	-148,54	15,53	-135,47	15,20
25,5	-141,34	27,55	-154,96	27,75	-139,05	27,03
26,5	-180,67	21,70	-192,89	20,55	-173,71	19,73
27,5	-188,23	24,50	-196,43	24,08	-178,14	23,17
28,5	-190,99	27,15	-212,00	31,14	-190,87	29,71
29,5	-186,92	22,92	-206,14	23,96	-187,99	22,77
30,5	-185,54	26,07	-206,77	27,60	-189,13	26,30
31,5	-217,46	19,12	-229,54	18,86	-213,92	18,35
32,5	-249,82	18,59	-260,75	18,89	-245,00	18,33
33,5	-243,85	24,37	-245,24	30,58	-231,78	29,91
34,5	-281,66	24,16	-294,21	24,51	-278,80	23,80
35,5	-281,41	33,04	-294,29	33,73	-279,38	32,66
36,5	-281,76	38,63	-293,50	38,01	-279,40	36,87
37,5	-283,54	40,68	-296,15	40,81	-281,48	39,51
38,5	-293,31	44,33	-305,89	45,25	-289,67	43,50
39,5	-254,61	44,34	-275,03	46,54	-257,99	44,52

ΔW (MeV)	$ \theta_\pi^* - 90^\circ < 15^\circ$		mit MAID2000		ohne MAID2000	
	σ_{TT} (nb/sr)	$\delta\sigma_{TT}$ (nb/sr)	σ_{TT} (nb/sr)	$\delta\sigma_{TT}$ (nb/sr)	σ_{TT} (nb/sr)	$\delta\sigma_{TT}$ (nb/sr)
0,5	-1,41	1,49	1,75	0,69	1,20	0,63
1,5	14,75	0,92	12,11	0,48	9,66	0,40
2,5	23,61	3,09	21,72	0,66	16,88	0,50
3,5	-9,28	22,30	25,98	1,09	14,98	0,78
4,5	10,46	12,10	23,53	1,64	9,86	1,11
5,5	21,93	16,35	24,51	3,08	7,58	1,96
6,5	15,16	8,73	41,76	6,40	13,74	3,86
7,5	36,96	16,59	6,48	13,18	-4,19	10,04
8,5	21,37	11,24	45,31	24,04	33,99	22,35
9,5	34,38	12,45	14,13	23,83	6,02	23,41
10,5	33,42	17,15	46,91	20,91	30,48	18,50
11,5	19,82	22,52	26,33	16,86	2,01	14,95
12,5	8,22	41,26	48,21	25,69	8,50	22,19
13,5	-28,81	79,49	-9,49	41,56	-42,75	35,44
14,5	39,96	44,13	1,46	103,51	-43,09	82,78
15,5	-40,07	75,03	-9,21	72,40	-36,05	65,27
16,5	26,41	67,00	0,97	112,43	-38,93	100,48
17,5	14,27	143,12	43,80	117,32	16,46	109,75
18,5	22,97	29,80	32,10	27,97	21,08	27,63
19,5	-33,83	75,81	27,46	76,58	15,47	75,16
20,5	-37,24	29,42	-32,78	29,09	-48,97	28,58
21,5	7,45	47,15	16,14	45,33	-6,40	45,01
22,5	-0,95	57,03	17,74	56,94	-11,69	56,15
23,5	-48,07	82,40	-15,84	83,36	-49,82	81,76
24,5	6,09	64,28	27,93	64,57	-10,92	63,20
25,5	-70,61	113,60	-34,11	114,42	-54,91	111,45
26,5	24,02	88,87	29,79	84,16	5,15	80,79
27,5	-3,55	99,61	3,25	97,93	-25,22	94,21
28,5	-16,85	113,45	16,07	125,76	-15,02	119,97
29,5	-62,78	95,11	-33,68	99,44	-61,60	94,49
30,5	-135,19	107,45	-60,03	109,98	-86,02	104,79
31,5	-49,91	73,89	-2,51	72,88	-32,24	70,94
32,5	-121,12	73,82	-104,86	75,00	-132,84	72,77
33,5	-114,29	96,15	22,59	116,62	-15,53	114,07
34,5	-167,65	94,68	-150,70	96,04	-191,64	93,25
35,5	-50,50	128,64	-18,97	131,32	-50,85	127,16
36,5	-209,03	149,40	-163,26	147,01	-201,50	142,62
37,5	-167,38	156,34	-108,28	156,83	-157,51	151,85
38,5	-51,43	168,27	11,58	171,75	-55,18	165,10
39,5	-396,60	182,56	-308,11	191,62	-393,39	183,31

ΔW (MeV)	$\sigma_{LT'}$ (nb/sr)	$\delta\sigma_{LT'}$ (nb/sr)	ΔW (MeV)	$\sigma_{LT'}$ (nb/sr)	$\delta\sigma_{LT'}$ (nb/sr)
0,5	4,97	8,10	20,5	21,93	20,14
1,5	2,79	5,43	21,5	49,54	22,02
2,5	3,61	4,69	22,5	18,45	28,17
3,5	-4,24	4,76	23,5	88,48	27,42
4,5	0,27	4,78	24,5	42,52	28,65
5,5	2,26	5,31	25,5	26,48	33,49
6,5	-2,92	5,88	26,5	61,41	38,86
7,5	6,64	6,65	27,5	80,28	37,01
8,5	7,76	7,64	28,5	117,00	47,87
9,5	1,70	8,40	29,5	96,23	45,99
10,5	12,11	8,95	30,5	328,65	57,86
11,5	20,15	10,07	31,5	-45,74	57,23
12,5	15,99	10,77	32,5	70,70	67,25
13,5	26,55	11,45	33,5	152,40	69,38
14,5	8,77	13,18	34,5	53,07	83,06
15,5	12,33	14,29	35,5	44,55	85,24
16,5	47,53	15,41	36,5	104,24	101,69
17,5	41,34	15,65	37,5	193,39	101,39
18,5	20,35	16,77	38,5	179,62	147,53
19,5	17,06	20,04	39,5	452,62	132,36

ΔW (MeV)	$A_{LT'}$	$\delta A_{LT'}$
2,5	0,49	0,74
7,5	0,79	0,44
12,5	2,12	0,40
17,5	2,18	0,39
22,5	2,08	0,40
27,5	2,29	0,44
32,5	2,66	0,52
37,5	2,75	0,64
42,5	1,03	0,82
47,5	2,90	1,07

A.2.2 Kinematik 4-3

Wirkungsquerschnitte σ_0 und σ_{LT} aus Kinematik 4-3.

ΔW (MeV)	$ \phi_\pi^* - 90^\circ < 135^\circ$		ohne Schnitt in ϕ_π^*	
	σ_0 (nb/sr)	$\delta\sigma_0$ (nb/sr)	σ_0 (nb/sr)	$\delta\sigma_0$ (nb/sr)
0,5	12,95	4,49	20,26	3,01
1,5	46,69	3,61	47,70	2,26
2,5	67,56	3,66	60,18	2,48
3,5	92,03	4,78	70,03	3,48

ΔW (MeV)	$ \phi_\pi^* - 90^\circ < 135^\circ$		ohne Schnitt in ϕ_π^*	
	σ_{LT} (nb/sr)	$\delta\sigma_{LT}$ (nb/sr)	σ_{LT} (nb/sr)	$\delta\sigma_{LT}$ (nb/sr)
0,5	0,85	1,87	-2,63	1,01
1,5	-10,18	1,53	-10,60	0,78
2,5	-16,03	1,48	-12,85	0,93
3,5	-21,55	1,80	-13,40	1,33

Anhang B

Steuerung

B.1 Befehle und Fehlermeldungen

CMD_REQUEST_PING: Fordert ein *ping* von der Gegenseite der aktuellen Verbindung an. Wenn innerhalb von 20 Sekunden keine Rückantwort kommt, wird die Verbindung getrennt.

CMD_REPLY_PING: Antwort auf die *ping*-Anfrage

CMD_REQUEST_CONNECT: Aufforderung, eine Verbindung zum angegebenen Gerät aufzubauen

Parameter	Typ	Bemerkung
1	string	port@ipnummer

CMD_REPLY_CONNECT: Rückantwort auf die Verbindungsanfrage

Parameter	Typ	Bemerkung
1	string	port@ipnummer
2	integer	erfolgreich (1), fehlgeschlagen (0)

CMD_REQUEST_CLOSE: Aufforderung, die angegebene Verbindung zu trennen. Ist das Argument ein leerer String, so wird die aktuelle Verbindung geschlossen.

Parameter	Typ	Bemerkung
1	string	port@ipnummer

CMD_REPLY_CLOSE: Rückantwort auf die Aufforderung, die Verbindung zu beenden

Parameter	Typ	Bemerkung
1	string	port@ipnummer
2	integer	erfolgreich (1), fehlgeschlagen (0)

CMD_DUMP: Fordert eine Kopie aller Eigenschaftslisten von der Gegenseite der aktuellen Verbindung an. Die Rückantwort besteht aus einer Liste von *CMD_SET_PROPERTY*-Anweisungen für jedes Feld der Eigenschaftslisten. Alle folgenden Änderungen an den Eigenschaftslisten werden automatisch an die Gegenseite gesendet (siehe *CMD_MONITOR*).

CMD_REQUEST_OBJECT_LIST: Aufforderung, eine Übersichtsliste bestehend aus Name und Typ aller Objekte von der Gegenseite der aktuellen Verbindung zu senden

CMD_REPLY_OBJECT_LIST: Sendet eine Übersichtsliste für alle Objekte

Parameter	Typ	Bemerkung
1	integer	Anzahl n der Objekte
$2 * n$	integer	Typ der Klasse
$2 * n + 1$	string	Objektname

CMD_REQUEST_CREATE_OBJECT: Aufforderung, ein neues Objekt vom angegebenen Klassentyp zu erzeugen. Eine Auflistung der Typen und ihre Kennzahl findet sich in der Datei *Resource.h* in den Mezzo-Quelltexten.

Parameter	Typ	Bemerkung
1	integer	Typ der Klasse
2	string	Objektname

CMD_REPLY_CREATE_OBJECT: Bestätigung für das Erzeugen eines Objektes

Parameter	Typ	Bemerkung
1	string	Objektname
2	integer	erfolgreich (1), fehlgeschlagen (0)

CMD_REQUEST_DELETE_OBJECT: Aufforderung, das angegebene Objekt zu löschen

Parameter	Typ	Bemerkung
1	string	Objektname

CMD_REPLY_DELETE_OBJECT: Bestätigung für das Löschen eines Objektes

Parameter	Typ	Bemerkung
1	string	Objektname
2	integer	erfolgreich (1), fehlgeschlagen (0)

Parameter	Typ	Bemerkung
1	string	Objektname
2	string	Eigenschaftsfeld

CMD_GET_PROPERTY: Eigenschaftsfeld der angegebenen Eigenschaftsliste (Objekt) auslesen

Wenn Objekt oder Eigenschaftsfeld nicht existieren, wird die Fehlermeldung *ERR_INVALID_PROPERTY* erzeugt. Wenn die Eigenschaftsliste vorhanden ist wird als Rückantwort der Befehl *CMD_SET_PROPERTY* erzeugt, sofern das Feld nicht die Flagge *PTO_ASYNC* besitzt. Andernfalls wird die Meldung *INFO_ASYNC_VALUE* verschickt.

CMD_SET_PROPERTY: Aktualisiert den Wert des Eigenschaftsfeldes. Im Fehlerfall gelten die gleichen Bedingungen wie beim Befehl *CMD_GET_PROPERTY*

Parameter	Typ	Bemerkung
1	string	Objektname
2	integer	Anzahl der Felder
$2 * i + 1$	string	Eigenschaftsfeld
$2 * i + 2$	unbekannt	Wert

CMD_REQUEST_RESET: Aufforderung, einen Reset des angegebenen Objektes auszuführen. Die auszuführende Aktion ist abhängig von der Implementation des zugehörigen Gerätetreibers. In der Grundeinstellung passiert gar nichts. Wenn die Eigenschaftsliste nicht existiert, wird die Fehlermeldung *ERR_INVALID_PROPERTY* generiert, ansonsten *CMD_REPLY_RESET* gesendet.

Parameter	Typ	Bemerkung
1	string	Objektname

CMD_REPLY_RESET: Bestätigung für den Reset am vorgegebenen Objekt

Parameter	Typ	Bemerkung
1	string	Objektname

CMD_REQUEST_TRIGGER: Triggert eine implementationsabhängige Aktion beim angegebenen Objekt. Wenn das Objekt nicht existiert, wird die Fehlermeldung *ERR_INVALID_PROPERTY* erzeugt, ansonsten der Befehl *CMD_REPLY_TRIGGER* zurück gesendet.

Parameter	Typ	Bemerkung
1	string	Objektname

CMD_REPLY_TRIGGER: Rückantwort auf die Triggeranforderung.

Parameter	Typ	Bemerkung
1	string	Objektname

CMD_GET_PROPERTY_LIST: Fordert alle Eigenschaftsfelder des angegebenen Objektes an. Wenn das Objekt nicht existiert wird die Fehlermeldung *ERR_INVALID_PROPERTY* erzeugt, andernfalls der Befehl *CMD_SET_PROPERTY* für die Felder verschickt.

Parameter	Typ	Bemerkung
1	string	Objektname

CMD_REQUEST_TRIM: Aufforderung, einen automatischen Abgleich für das angegebene Objekt durchzuführen. Wenn das Objekt nicht existiert wird die Meldung *ERR_INVALID_PROPERTY* generiert. Dieser Befehl gilt nur für einen *Alias*.

Parameter	Typ	Bemerkung
1	string	Objektname
2	integer	Trimlevel

CMD_REPLY_TRIM: Bestätigung für den automatischen Abgleich.

Parameter	Typ	Bemerkung
1	string	Objektname
2	integer	Trimlevel

CMD_REQUEST_RENAME_OBJECT: Ändert den Namen der angegebenen Eigenschaftsliste sofern es sich um einen *Alias* handelt und der neue Name nicht schon vergeben ist.

CMD_REPLY_RENAME_OBJECT: Bestätigung für das Umbenennen eines Objektes

Parameter	Typ	Bemerkung
1	string	alter Objektname
2	string	neuer Objektname

Parameter	Typ	Bemerkung
1	string	Objektname
2	integer	erfolgreich (1), fehlgeschlagen (0)

CMD_REQUEST_UPDATE: Schickt eine Aufforderung an den angegebenen Kanal, das entsprechende Eigenschaftsfeld zu aktualisieren. Wenn das Objekt nicht existiert wird die Meldung *ERR_INVALID_PROPERTY* generiert.

Parameter	Typ	Bemerkung
1	string	Objektname
2	string	Eigenschaftsfeld

CMD_REPLY_UPDATE: Bestätigung für den Erhalt der Aktualisierungsanforderung.

Parameter	Typ	Bemerkung
1	string	Objektname
2	string	Eigenschaftsfeld

CMD_MONITOR: Nach Erhalt dieses Befehls werden alle Änderungsmeldungen des Systems an den angeschlossenen TCP/IP Endpunkt geschickt. Im Unterschied zum Befehl *CMD_DUMP* wird zuvor keine Kopie aller Objekte erzeugt.

CMD_REQUEST_LOCK: Sperrt die angegebene Semaphore für den Zugriff durch andere Verbindungen. Die Sperre ist an die aktuelle Verbindung gebunden und wird automatisch aufgelöst, wenn sie unterbrochen wird. Rekursives oder wiederholtes Sperren der Semaphore durch die gleiche Verbindung ist nicht zulässig.

Parameter	Typ	Bemerkung
1	string	Objektname

CMD_REPLY_LOCK: Bestätigung für das Sperren der angegebenen Semaphore. Diese Meldung wird erst generiert, nachdem die Semaphore tatsächlich gesperrt wurde.

Parameter	Typ	Bemerkung
1	string	Objektname
2	integer	erfolgreich (1), fehlgeschlagen (0)

CMD_REQUEST_UNLOCK: Hebt die Sperre für die angegebene Semaphore wieder auf. Dabei ist es unerheblich, ob die Semaphore auch vom aktuellen Teilnehmer gesperrt wurde. Wartet bereits eine weitere Verbindung darauf, die Semaphore zu sperren, so bekommt sie automatisch die Meldung *CMD_REPLY_LOCK* geschickt.

Parameter	Typ	Bemerkung
1	string	Objektname

CMD_REPLY_UNLOCK: Bestätigung für die Aufhebung der Sperre an der angegebenen Semaphore.

Parameter	Typ	Bemerkung
1	string	Objektname
2	integer	erfolgreich (1), fehlgeschlagen (0)

ERR_INVALID_PROPERTY: Die Fehlermeldung wird verschickt, wenn entweder die angegebene Eigenschaftsliste oder das zugehörige Feld nicht existiert.

Parameter	Typ	Bemerkung
1	string	Objektname
2	string	Eigenschaftsfeld

ERR_INVALID_VALUE: Diese Fehlermeldung wird generiert, wenn versucht wurde einen ungültigen Wert in ein Eigenschaftsfeld zu schreiben, z.B. außerhalb des erlaubten Wertebereichs oder wenn der *Alias* nicht mit einem Hardwarekanal verbunden ist.

Parameter	Typ	Bemerkung
1	string	Objektname
2	string	Eigenschaftsfeld

ERR_INVALID_UPDATE: Fehler bei der Aktualisierung eines zum Alias gehörenden Hardwarekanals

INFO_CREATE_OBJECT: Meldung über das Erzeugen eines Objektes

Parameter	Typ	Bemerkung
1	string	Objektname
2	string	Eigenschaftsfeld

Parameter	Typ	Bemerkung
1	integer	Typ der Klasse
2	string	Objektname

Parameter	Typ	Bemerkung
1	integer	Typ der Klasse
2	string	Objektname

INFO_DELETE_OBJECT: Meldung über das Vernichten eines Objektes

INFO_ASYNC_VALUE: Die Informationsmeldung wird verschickt, wenn das Eigenschaftsfeld einem Hardwarekanal zuzuordnen ist und aus Zeitgründen keine Bestätigungsmeldung für den letzten Befehl erzeugt werden kann.

Parameter	Typ	Bemerkung
1	integer	Typ der Klasse
2	string	Objektname

INFO_NOT_OWNER: Es wurde versucht eine geschützte Eigenschaftsliste zu ändern. Zu diesen Objekten gehören beispielsweise die Kopien der Hardwarekanäle im zentralen Prozeß.

Parameter	Typ	Bemerkung
1	integer	Typ der Klasse
2	string	Objektname

B.2 Datenbank

In der Datenbank wird die Konfiguration der Aliase, der Gerätetreiber und der Semaphoren gespeichert. Entsprechend müssen verschiedene Tabellen in der gemeinsamen Datenbank existieren: "alias" für die Aliase, "semaphore" für die Semaphoren und für jeden Gerätetreiber eine Tabelle mit dem Netzwerknamen des Computers, auf dem sie gestartet werden.

B.2.1 Gerätetreiber

Die Tabelle für die Gerätetreiber hat drei Felder: Ein Index legt die Reihenfolge fest, in der die Module gestartet werden, der Bezeichner für das jeweilige Modul und ein Feld für die Argumente zum Start des Gerätetreibers.

```
CREATE TABLE "aicontrol123" (  
    "index" int4,  
    "modul" character varying(128),  
    "parameter" character varying(128));
```

B.2.2 Semaphore

Die Tabelle für die Semaphore benötigt nur ein Feld mit dem Namen des entsprechenden Objekts.

```
CREATE TABLE "semaphore" (  
    "name" character varying(128)  
);
```

B.2.3 Alias

Diese Tabelle speichert alle beschreibbaren Eigenschaftsfelder eines Alias. Die internen Datentypen werden entsprechend angepaßt, z.B. wird der Typ *float8* für einen *double* verwendet.

```
CREATE TABLE "alias" (  
    "name" character varying(128),  
    "channel" character varying(128),  
    "fatalflags" int4,  
    "seriousflags" int4,  
    "state" int4,  
    "polling" float8,  
    "alarmstate" int4,  
    "alarmtype" int4,  
    "autotrimlevel" int4,  
    "digits" int4,  
    "epsilon" float8,  
    "delta" float8,  
    "limithigh" float8,  
    "limitlow" float8,  
    "alarmhigh" float8,  
    "alarmlow" float8,
```

```
"unit" character varying(128),
"link" character varying(128),
"range" int4,
"startbit" int4,
"bitcount" int4,
"expression" character varying(255),
"demandvalue" text,
"mask" text,
"label" text
);
```

B.3 Eigenschaftslisten

B.3.1 Analog Digital Konverter (ADC)

ActualValue (double, R): Dieses Feld enthält den zuletzt vom Gerätetreiber ausgelesenen Wert. Aktualisiert wird er entweder durch das manuelle senden des Befehls *CMD_REQUEST_UPDATE* oder durch die automatische periodische Auslese, festgelegt im Feld *Polling*.

AlarmFlags (integer, R): Hier wird der augenblickliche Alarmzustand des Kanals abgelegt. Wenn die untere Alarmschwelle unterschritten wurde ist Bit 1 gesetzt und beim Überschreiten der oberen Alarmschwelle Bit 2. Verfügt der Kanal über keine hardwareseitige Überwachung, wird dieses Eigenschaftsfeld zusammen mit dem Feld *ActualValue* durch einfache Vergleiche im Gerätetreiber aktualisiert.

AlarmHigh (double, R/W): obere Alarmschwelle.

AlarmLow (double, R/W): untere Alarmschwelle

AlarmState (integer, R/W): Aktiviert (1) bzw. deaktiviert (0) die Überwachung der Alarmschwellen.

AlarmType (integer, R/W): Mit diesem Feld wird der Typ des erzeugten Alarmzustands festgelegt. Mögliche Werte sind *MOMENTARY* (0) und *LATCHING* (1). Bei *MOMENTARY* wird das entsprechende Bit in *AlarmFlags* nur solange gesetzt, wie die Schwellen über- bzw. unterschritten sind. Im Modus *LATCHING* bleibt die Flagge aktiv bis der Befehl *CMD_REQUEST_RESET* gesendet wird.

Class (integer, R): feste Identifikationsnummer für die Klasse

Epsilon (double, R/W): Die Größe *Epsilon* legt die Genauigkeit für die Auslese fest, da in der Regel das niederwertigste Bit eines ADC „wackelt“.

Error (string, R): Wenn der Zugriff auf das Gerät aus irgendeinem Grund nicht funktioniert hat, legt der Gerätetreiber in diesem Feld die zugehörige Fehlermeldung ab. Die Meldung wird nur an den zentralen Prozeß weitergeleitet, wenn sie sich von der letzten Fehlermeldung unterscheidet.

Init (integer, R): Unmittelbar nach dem Start des Gerätetreibers muß die Hardware erst initialisiert und die augenblickliche Konfiguration ermittelt werden. Während dieser Zeit wird das Feld auf 0 und nach erfolgreicher Initialisierung auf 1 gesetzt.

Polling (double, R/W): Mit dem Feld *Polling* wird die Frequenz für die periodische Auslese der Hardware eingestellt. Erwartet wird ein Wert, der den Abstand zweier Auslesezyklen in Sekunden angibt. Bei 0 wird das Polling abgeschaltet. Die höchste erlaubte Frequenz ist 10Hz.

Range (integer, R/W): Dieses Feld legt die Nummer des augenblicklichen Meßbereichs fest wie sie von *RangeList* aufgeführt wurde.

RangeList (rangelist[], R): Liste aller verfügbaren Meßbereiche.

State (integer, R/W): Mit dem Eigenschaftsfeld *State* wird der gesamte Kanal ein- (1) bzw. ausgeschaltet (0). Ein abgeschalteter Kanal pollt nicht und überwacht auch keine Alarmschwellen.

B.3.2 Digital Analog Konverter (DAC)

AlarmFlags (integer, R): nicht benutzt

AlarmState (integer, R/W): nicht benutzt

AlarmType (integer, R/W): nicht benutzt

Class (integer, R): feste Identifikationsnummer für die Klasse

DemandValue (double, R/W): Hier wird der Sollwert für den DAC festgelegt. Eine hardwareseitige Auslese erfolgt während der Initialisierungsphase des Kanals oder unmittelbar nach dem Senden des Wertes an das Gerät. Eine regelmäßige Auslese findet nicht statt.

Epsilon (double, R/W): Die Größe *Epsilon* legt die Genauigkeit für den Vergleich zweier Werte fest. Es wird auch für die Filterung des niederwertigsten Bits bei der Auslese der Stellgröße verwendet.

Error (string, R): siehe *ADC*

Init (integer, R): siehe *ADC*

Polling (double, R/W): Mit dem Feld *Polling* wird die Frequenz für die periodische Auslese der Hardware eingestellt. Erwartet wird ein Wert, der den

Abstand zweier Auslesezyklen in Sekunden angibt. Bei 0 wird das Polling abgeschaltet. Die schnellste erlaubte Frequenz ist 10Hz. Es wird nur die Stellgröße regelmäßig abgefragt (siehe *SetPoint*).

Range (integer, R/W): siehe *ADC*

RangeList (rangelist[], R): siehe *ADC*

SetPoint (double, R): Dieses Feld enthält die zuletzt vom Gerätetreiber ausgelesene Stellgröße. Aktualisiert wird er entweder durch das manuelle senden des Befehls *CMD_REQUEST_UPDATE* oder durch die automatische periodische Auslese, festgelegt im Feld *Polling*.

State (integer, R/W): Mit dem Eigenschaftsfeld *State* wird der gesamte Kanal ein- (1) bzw. ausgeschaltet (0). Ein abgeschalteter Kanal pollt nicht.

B.3.3 Digital In (DI)

ActualValue (bit[], R): Dieses Feld enthält den zuletzt vom Gerätetreiber ausgelesenen Wert. Aktualisiert wird er entweder durch das manuelle senden des Befehls *CMD_REQUEST_UPDATE* oder durch die automatische periodische Auslese, festgelegt im Feld *Polling*.

AlarmFlags (integer, R): nicht benutzt

AlarmState (integer, R/W): nicht benutzt

AlarmType (integer, R/W): nicht benutzt

BitCount (integer, R): Dieses Feld enthält die Anzahl tatsächlich gültiger Bits. Zu beachten ist, daß *ActualValue* stets einen Wert mit einer durch vier teilbaren Bitanzahl verschickt.

Class (integer, R): feste Identifikationsnummer für die Klasse

Error (string, R): siehe *ADC*

Init (integer, R): siehe *ADC*

Polling (double, R/W): Mit dem Feld *Polling* wird die Frequenz für die periodische Auslese der Hardware eingestellt. Erwartet wird ein Wert, der den Abstand zweier Auslesezyklen in Sekunden angibt. Bei 0 wird das Polling abgeschaltet. Die schnellste erlaubte Frequenz ist 10Hz.

State (integer, R/W): siehe *ADC*

B.3.4 Digital Out (DO)

AlarmFlags (integer, R): nicht benutzt

AlarmState (integer, R/W): nicht benutzt

AlarmType (integer, R/W): nicht benutzt

BitCount (integer, R): Dieses Feld enthält die Anzahl tatsächlich gültiger Bits. Zu beachten ist, daß *DemandValue* und *SetPoint* stets einen Wert mit einer durch vier teilbaren Bitanzahl verschicken.

Class (integer, R): feste Identifikationsnummer für die Klasse

DemandValue (bit[], R/W): Hier wird der Sollwert für den DO festgelegt. Eine hardwareseitige Auslese erfolgt während der Initialisierungsphase des Kanals oder unmittelbar nach dem Senden des Wertes an das Gerät. Eine regelmäßige Auslese findet nicht statt.

Error (string, R): siehe *DI*

Init (integer, R): siehe *DI*

Polling (double, R/W): Mit dem Feld *Polling* wird die Frequenz für die periodische Auslese der Hardware eingestellt. Erwartet wird ein Wert, der den Abstand zweier Auslesezyklen in Sekunden angibt. Bei 0 wird das Polling abgeschaltet. Die schnellste erlaubte Frequenz ist 10Hz. Es wird nur der *SetPoint* regelmäßig abgefragt.

SetPoint (bit[], R): Dieses Feld enthält die zuletzt vom Gerätetreiber ausgelesene Stellgröße. Aktualisiert wird er entweder durch das manuelle senden des Befehls *CMD_REQUEST_UPDATE* oder durch die automatische periodische Auslese, festgelegt im Feld *Polling*.

State (integer, R/W): siehe *DI*

B.3.5 Text

AlarmFlags (integer, R): nicht benutzt

AlarmState (integer, R/W): nicht benutzt

AlarmType (integer, R/W): nicht benutzt

Class (integer, R): feste Identifikationsnummer für die Klasse

DemandValue (string, R/W): Hier wird der Sollwert für TEXT festgelegt. Eine regelmäßige Auslese findet nicht statt.

Error (string, R): siehe *DO*

Init (integer, R): siehe *DO*

Polling (double, R/W): siehe *DO*

SetPoint (string, R): Dieses Feld enthält den zuletzt vom Gerätetreiber ermittelten Text. Aktualisiert wird er entweder durch das manuelle senden des Befehls *CMD_REQUEST_UPDATE* oder durch die automatische periodische Auslese, festgelegt im Feld *Polling*.

State (integer, R/W): siehe *DO*

B.3.6 Alias

ActualValue (string, R): Hier wird der zuletzt von einem *ADC* oder *DI* ermittelte und angepasste *ActualValue* eingetragen. Wenn es sich bei dem Kanal um einen *ADC* handelt, wird der Rohwert über die in *Expression* angegebene Formel in eine physikalisch sinnvolle Größe konvertiert und auf die in *Digits* angegebene Anzahl Nachkommastellen gerundet. Bei einem *DI* werden aus den Rohdaten angefangen bei *StartBit BitCount* Bits kopiert und in dem Feld abgelegt.

AlarmHigh (double, R/W): Für einen *ADC* wird hier die obere Alarmschwelle in physikalischen Einheiten eingetragen. Der Wert wird mit der in *Expression* angegebenen Umrechnungsformel konvertiert und zum Gerätetreiber geschickt.

AlarmLow (double, R/W): untere Alarmschwelle für einen *ADC* (siehe auch *AlarmHigh*)

AlarmState (integer, R/W): Aktiviert (*1*) bzw. deaktiviert (*0*) die Überwachung der Alarmschwellen durch den Gerätetreiber.

AlarmType (integer, R/W): Mit diesem Feld wird der Typ des erzeugten Alarmzustands festgelegt. Mögliche Werte sind *MOMENTARY* (*0*) und *LATCHING* (*1*). Bei *MOMENTARY* wird das entsprechende Bit in *AlarmFlags* nur solange gesetzt, wie die Schwellen über- bzw. unterschritten sind. Im Modus *LATCHING* bleibt die Flagge aktiv bis der Befehl *CMD_REQUEST_RESET* zum Alias oder direkt zum entsprechenden Kanal gesendet wird.

AutoTrimLevel (integer, R/W): Dieses Feld legt den Umfang des automatischen Abgleichs zwischen einem Kanal und dem zugehörigen Alias nach dem Verbindungsaufbau fest.

BitCount (integer, R/W): Bestimmt die Anzahl tatsächlich genutzter Bits bei einem *DI* oder *DO*. Die Zahl muß kleiner oder gleich der vom Kanal bereitgestellten Bitzahl sein. Das erste Bit wird im Eigenschaftsfeld *StartBit* angegeben.

Channel (string, R/W): Hier steht der Name des mit dem Alias verknüpften Kanals. Dieser Name muß das Format *port@ipnummer:bezeichner* haben. Wenn keine aktive Verbindung zu dem entsprechenden Gerätetreiber besteht, wird sie automatisch aufgebaut.

Class (integer, R): feste Identifikationsnummer für die Klasse

Delta (double, R/W): Definiert den maximalen erlaubten Unterschied (*Regelabweichung*) zwischen dem *ActualValue* eines *ADC* und dem *SetPoint* eines *DAC*. Beim Überschreiten des erlaubten Wertebereichs wird die Flagge *AF_ACTUALVALUE* gesetzt. Das Eigenschaftsfeld wird beim *ADC* ausgewertet (siehe auch *Link*).

DemandValue (string, R/W): Bei einem *DAC*, *DO* oder *TEXT* enthält dieses Feld den Sollwert für den Kanal. Bevor der Wert an den Gerätetreiber geschickt wird erfolgt eine Umrechnung des Wertes über die in *Expression* angegebene Formel (*DAC*) bzw. das Setzen der entsprechenden Bits bei einem *DO*.

Digits (integer, R/W): Dieses Feld legt die Anzahl gültiger Nachkommastellen für einen *ADC* oder *DAC* fest. Die Zahl wird bei allen Feldern vom Typ *double* angewendet.

Epsilon (double, R/W): *Epsilon* wird als Filterkriterium zur Reduktion des Datenstromes verwendet. Zwei Werte die sich um weniger als den angegebenen Wert unterscheiden werden als identisch angesehen und dementsprechend keine Aktualisierung an die Klienten des zentralen Prozessen geschickt. Das eingestellte *Epsilon* ist in Einheiten des verknüpften Kanals anzugeben. Es erfolgt keine Umrechnung über die in *Expression* vorgegebene Formel.

Error (string, R): Fehlermeldung des verknüpften Kanals

Expression (string, R/W): Die von einem *ADC* oder *DAC* gelieferten Rohdaten werden mit Hilfe der Formel in diesem Eigenschaftsfeld in eine physikalisch sinnvolle Größe umgerechnet. Erlaubt sind folgende Operationen und Funktionsaufrufe: +, -, *, /, ^, *sqrt*, *ln*, *log*, *exp*, *abs*, *sin*, *cos*, *tan*, *ceil*, *floor*, *modf* sowie beliebig viele Klammerebenen. Der Rohwert wird in der Variablen *this* gespeichert. Auf andere Aliase kann durch Angabe des entsprechenden Bezeichners zugegriffen werden. Beispiel:

$$(2.00455 + 1007.7 * this + 1.2224 * this^2 - 0.129974 * this^3) / beam.turns$$

FatalFlags (integer, R/W): Dieses Feld enthält die Flaggen zum Signalisieren fataler Fehlerzustände. Ein fataler Fehler liegt vor, wenn eine UND-Verknüpfung zwischen *Flags* und *FatalFlags* einen Wert ungleich Null ergibt (siehe auch *Flags*).

Flags (integer, R): Hier wird der augenblickliche Fehlerzustand eines Alias in Form eines Bitmusters abgelegt, wobei ein gesetztes Bit einem Fehler entspricht (niederwertigstes Bit zuerst):

AF_NOTCONNECTED Der Alias ist nicht mit einem Kanal des Gerätetreibers verbunden. Sofern der Alias nicht deaktiviert ist, versucht das System selbstständig eine Verbindung aufzubauen.

AF_DISABLED Der Kanal ist deaktiviert (siehe auch *State*).

AF_NOTINITIALIZED Während der Initialisierungsphase des Kanals wird dieses Bit gesetzt.

AF_DBFAILED Beim Zugriff auf die Datenbank ist ein Fehler aufgetreten. Die nächste erfolgreiche Aktualisierung der Datenbank löscht dieses Bit wieder.

AF_ACCESSFAILED noch nicht implementiert

AF_LIMITLOW *ActualValue* oder *SetPoint* haben das untere physikalische Limit unterschritten

AF_LIMITHIGH siehe *AF_LIMITLOW*

AF_ALARMLOW Der *ActualValue* hat die untere Alarmschwelle unterschritten. Dieses Bit wird vom *ADC* selbst generiert.

AF_ALARMHIGH siehe *AF_ALARMLOW*

AF_COMPAREFAILED Es gibt einen Unterschied zwischen der Konfiguration in der Datenbank und dem augenblicklichen Zustand des Kanals. Mit dem Befehl *CMD_REQUEST_TRIM* kann der zentrale Prozeß selbstständig einen Abgleich durchführen.

AF_SETPOINT Die Stellgröße und der vorgegebene Sollwert unterscheiden sich um mehr als *Epsilon*.

AF_ACTUALVALUE Der *ActualValue* eines *ADC* und der *SetPoint* des zugehörigen *DAC* unterscheiden sich um mehr als das vorgegebene *Delta*, oder *Mask* bei einem *DI* signalisiert einen Alarm

Label (string[], R/W): Dieses Feld enthält eine Liste vorgegebener Werte für den *DemandValue*, die über einen beliebigen Bezeichner angesprochen werden. Erwartet wird eine Stringliste mit abwechselnd einem Namen und dem zugehörigen Wert.

LimitHigh (double, R/W): oberes physikalisch sinnvolles Limit für einen *ADC* oder *DAC*

LimitLow (double, R/W): siehe *LimitHigh*

Link (string, R/W): Über dieses Feld wird ein *ADC* mit einem *DAC* (und umgekehrt) verknüpft. siehe auch *Delta*

Mask (bit[], R/W): Hier wird eine Maske zur Überprüfung bestimmter Bitmuster eines *DI* abgelegt. Ergibt die UND-Verknüpfung zwischen dem *ActualValue* und *Mask* einen Wert ungleich Null, wird die Flagge *AF_ACTUALVALUE* gesetzt.

Polling (double, R/W): Mit dem Feld *Polling* wird die Frequenz für die periodische Auslese der Hardware eingestellt. Erwartet wird ein Wert, der den Abstand zweier Auslesezyklen in Sekunden angibt. Bei 0 wird das Polling abgeschaltet. Die schnellste erlaubte Frequenz ist 10Hz.

Range (integer, R/W): Enthält den eingestellten Meßbereich eines *ADC* oder *DAC*

SeriousFlags (integer, R/W): Flaggenmuster für ernste Alarmzustände. siehe auch *Flags*

SetPoint (string, R): Hier wird der zuletzt von einem *DAC* oder *DO* ermittelte und angepasste *SetPoint* eingetragen. Wenn es sich bei dem Kanal um einen *DAC* handelt, wird der Rohwert über die in *Expression* angegebene Formel in eine physikalisch sinnvolle Größe konvertiert und auf die in *Digits* angegebene Anzahl Nachkommastellen gerundet. Bei einem *DO* werden aus den Rohdaten angefangen bei *StartBit BitCount* Bits kopiert und in dem Feld abgelegt.

StartBit (integer, R/W): definiert das erste gültige Bit für einen *DI* oder *DO*

State (integer, R/W): Über dieses Eigenschaftsfeld wird der Alias und sein zugehöriger Kanal aktiviert (1) bzw. deaktiviert (0). Ein deaktivierter Alias

- pollt nicht
- versucht keine Verbindung mit dem Gerätetreiber herzustellen
- überwacht keine Fehlerzustände

Unit (string, R/W): Im Feld *Unit* wird die Maßeinheit der entsprechenden physikalischen Größe eines *ADC* oder *DAC* abgelegt.

B.3.7 Database

Class (integer, R): feste Identifikationsnummer für die Klasse

DataSourceName (string, R/W): Dieses Feld enthält alle für den Zugriff auf die Datenbank nötigen Parameter. Ein Schreibbefehl auf dieses Feld führt zu einem erneuten Verbindungsaufbau. Im Falle von Postgres an der Dreispektrometer-Anlage beispielsweise

```
host=a1server2 dbname=mezzo
```

DatabaseError (string, R): enthält die letzte vom Datenbankbackend geschickte Fehlermeldung

B.3.8 Semaphore

Class (integer, R): feste Identifikationsnummer für die Klasse

LockList (string[], R): Hier kann die Liste aller Endpunkte abgefragt werden, die einen exklusiven Zugriff auf die Semaphore haben wollen. Gesperrt wird eine Semaphore mit dem Befehl *CMD_REQUEST_LOCK* und entriegelt mit *CMD_REQUEST_UNLOCK*.

B.3.9 TCP Interface

Class (integer, R): feste Identifikationsnummer für die Klasse

Latency (double, R): Alle 10 Sekunden schickt der zentrale Prozeß ein Ping an die Klienten. In diesem Feld steht die Zeit zwischen dem Absenden des Befehls und dem Empfang der Rückantwort, angegeben in Millisekunden. Wenn nach 20 Sekunden keine Antwort kommt, wird die Verbindung zum Klienten unterbrochen.

State (integer, R): Hier wird der Status der aktuellen Verbindung abgelegt. Mögliche Zustände sind:

Name	Wert	Bemerkung
OFFLINE	0	keine Verbindung zum Endpunkt
PENDING CONNECT	1	Verbindungsaufbau
ONLINE	2	Verbindung aktiv

B.4 Gerätetreiber

B.4.1 ADAM Bus

Die ADAM 4000 Serie der Firma Advantech besteht aus modularen und intelligenten Computer-Sensor-Schnittstellen für verteilte Anwendungen. Im Normalbetrieb können bis zu 16 Module über eine RS-485 Zweidrahtleitung und einen RS-485 nach RS-232 Konverter an einen PC angeschlossen werden. Jedes ADAM-Modul ist einzeln und individuell über eine Identifikationsnummer ansprechbar,

d.h. es können Änderungen an der Konfiguration vorgenommen werden, ohne Datenbankeinträge oder verknüpfte Aliase ändern zu müssen. An der Drei-Spektrometer-Anlage werden hauptsächlich die Analog-Digital-Konverter 4011 und 4017 für die Strahlmonitore und die Auslese der Hallsonden eingesetzt.

Das Steuerungssystem Mezzo stellt einen Gerätetreiber für alle an einem Strang angeschlossenen Module bereit. Während der Initialisierungsphase werden bis zu 16 Module erkannt und die entsprechenden Eigenschaftslisten erzeugt. Eintrag für die Datenbank:

Name	Argument 1	Argument 2
ADAMBUS	Schnittstelle	Baudrate

Beispiel:

```
1 ADAMBUS /dev/ttyS1 9600
```

B.4.1.1 4011, 4012, 4014D

Die Module ADAM 4011, 4012 und 4014D sind Analog-Digital-Konverter mit einem einzigen Kanal. Sie unterscheiden sich durch die Möglichkeit der Temperaturmessung (4011) bzw. eine zusätzliche Sieben-Segment-Anzeige auf dem Modul (4014D). Die Module verfügen zusätzlich über einen Ereigniszähler, der über digitale Eingänge getriggert wird, und sie können Alarmschwellen selbstständig überwachen. Die maximale Auslesefrequenz beträgt 10Hz.

Eigenschaftslisten:

IN, T/C: ADC-Kanal, mit dem die angeschlossene Spannung gemessen wird.

DI0/EV: Digitaler Eingang für den Ereigniszähler.

EC: ADC-Kanal zur Auslese des Ereigniszählers. Der Zähler wird mit dem Befehl *CMD_REQUEST_RESET* zurückgesetzt.

B.4.1.2 4017, 6017

Das Modul ADAM 4017 ist ein 8fach gemultiplexter ADC bei dem die Alarmschwellen nur softwareseitig überwacht werden. Die maximale Auslesefrequenz aller Kanäle beträgt insgesamt 10Hz, d.h. wenn alle ADC-Kanäle aktiviert sind nur etwa 1Hz pro Kanal. Das Modul 6017 ist ein von der Firma μ DAM hergestelltes zum ADAM 4017 kompatibles Modell.

Eigenschaftslisten:

Vin0..Vin7: Anschlüsse der 8 möglichen ADC-Kanäle

B.4.1.3 4050

Bei diesem Modul handelt es sich um einen Digital-IO mit 7 Eingängen und 8 digitalen Ausgängen.

Eigenschaftslisten:

DO: digitaler Ausgang

DI: digitaler Eingang

B.4.1.4 4060

Das Modul ADAM 4060 ist ein 8-Kanal *DO* mit schaltbaren Relais. Die Schaltzeit beträgt 10ms.

Eigenschaftslisten:

DO: digitaler Ausgang

B.4.2 D1000 Drucksensor

Bei der D1000 Serie der Firma DGH Corporation handelt es sich wie beim *ADAM Bus* um intelligente Schnittstellen für verteilte Anwendungen. Kommuniziert wird über eine Zweidrahtleitung mittels RS-485 oder RS-232C bei der jedes Modul eine individuelle Identifikationsnummer erhält.

An der Spektrometeranlage werden die Drucksensoren der D1000 Familie beim Betrieb des Cryo-Targets verwendet.

Der Gerätetreiber erkennt bis zu 16 an einem Strang hängende Module selbständig und erzeugt auch die benötigten Eigenschaftslisten. Eintrag für die Datenbank:

Name	Argument 1	Argument 2
D1000BUS	Schnittstelle	Baudrate

Beispiel:

```
2 D1000BUS /dev/ttyS2 9600
```

Das Modul zur Druckmessung ist sehr ähnlich zum Modul *ADAM 4011* und verfügt auch über die gleichen Möglichkeiten der Überwachung von Alarmschwellen und die Ereigniszählung.

Eigenschaftslisten:

Input: *ADC*-Kanal zur Druckmessung

DI0/EV: Digitaler Eingang für den Ereigniszähler.

EC: ADC-Kanal zur Auslese des Ereigniszählers. Der Zähler wird mit dem Befehl *CMD_REQUEST_RESET* auf 0 zurückgesetzt.

B.4.3 DRC-91CA Temperaturkontroller

Das Cryo-Target der Drei-Spekrometer-Anlage wird mit dem Temperaturkontroller DRC-91CA der Firma LakeShore gesteuert, welcher selbstständig die Targettemperatur nach einem einstellbaren Sollwert regelt. Die Kommunikation mit dem Gerätetreiber erfolgt über die serielle Schnittstelle bei einer festen Geschwindigkeit von 300 Baud. Datenbankeinträge:

Name	Argument
DRC-91CA	Schnittstelle

Beispiel:

```
3 DRC-91CA /dev/ttyS3
```

Der Gerätetreiber stellt zahlreiche Eigenschaftslisten zur Konfiguration des Temperaturkontrollers zur Verfügung:

Mode: Bei diesem Kanal handelt es sich um einen *DAC* über den das Gerät in den lokalen Modus (0), auf remote (1) oder remote mit lokaler Sperre (2) gestellt wird.

Heater Gain: *DAC* mit dem der Parameter *Gain* im erlaubten Bereich zwischen 0.0 und 99.0 vorgegeben wird.

Heater Rate: Stellt den Parameter *Rate* im Bereich zwischen 0.0s und 99s ein

Heater Reset: Vorgabe für den Parameter *Reset*. Der gültige Wertebereich ist zwischen 0.0 und 99.0, was einer Zeit zwischen 0 und 990 Sekunden entspricht.

Heater Power: Dieser *ADC* liest den Parameter *Power* vom Gerät. Der Rückgabewert liegt zwischen 0 und 99 Prozent.

Heater Range: Parameter *Range* im zulässigen Wertebereich (0..5)

SetPoint: Vorgabe (Sollwert) für den Temperaturregler im Gerät

Control Sensor; Display Sensor A0..A4, B0: Temperatur der angeschlossenen Sensoren.

B.4.4 A&D Instruments FG-150 Waage

Das Gewicht der Flaschen für die Gasversorgung der Driftkammern wird mit Waagen des Typs FG-150 der Firma A&D Instruments gemessen. Die Kommunikation erfolgt über die serielle Schnittstelle bei 2400 Baud, 7 Bit, gerader Parität und einem Stopbit. Datenbankeinträge:

Name	Argument
FG150	Schnittstelle

Beispiel:

```
4 FG150 /dev/ttyS4
```

Der Gerätetreiber stellt folgende Kanäle zur Verfügung:

Weight: *ADC*-Kanal zur Auslese des gemessenen Gewichts

Tara: *DAC* mit dem das Tara von der Gewichtsmessung abgezogen wird

B.4.5 ILM200 Cryogen Level Meter

ILM200 ist eine Familie intelligenter Füllstandsmeßgeräte in Systemen die flüssiges Helium oder Stickstoff enthalten. An der Spektrometeranlage wird solch ein Meßgerät am Møller-Polarimeter eingesetzt. Datenbankeinträge:

Name	Argument
ILM2XX	Schnittstelle

Beispiel:

```
5 ILM2XX /dev/ttyS5
```

Auf die Kanäle des Füllstandsmeßgerätes kann über die folgenden Eigenschaftslisten zugegriffen werden:

Level #1..#3: Diese drei *ADC*-Kanäle lesen den Füllstand des Helium- oder Stickstoffgefäßes. Der Rückgabewert ist in Prozent*10 zum maximalen Füllstand angegeben und entspricht den Inhalten der Register R1 bis R3 des Gerätes.

Current #1..#2: *ADC*-Kanäle zur Auslese des Stromes bei der Messung des Heliumfüllstandes (R6 und R7).

Frequency #1..#3: Diese Kanäle liefern die gemessene Frequenz bei der Bestimmung des Stickstofffüllstandes. Der Inhalt entspricht den Registern R11 bis R13.

Valve Position: Position des Nadelventils

Control: Schaltet das Füllstandsmeßgerät in den Modus lokal (0), remote & gesperrt (1) oder remote (3)

Status: 48 Bit *DI*, mit dem Status der drei Kanäle des Gerätes

Bit	Bedeutung
0, 5, 10	Kanal 1, 2, 3 nicht benutzt
1, 6, 11	Kanal benutzt für Stickstoffmessung
2, 7, 12	Helium (gepulster Betrieb)
3, 8, 13	Helium (kontinuierlicher betrieb)
4, 9, 14	Fehler: Sonde nicht angeschlossen
15..40	Status der Füllstandsmessung entsprechend dem Befehl <i>Suu- wwwww</i>
41..48	Zustand der Relais. siehe Befehl <i>Rzz</i>

B.4.6 IPS120-10 Magnetnetzteil

Das IPS120-10 ist ein Netzteil zum Betrieb supraleitender Magnete. Es liefert einen Maximalstrom von 120A bei 10V Ausgangsspannung. Angesprochen wird es über die serielle Schnittstelle bei 9600 Baud. Erforderliche Datenbankeinträge:

Name	Argument
IPS120-10	Schnittstelle

Beispiel:

```
6 IPS120-10 /dev/ttyS6
```

Bei Übertragungsfehlern neigt das Gerät dazu, Befehle falsch zu interpretieren und schickt als Antwort den Auslesewert des Registers *R0* (*Current*). Der Gerätetreiber erkennt dies zwar in den meisten Fällen anhand der Rückantwort, erzeugt aber dennoch einen kurzzeitigen Fehlerzustand bis der entsprechende Kanal erneut ausgelesen wird.

Eigenschaftslisten:

SetPoint: Mit diesem *DAC* wird der Sollwert für den Spulenstrom vorgegeben, den das Netzteil einstellen soll.

Ramp: Hier wird die Rate in Ampere pro Minute festgelegt, mit der der vorgegebene Sollwert angefahren wird.

Current: Über diesen *ADC* wird der augenblickliche Strom des Magnetnetzteils ausgelesen.

Voltage: Aktuelle Ausgangsspannung des Netzteils

Activity: *Activity* ist ein *ADC* der folgende Zustände annehmen kann:

Wert	Bedeutung
0	hold
1	to setpoint
2	to zero
4	clamped

Lock: Lokale Tatstatur wird gesperrt (1) oder freigegeben (0)

Remote: Netzteil auf *remote* (1) oder *lokal* (0) stellen

HeaterSwitch: Dieser *DO* aktiviert (1) oder deaktiviert (0) den *Heater*.

HeaterStatus: Das Feld *HeaterStatus* ist ein 4-Bit *DI* welcher den augenblicklichen Zustand des *Heater* anzeigt:

Wert	Bedeutung
1	off magnet at zero
2	on
4	off magnet at field
8	heater fault

SystemStatus: Dies ist ein 5-Bit *DI* der folgende Werte annehmen kann:

Wert	Bedeutung
1	normal
2	quenched
4	over heated
8	warming up
16	fault

ActivityStatus: Die Eigenschaftsliste *ActivityStatus* ist ein 4-Bit *DI*:

B.4.7 LeCroy 4032a HV-Netzteil

Die Photomultiplier der Szintillatoren von Spektrometer A, B, C und D werden von HV-Geräten des Typs 4032a der Firma LeCroy mit Hochspannung versorgt.

Wert	Bedeutung
1	hold
2	to setpoint
4	to zero
8	clamped

Die Netzteile können auf zwei verschiedene Arten ferngesteuert werden. Einmal über das Camac Interface 2132 oder die serielle Schnittstelle. Die Gerätetreiber im Steuerungssystem Mezzo verwenden ausschließlich die serielle Verbindung, da nur diese Methode eine eindeutige Trennung zwischen Steuerung und Datenerfassung erlaubt. Bis zu 15 an einem Strang verbundene Geräte werden automatisch erkannt. Datenbankeinträge:

Name	Argument 1	Argument 2
LECROY-4032ABUS	Schnittstelle	Baudrate

Beispiel:

```
7 LECROY-4032ABUS /dev/ttyS7 300
```

Die HV-Netzteile kommunizieren bei maximal 300Baud mit dem Gerätetreiber. Dies ist sehr langsam wenn man bedenkt, daß permanent 32 Kanäle überwacht werden müssen: Das Gerät erlaubt die blockweise Auslese aller bzw. die gezielte Abfrage einzelner Kanäle. Da der Kommunikationsaufwand im Einzelabfragemodus erheblich größer ist, wird nur der *Blockmodus* verwendet. Dies führt zu verhältnismäßig langen Antwortzeiten bei Einstellungen an einem einzelnen Kanal, die Steigerung der Gesamtperformance des Gerätetreibers rechtfertigt aber diesen Umstand.

Eigenschaftslisten:

OnOff: Mit dem Feld *OnOff* wird das HV-Netzteil ein- und ausgeschaltet. Änderungen an der Hochspannung einzelner Kanäle im eingeschalteten Zustand führen dazu, daß die HV des entsprechenden Kanals sofort geändert wird und das Netzteil während dieser Zeit nicht ansprechbar ist. Ausgeschaltet wird lediglich der Sollwert angepaßt und beim nächsten Einschalten des Netzteils werden alle Kanäle gleichzeitig auf die vorgegebenen Werte gerampt.

Dieses Verhalten ist eine Eigenschaft des Gerätes und nicht des Softwaretreibers. Wenn versehentlich mehrere Kanäle im eingeschalteten Zustand geändert werden kann das Netzteil unter Umständen für mehrere Minuten nicht ansprechbar sein.

Rate: Legt die Geschwindigkeit fest, mit der die Hochspannung geändert wird.
Zulässige Werte:

Wert	Bedeutung	Wert	Bedeutung
0	1kV/s	5	31V/s
1	500V/s	6	16V/s
2	250V/s	7	8V/s
3	125V/s	8	4V/s
4	62.5V/s		

ADC #0..#31: Dieses Feld enthält den aktuellen Auslesewert der Hochspannung des jeweiligen Kanals.

DAC #0..#31: Mit diesem Feld wird der Sollwert für die Hochspannung vorgegeben.

Zur Ansteuerung des Netzteils über die serielle Schnittstelle wird folgender Adapter benötigt:

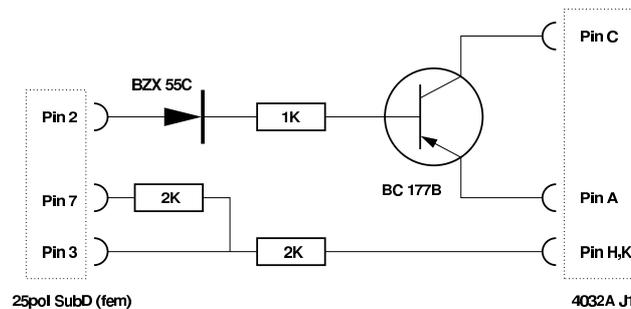


Abb. B.1: Adapter zur Ansteuerung des LeCroy 4032a über die serielle Schnittstelle. Beim 4032a sind an Stecker J1 zusätzlich die Pins L-E sowie B-N zu verbinden.

B.4.8 LeCroy 1440 HV-Netzteil

Das LeCroy 1440 ist ein Netzgerät für die Hochspannungsversorgung von Photomultipliern. Wie das Modul 4032a wird es über die serielle Schnittstelle angesprochen, kann dabei aber mit einer Geschwindigkeit bis zu 9600Baud kommunizieren. Wegen der hohen Zahl an Kanälen wird es wie das 4032a standardmäßig im *Blockmodus* betrieben. Datenbankeinträge:

Name	Argument 1	Argument 2
LECROY-1440BUS	Schnittstelle	Baudrate

Beispiel:

```
8 LECROY-1440BUS /dev/ttyS8 9600
```

Das Netzgerät kann über folgende Eigenschaftslisten angesprochen werden:

OnOff: Mit diesem *DO* wird die Hochspannung für alle Kanäle gleichzeitig ein- (1) bzw. ausgeschaltet (0).

BlockMode: Deaktiviert (0) die blockweise Auslese aller 256 Kanäle des Netzgerätes. Die Voreinstellung ist aktiviert (1).

ADC #0..#255: Aktueller Auslesewert der Hochspannung des jeweiligen Kanals

DAC #0..#255: Vorgabewert (Sollwert) für die Hochspannung

B.4.9 LakeShore LS208 Temperatursensor

Das LakeShore LS208 ist ein 8-Kanal Digitalthermometer zur Messung von Temperaturen zwischen 1.4K und 475K. Angesprochen wird es über die serielle Schnittstelle bei 300Baud. Datenbankeinträge:

Name	Argument
LS208	Schnittstelle

Beispiel:

```
9 LS208 /dev/ttyS9
```

Der Gerätetreiber stellt für jeden Hardwarekanal eine Eigenschaftsliste in Form eines *ADC* zur Verfügung:

ADC #0..#7: Temperaturwert des angegebenen Kanals

B.4.10 MPS8000 Magnetnetzteil

Die Magnete der Drei-Spektrometer-Anlage werden mit Netzgeräten des Typs MPS8000 der Firma Danfysik betrieben. Die Kommunikation erfolgt über die serielle Schnittstelle. Datenbankeinträge:

Name	Argument 1	Argument 2
MPS8000	Schnittstelle	Baudrate

Beispiel:

```
10 MPS8000 /dev/ttyS10 9600
```

Konfiguriert wird das Netzteil mit folgenden Eigenschaftslisten:

SetPoint: Mit diesem *DAC* wird der Sollwert für den Strom eingestellt. Das Netzteil rampt selbstständig auf den vorgegebenen Wert.

Current: Das Feld *Current* enthält die zuletzt ausgelesenen Stellgröße des Stromwertes.

MainPower: Dieser *DO* schaltet das Magnetnetzteil ein (1) oder aus (0).

Mode: Über den *DO Mode* wird das Gerät in den lokal- (1) bzw. remote- (0) Modus gestellt.

Status: *Status* liefert den aktuellen Zustand des Netzteils in Form eines 24Bit *DI*. Die einzelnen Bits haben folgende Bedeutung:

Bit	Bedeutung	Bit	Bedeutung	Bit	Bedeutung
0	Power off	8	Transistor failure	16	Earthleakage
1	Positive Polarity	9	Interlock	17	Thermal Breaker
2	Negative Polarity	10	Overcurrent	18	MPS Temperature
3	Transformer	11	Overload	19	Panicbutton
4	Not used	12	Regulation	20	Waterflow Failure
5	Not used	13	Phase Error	21	Overtemperature
6	Percentage	14	Phase Error	22	Not ready
7	Spare	15	MPS Water	23	Not used

B.4.11 Out-of-Plane-Winkelauslese

Zum Nachweis von Reaktionsprodukten außerhalb der Streuebene kann Spektrometer B um bis zu 10 Grad aus der Ebene gekippt werden. Zu diesem Zweck ruht es auf drei unabhängig voneinander in der Höhe verstellbaren Spindeln. Der Neigungswinkel wird ermittelt, indem die Anzahl Umdrehungen jeder Spindel mittels mechanischer Zählwerke bestimmt und mit einem Polynom in den aktuellen Winkel umgerechnet wird. Abbildung B.2 zeigt die entwickelte Elektronik, um die BCD-Kodierten Zählwerke auszulesen. Angeschlossen wird sie an einen Kleincomputer vom Typ BASIC Briefmarke II der Firma Wilke Technology, der über die serielle Schnittstelle mit dem Gerätetreiber vom Steuerungssystem kommuniziert. Der folgende Code ist das von der Briefmarke ausgeführte Programm:

```

1 ' OOP. BS2
2 '
3 '
4 baud con 84+$0000
5 lcd con 813+$8000
6
7 mux var byte ' Selektierter Kanal vom Multiplexer
8 zaehlwerk var byte ' Schleifenvariable (Nummer des Zaehlwerks)
9 digit var byte ' Schleifenvariable (Nummer der BCD-Zahl)
10 value var word ' eingelesener Wert

```



```

55     getdigit:
56         pause 1
57         if ( dummy = ina ) then nextdigit
58         dummy = ina
59         goto getdigit
60     nextdigit:
61         value = ( value << 4 ) + dummy      ' lese BCD-Zahl
62     next
63
64     result = 0                               ' verdrehe Ergebnis
65     for digit = 0 to 15
66         result = ( result << 1 ) | ( value & 1 )
67         value = value >> 1
68     next
69
70     serout 15,lcd,["LT", hex4 result, " ", cr] ' Ausgabe auf Display
71     serout 6,baud,[ hex4 result, " " ]        ' Ausgabe an PC
72     next
73     serout 6,baud,[ cr ]
74
75     goto loop

```

Datenbankeinträge für den Gerätetreiber:

Name	Argument
OOP	Schnittstelle

Beispiel:

```
11 OOP /dev/ttyS11
```

Der Treiber hat drei *ADC*-Kanäle zur Abfrage des aktuellen Zählstandes:

Swivel[0..2]: Stand des *n*ten Zählwerkes

B.4.12 Parallelport

Dieser Gerätetreiber nutzt die Eingangsleitungen eines handelsüblichen Parallelports und exportiert diese in Form eines 8-Bit *DI*. Datenbankeinträge:

Name
PARPORT

Beispiel:

```
12 PARPORT
```

Eigenschaftslisten:

DI: 8-Bit digitaler Eingang

B.4.13 PC20 MultiIO PC-Meßkarte

Die PC-Einsteckkarte PC20 der Firma BMC bietet 16 *ADC*- und zwei *DAC*-Kanäle mit einer Auflösung von 12Bit. Außerdem besitzt sie 2*16 digitale Ein- bzw. Ausgänge die in 8er Gruppen programmierbar sind. Der Gerätetreiber besteht aus zwei Teilen: Einem Kernelmodul unter Linux für die Ansteuerung der Einsteckkarte und der Schnittstelle zum zentralen Prozeß.

Die Karte benötigt freie Ports im PC bei den Startadressen \$200, \$210, \$220, \$230, \$240, \$250, \$260, \$270 oder \$280. Zusätzliche Interrupts sind nicht erforderlich. Bis zu vier dieser Meßkarten können gleichzeitig eingesetzt werden. Der Betrieb unter Linux erfordert zusätzlich folgende Eintragungen:

```
mknod /dev/pc200 c 253 0
mknod /dev/pc201 c 253 1
mknod /dev/pc201 c 253 2
mknod /dev/pc203 c 253 3
chmod 666 /dev/pc200
chmod 666 /dev/pc201
chmod 666 /dev/pc202
chmod 666 /dev/pc203
```

Die Datei *modules.conf* muß um diese beiden Zeilen erweitert werden:

```
alias char-major-253 pc20
options pc20 ADDRIO=0x210,0x240 PC20_MAJOR=253
```

Das Argument *ADDRIO* enthält die durch Kommata getrennten Portnummern der verwendeten PC-Einsteckkarten.

Die Schnittstelle zum zentralen Prozeß ist im Aufbau identisch zu den anderen Gerätetreibern und erfordert auch vergleichbare Datenbankeinträge:

Name	Argument 1	Argument 2
PC20	Schnittstelle	Konfiguration der digitalen Ein- und Ausgänge 0x8080 = <i>DO DO</i> 0x9090 = <i>DI DO</i> 0x8282 = <i>DO DI</i> 0x9292 = <i>DI DI</i>

Beispiel:

```
13 PC20 /dev/pc200 0x9090
```

Je nachdem welche Konfiguration angegeben wurde, stellt der Gerätetreiber unterschiedliche Eigenschaftslisten zur Verfügung:

ADC #0..#15: Wert der zuletzt am analogen Eingang der Karte gemessenen Spannung.

DAC #0..#1: Analoger Ausgang der Karte

DI #0..#1: 16Bit digitaler Eingang

DO #0..#1: 16Bit digitaler Ausgang

Zu Beachten ist, daß die analogen und digitalen Ausgänge der Einsteckkarte nach einem Reset des Computers auf Null gesetzt werden.

B.4.14 NRVD Leistungsmessgerät

In der A1-Strahlführung befinden sich zwei *Cavities* um die Leistung des Elektronenstrahls zu ermitteln und damit den Strom bestimmen zu können. Die Messung erfolgt mit dem Zweikanal-Leistungsmesser NRVD der Firma Rohde&Schwarz. Das Gerät wird über eine IEEE-488 Schnittstelle angesprochen und ist mittels eines IEEE-488 nach RS-232 Konverters an die serielle Schnittstelle eines PC's angeschlossen. Datenbankeinträge:

Name	Argument 1	Argument 2
NRVD	Schnittstelle	IEEE-488 Gerätenummer

Beispiel:

```
14 NRVD /dev/ttyS3 12
```

Es ist wichtig, daß der Treiber für den Konverter eine kleinere Startnummer als der für das Leistungsmeßgerät hat, da das NRVD sonst nicht gefunden wird. Der Gerätetreiber stellt zwei *ADC*-Kanäle für die Auslese zur Verfügung:

A: Kanal A des Leistungsmeßgeräts

B: Kanal B des Leistungsmeßgeräts

B.4.15 Programmierbares Netzteil DIGI-35-CPU

Dieses programmierbare Niederspannungsnetzteil steuert das Magnetnetzteil für den Korrekturmagneten am Ausgangsstrahlrohr. Die Kommunikation erfolgt über die serielle Schnittstelle. Datenbankeinträge:

Name	Argument
DIGI35	Schnittstelle

Beispiel:

```
15 DIGI35 /dev/ttyS15
```

Der Gerätetreiber exportiert zwei Eigenschaftslisten über die der maximale Strom und die Spannung gesetzt werden können:

Current: Mit diesem *DAC* wird der Ausgangsstrom festgelegt.

Voltage: Das Feld *Voltage* definiert die Ausgangsspannung.

B.4.16 WCSI Winkelauslese

Bei diesem Modul handelt es sich um die Auslese der Winkelkodierschiene für die drei Spektrometer. Das Gerät sendet periodisch die aktuelle Position über die serielle Schnittstelle an den PC. Datenbankeinträge:

Name	Argument
WCSI	Schnittstelle

Beispiel:

```
16 WCSI /dev/ttyS16
```

Der Treiber stellt folgende Eigenschaftslisten zur Verfügung:

Angle: Dieser *ADC* enthält den zuletzt vom Gerät geschickten Winkel.

Status: Status des zuletzt übermittelten Wertes:

Wert	Bedeutung
0	kein Fehler
1	Kodierschiene falsch ausgerichtet
2	Hardwarefehler
4	Prüfsummen / Übertragungsfehler

B.4.17 KPH Wasserüberwachungssystem

Das Wasserüberwachungssystem ist ein von der Elektronikwerkstatt des Instituts entwickeltes Gerät zur Kontrolle des Durchflusses von bis zu 15 Wasseruhren. Es prüft die gemessenen Wasserflüsse auf Unterschreiten eines Minimal- oder Überschreiten eines Maximalwertes und löst gegebenenfalls einen Interlock aus. Bedient wird es über eine lokale Tastatur oder die serielle Schnittstelle. Datenbankeinträge:

Beispiel:

Name	Argument
WATERMETER	Schnittstelle

```
17 WATERMETER /dev/ttyS17
```

Für jede angeschlossene Wasseruhr gibt es eine Reihe Eigenschaftlisten mit denen die Uhr konfiguriert werden kann:

Type #1..#15: Mit diesem Feld wird der Typ der Wasseruhr festgelegt. Erlaubt sind die Werte 1, 2, 3, 6, 10 entsprechend einem Fluß von 1.5, 2.5, 3.5, 6 und 10 m³.

Flow #1..#15: Zuletzt gemessener Durchfluß der Wasseruhr in l/h.

LimitLow #1..#15: Beim Unterschreiten der Schwelle in *LimitLow* wird ein Interlock ausgelöst.

LimitHigh #1..#15: Obere Schwelle. siehe auch *LimitLow*

Interlock #1..#15: Dieser digitale Eingang signalisiert einen Interlock. Mit dem Befehl *CMD_REQUEST_RESET* kann der Interlock gelöscht werden.

B.4.18 Kurzschlußkompensation für Spektrometer C

Durch einen Kurzschluß in einer der Spulen von Dipol 1 bei Spektrometer C ist es erforderlich, den stark schwankenden Strom gesondert zu regulieren. Die zugehörige Elektronik und ihre Software wurde im Hause entwickelt. Bei dem vorliegenden Gerätetreiber handelt es sich lediglich um eine Portierung dieser Software nach Linux, um im Steuerungssystem Mezzo verwendet werden zu können. Datenbankeinträge:

Name	Argument 1	Argument 2
SCC	serielle Schnittstelle	PC20 Gerätetreiber

Beispiel:

```
18 SCC /dev/ttyS18 /dev/pc200
```

Der Gerätetreiber für die Kurzschlußkompensation kommuniziert mit drei untergeordneten Geräten: Einem digitalen Voltmeter zur Messung der Hallspannung einer Magnetfeldsonde (RS232), einem Netzgerät zur Korrektur des Spulenstromes von Dipol 1 (PC20) sowie einem Plotter, um die Magnetfeldschwankungen aufzuzeichnen (PC20). Die drei Geräte werden nicht einzeln angesprochen, da der Regelkreis so schnell wie möglich arbeiten soll und die Kommunikation über Ethernet das Netzwerk zu stark belastet hätte. Stattdessen gibt es einen einzigen

Treiber, dessen Stellgrößen und Parameter über den zentralen Prozeß konfigurierbar sind:

Difference: Dieser *ADC* liefert den Differenzwert zwischen der Auslese des DVM und dem Referenzwert, normiert auf den Referenzwert. Es werden alle Werte zwischen zwei Abfragezyklen durch den zentralen Prozeß aufintegriert.

Percentage: Hier wird die prozentuale Ausgangsleistung des Netzgerätes abgeleitet. Sobald der Wert $\pm 100\%$ übersteigt oder unterschreitet, kann die Abweichung des Stromes vom Referenzwert nicht mehr kompensiert werden. In diesem Fall muß der Sollwert des Hauptnetzteils korrigiert werden. Dies kann bei laufender Regelung erfolgen.

Digital Volt Meter: Diese Eigenschaftsliste enthält den seit der letzten Abfrage durch den zentralen Prozeß aufintegrierten Auslesewert des DVM.

Reference: Mit diesem *DAC* wird der Referenzwert eingestellt, auf den das System regeln soll.

Control: Aktiviert (*1*) oder deaktiviert (*0*) den Regelkreis.

B.4.19 PT2025 NMR Auslese

Das PT2025 ist ein Gerät zur präzisen Messung von Magnetfeldern mittels der Kernresonanzmethode. Es können bis zu fünf Wasserstoffproben für einen Gesamtmeßbereich von 0.043T bis 2.1T angeschlossen werden. Ein zusätzlicher Multiplexer ermöglicht den Betrieb mehrerer Sonden für verschiedene Magnete; zu einem beliebigen Zeitpunkt kann jedoch immer nur eine Magnetfeldmessung stattfinden. Angesprochen wird es über die serielle Schnittstelle. Datenbankeinträge:

Name	Argument 1	Argument 2
PT2025	Schnittstelle	Baudrate

Beispiel:

```
19 PT2025 /dev/ttyS19 9600
```

Der Gerätetreiber erlaubt das PT2025 zu konfigurieren und eine Messung zu starten. Die notwendigen Vorkehrungen für die Messung werden über spezielle Skripte außerhalb des Treibers vorgenommen. Folgende Eigenschaftslisten stehen zur Verfügung:

Mode: Mit diesem *DAC* wird das Gerät in den lokalen Modus (*0*), lokal mit verriegelter Tastatur (*1*) oder den remote Modus (*2*) geschaltet.

Auto: Versetzt das PT2025 in den manuellen (0) oder automatischen Modus (1). Im Automatikmodus folgt das Gerät dem NMR-Signal bis zu einer Drift von 5

Display Unit: Hiermit wird die Anzeige von MHz (0) auf Tesla (1) gestellt.

Field Sense: Diese Eigenschaftsliste legt die Feldrichtung fest: (0) = negativ, (1) = positiv

Search Time: Die Geschwindigkeit mit der das NMR-Signal gesucht wird kann mit diesem *DAC* vorgegeben werden. Je größer die Zahl (1..6), desto langsamer und genauer ist die Suche.

Mux2032: Bis zu acht unterschiedliche Probensets können an den Multiplexer angeschlossen werden. Ausgewählt werden sie mit dem Eigenschaftsfeld *Mux2032*.

Range: Die Eigenschaftsliste legt die Anzahl der zu prüfenden Proben bei der Signalsuche fest.

Multiplexer: Mit diesem *DAC* wird die Startprobe für die nächste Messung definiert.

Status: *Status* ist ein 24Bit *DI* mit den Inhalten der Register S1 bis S3 des Gerätes.

SearchMode: Mit einem Schreibbefehl auf diesen *DAC* wird die NMR-Messung gestartet. Der anzugebende Wert ist die Startfrequenz für die Signalsuche im Bereich vom 0..4095.

Value: Hier wird der zuletzt gemessene Wert abgelegt.

B.4.20 Phoenix Contact Interbus-S

Die Kommunikation zwischen dem Steuerungssystem und den Teilnehmern des Interbus-S erfolgt über mehrere Anschaltbaugruppen des Typs *IBS PC ISA SC QS UM* in einem der Kontrollrechner. Die Einsteckkarte wird über einen Linux-Kerneltreiber angesprochen. Benötigt werden dafür mehrere Einträge:

```
mknod /dev/ibb0n0_d c 254 0;  chmod 666 /dev/ibb0n0_d
mknod /dev/ibb0n0_m c 254 1;  chmod 666 /dev/ibb0n0_m
mknod /dev/ibb0n1_d c 254 2;  chmod 666 /dev/ibb0n1_d
mknod /dev/ibb0n1_m c 254 3;  chmod 666 /dev/ibb0n1_m
mknod /dev/ibb0n2_d c 254 4;  chmod 666 /dev/ibb0n2_d
mknod /dev/ibb0n2_m c 254 5;  chmod 666 /dev/ibb0n2_m
mknod /dev/ibb0n3_d c 254 6;  chmod 666 /dev/ibb0n3_d
mknod /dev/ibb0n3_m c 254 7;  chmod 666 /dev/ibb0n3_m
```

```

mknod /dev/ibb1n0_d c 254 8;  chmod 666 /dev/ibb1n0_d
mknod /dev/ibb1n0_m c 254 9;  chmod 666 /dev/ibb1n0_m
*      *      * * *      *      *      *
mknod /dev/ibb3n3_d c 254 30; chmod 666 /dev/ibb3n3_d
mknod /dev/ibb3n3_m c 254 31; chmod 666 /dev/ibb3n3_m

```

In der Datei *modules.conf* müssen folgende Zeilen hinzugefügt werden:

```

alias char-major-254 ibs
options ibs IRQ=5,7 ADDRIO=0x120,0x128 ADDRMPM=0xD000,0xD100

```

Der Parameter *IRQ* enthält eine Liste der IRQs jeder Anschaltbaugruppe, *ADDRIO* die Portadressen und der Parameter *ADDRMPM* die Adressen für das *Memory Mapped IO*. Der Treiber unterstützt kein IRQ-Sharing.

Die Schnittstelle zum Steuerungssystem erfordert folgende Datenbankeinträge pro Anschaltbaugruppe:

Name	Argument
INTERBUS	Nummer der Anschaltbaugruppe

Beispiel:

```
20 INTERBUS 1
```

Der Gerätetreiber exportiert lediglich eine Eigenschaftsliste die den Status des Bussystems enthält.

Diagnostic: 16Bit *DI* mit dem Status des Bussystems

B.4.20.1 Bruker Magnetnetzteil

Der Magnet von Spektrometer D (Short-Orbit-Spektrometer) wird von einem Netzteil der Firma Bruker versorgt, welches über ein in der Elektronikwerkstatt des Instituts entwickeltes Interbus-Interface angesprochen wird. Datenbankeinträge:

Name	Argument 1	Argument 2
IBSBRUKER	Anschaltbaugruppe	Nummer des Teilnehmers

Beispiel:

```
21 IBSBRUKER 1 6
```

SetPoint: Über diesen *DAC* wird der Sollwert für den Strom vorgegeben und mit dem Befehl *CMD_REQUEST_TRIGGER* eingestellt. Zu beachten ist, daß ein Reset auf dem Interbus den Vorgabewert auf Null setzt. Solange dieser Wert nicht getriggert wird, bleibt der Strom den das Netzteil liefert unverändert.

Status: *DI* mit den Statusbits des Netzteils

B.4.20.2 Kollimator-Kontroller

Die Motoren für die Kollimatoren der Spektrometer und die Mechanik des Targets werden über Interbusmodule aus dem Hause gesteuert. Dabei fahren sie selbständig eine vorgegebene Zielposition an. Datenbankeinträge:

Name	Argument 1	Argument 2
IBSCOLLI	Anschaltbaugruppe	Nummer des Teilnehmers

Beispiel:

```
22 IBSCOLLI 1 8
```

Position: Dieser *ADC* liefert die aktuelle Position der Motorsteuerung. Durch senden des Befehls *CMD_REQUEST_TRIGGER* kann ein Stoppsignal erzwungen werden.

SetPoint: Hier wird die Zielposition vorgegeben und anschließend mit dem Befehl *CMD_REQUEST_TRIGGER* eingestellt. Ein Reset des Interbus löscht den Vorgabewert, der Motor bewegt sich aber weiter solange kein Triggersignal kommt.

Status: Zuletzt ermittelter Status der Motorsteuerung:

Bit	Bedeutung
0	unterer Endschalter gedrückt
1	oberer Endschalter gedrückt
2	Remote-Modus
3	Motor bewegt sich

Control: Hier kann manuell ein Handshake ausgelöst werden. Wird nur für Testzwecke verwendet.

B.4.20.3 MultiIO-Kontroller

Dieses Interbus Modul ist universell einsetzbar und wird an der Spektrometeranlage für die Steuerung der Driftkammernetzteile und die Vakuumauslese verwendet. Datenbankeinträge:

Name	Argument 1	Argument 2
IBSMULTIIO	Anschaltbaugruppe	Nummer des Teilnehmers

Beispiel:

23 IBSMULTIIO 1 3

Eigenschaftslisten:

ADC #0..#4: ADC-Kanäle mit einem Meßbereich von 0 - 10V

OVERRANGE ADC #0..#4: Dieser 1Bit *DI* signalisiert eine Meßbereichsüberschreitung des zugehörigen *ADC*.

POLARITY ADC #0..#4: Vorzeichen des entsprechenden Kanals.

DAC #0..#2: *DAC* mit einem Ausgangsspannungsbereich von 0 - 10V. *DAC* Nummer #2 hat einen invertierten Ausgang.

DI #0..#3: 1Bit digitaler Eingang

DO #0..#1: 1Bit digitaler Ausgang

B.4.20.4 NMR Kompensation

Das Magnetfeld in Spektrometer B hat einen zu großen Gradienten um eine Feldmessung mittels NMR durchzuführen. Ein zusätzliches Paar Spulen um die NMR-Sonden kompensiert diese Feldinhomogenitäten. Die Kommunikation mit Netzgerät für die Kompensation erfolgt über ein Interbus Modul. Datenbankeinträge:

Name	Argument 1	Argument 2
IBSNMRCOMP	Anschaltbaugruppe	Nummer des Teilnehmers

Beispiel:

24 IBSNMRCOMP 1 2

Eigenschaftsfelder:

DAC #0..#1: Dieser *DAC* legt den Strom für das Spulenpaar fest.

Polarity: Vorgabe der Magnetfeldrichtung

Probe: Mit dieser Eigenschaftsliste wird eine der vier Sonden für die NMR-Messung ausgewählt.

B.4.20.5 VDC Netzteil

Mit je zwei Delta-Netzgeräten werden die Vorverstärkerkarten für die Driftkammern betrieben. Ein Interbusmodul übernimmt die Kommunikation mit dem Steuerungssystem. Datenbankeinträge:

Beispiel:

Name	Argument 1	Argument 2
IBSVDCPOWER	Anschaltbaugruppe	Nummer des Teilnehmers

25 IBSVDCPOWER 1 9

Eigenschaftslisten:

V-Monitor 1..2: Spannungsmonitor des Netzteils *ADC*

I-Monitor 1..2: Strommonitor des Netzteils *ADC*

DAC A..D: Mit diesem *DAC* wird der Sollwert für die Schwelle eingestellt.

Power A..D: Schwellen-Stellgröße *ADC*

Status 1..2: Diese Eigenschaftslisten (*DI*) enthalten den aktuellen Status der beiden Netzteile.

Bit	Bedeutung
0	CC (Operation)
1	OT (OverTemperature)
2	LIM (Current or Voltage Limit)
3	DCF (DC Failure)
4	ACF (AC Failure)

B.4.21 Datenerfassung AQUA

Im Rahmen dieser Arbeit erfolgte eine Trennung der Hardware zwischen Datenerfassung und Steuerung. Dennoch wird das Steuerungssystem verwendet, um die Konfiguration für die Triggerelektronik in der Datenbank abzulegen und die eingestellten und zurückgelesenen Werte miteinander zu vergleichen.

Aus Sicht der Steuerung ist die Datenerfassung eine Komponente wie jedes andere Gerät auch. Der Zugriff auf die Hardware erfolgt allerdings nicht vom Gerätetreiber aus sondern über eine verborgene TCP/IP Verbindung mittels der *AQUA* und *Mezzo* Daten austauschen. Diese Verbindung darf nicht verwechselt werden mit der Möglichkeit, daß *AQUA* Steuerungsdaten in den Detenstrom einbettet. Dies erfolgt auf einem anderen Weg. Datenbankeinträge:

Name	Argument
AQUA	Portnummer für die zusätzliche Verbindung

Beispiel:

26 AQUA 5003

Der Gerätetreiber kann leicht um Eigenschaftslisten erweitert werden. Im Sourcecode gibt es eine Tabelle *g_AquaChannel* in die alle von *AQUA* benötigten Kanäle eingetragen werden. Das erste Element eines Eintrags ist der Name des Kanals und das zweite Element die Bitbreite. Eine Breite von *-1* signalisiert ein Textfeld. Die einzige vom Treiber vorgegebene Eigenschaftsliste ist *State*. Dort kann die Anzahl der zusätzlichen TCP/IP Verbindungen abgefragt werden.

B.5 Skriptsprache

Komplexe Einstellvorgänge oder Abläufe, wie beispielsweise die Messung des Magnetfeldes mit der NMR-Sonde, werden mit Hilfe spezieller Skripte durchgeführt. Diese Skripte sind Textdateien, deren Code durch das Programm *Script* kompiliert und anschließend ausgeführt wird.

Die Skriptsprache ähnelt in Syntax und Semantik der Programmiersprache C, wengleich sehr viele Besonderheiten dieser Sprache weggelassen wurden.

Ein Programm besteht aus einer Folge von Anweisungen. Optional können eigene Funktionen und Prozeduren der Anweisungsliste vorangestellt werden. Zusätzliche Dateien werden mit dem Befehl *#include "dateiname"* eingebunden. Der folgende Programmcode zeigt ein sehr einfaches Beispiel für ein Skript:

```

1 // Deklaration einer Prozedur
2 procedure print_hv( number, hv ) {
3     printf( "HV[%d] = %dV\n", number, hv );
4 }
5
6 // Variablendeklaration
7 var i;
8
9 // In einer Schleife werden Hochspannungen ausgedruckt
10 for ( i = 0; i < 10; i = i + 1 ) {
11     print_hv( i, a.trigger.hv[i].get );
12 }

```

B.5.1 Variablen und Variablendeklaration

Variablen werden durch die Angabe des Schlüsselwortes *var* und einer durch Kommata separierten Liste von Bezeichnern deklariert. Für Arrays muß die Anzahl der Elemente in eckigen Klammern ([]) angegeben werden. Variablen haben keinen fest vorgegebenen Typ wie beispielsweise *double* oder *string*. Erst wenn der Variablen zur Laufzeit des Programms ein Wert zugewiesen wird, steht der Typ fest. Beispiel:

```
var i, test[10];
```

Wurde eine Variable über *var* deklariert, so handelt es sich um eine *lokale* Variable. Alle Aliase des zentralen Prozesses sind dem Skript als *globale* Variablen bekannt.

Auf das Feld eines Alias kann zugegriffen werden, indem der Aliasname gefolgt von einem Doppelpunkt und dem Feldbezeichner angegeben wird. Enthält der Aliasbezeichner eine eckige Klammer (`[]`), so wird er behandelt wie ein Array und es sind arithmetische Operationen zwischen den Klammern erlaubt um bestimmte Elemente indizieren zu können. Das Ergebnis dieser Operation ist immer eine Zahl ohne führende Null. Aliasnamen wie beispielsweise `test[05]` sind demnach nicht erlaubt.

B.5.2 Prozeduren und Funktionen

Prozeduren und Funktionen werden durch Angabe des Schlüsselwortes *procedure* oder *function* gefolgt von dem Funktionsnamen, einer optionalen Argumentenliste und schliesslich einer Anweisungsliste deklariert. Der Unterschied zwischen einer Funktion und einer Prozedur besteht darin, daß Funktionen einen Wert zurückliefern, der innerhalb der Anweisungsliste mit dem Schlüsselwort *return* angegeben wurde. Wird in der Argumentenliste einem Variablenbezeichner das Zeichen `&` vorangestellt, so wird die Variable statt „call by value“ als „call by reference“ behandelt. Beispiel:

```
function beispiel( test1 , @test2 )
{
    test2 = test1 ;
    return test1 * test2 ;
}
```

B.5.3 Anweisungen und Kontrollstrukturen

Eine Anweisungsliste besteht aus Variablenzuweisungen, bedingten Anweisungen, Programmschleifen oder Kontrollstrukturen:

for (.. ; .. ; ..) Programmschleife, die einen Anweisungsblock bis zum Erfüllen einer Abbruchbedingung ausführt.

```
for ( i = 0, j = 0; i < 100; i = i + 1, j = j + 2 ) {
    printf( "%d\n", i );
}
```

while (..) Programmschleife, die einen Anweisungsblock ausführt, solange die Abbruchbedingung nicht erfüllt ist.

```
i = 0;
while ( i < 100 ) {
    printf( "%d\n", i );
    i = i + 1;
}
```

if (..) Anweisungsblock der ausgeführt wird, wenn eine Bedingung zutrifft

```
if ( i > 100 ) {
    printf( "%d\n", i );
}
```

if (..) else Anweisungsblöcke die ausgeführt werden, wenn eine Bedingung zutrifft/nicht zutrifft

```
if ( i > 100 ) {
    printf( "%d\n", i );
} else {
    printf( "Pech gehabt!\n" );
}
```

printf (..) Formatierte Textausgabe analog zur Funktion *printf* in C

return <wert> Nach der *return*-Anweisung kehrt das Programm in die übergeordnete Anweisungsliste zurück. Sie wird verwendet in Prozeduren und Funktionen. Letztere erwarten als Argument den Rückgabewert für die Funktion.

break Vorzeitiger Schleifenabbruch

```
for ( i = 0, j = 0; i < 100; i = i + 1, j = j + 2 ) {
    printf( "%d\n", i );
    if ( j > 40 )
        break;
}
```

sleep (..) Das Programm schläft für die angegebene Zeitspanne (Millisekunden)

trigger (..) Sendet den Befehl *CMD_REQUEST_TRIGGER* an das angegebene Objekt.

reset (..) Sendet den Befehl *CMD_REQUEST_RESET* an das angegebene Objekt

update (..) Sendet den Befehl *CMD_REQUEST_UPDATE* an das angegebene Objekt

lock (..) Die angegebene Semaphore wird für den weiteren Zugriff gesperrt. Wenn sie bereits markiert ist, wartet das Programm bis sie über *unlock* freigegeben wurde. Rekursive Aufrufe von *lock* sind nicht zulässig.

unlock (..) Entriegelt die angegebene Semaphore

wait (.. , ..) Diese Funktion wartet die vorgegebene Zeitspanne auf eine Aktualisierungsmeldung vom gewünschten Objekt. Der Rückgabewert ist 1, wenn innerhalb der Zeitspanne eine Meldung eintrifft, ansonsten 0.

```
if ( wait( beam.foerster.multi, 1000 ) ) {
    printf( "%f\n", beam.foerster.multi );
}
```

bisection (.. , .. , .. , ..) Invertiert die angegebene Funktion nach dem Bisektionsverfahren. Die zu invertierende Funktion darf nur von einem Funktionswert abhängen. Das erste Argument ist der Funktionsname, das zweite der aktuelle Funktionswert und die letzten beiden Argumente sind der untere und obere Grenzwert. Rückgabewert ist der ermittelte Funktionswert.

```
function test( x ) {  
    return 2 * x;  
}  
  
printf( "%f\n", bisection( test , 1.0 , 0.0 , 4.0 ) );
```

time () Diese Funktion liefert die aktuelle Unixzeit zurück.

Folgende arithmetische Operationen und Anweisungen sind in den Skripten erlaubt: +, -, *, /, ^, |, &, *ln*, *log*, *exp*, *fabs*, *sin*, *cos*, *tan*, *ceil*, *floor*

Literaturverzeichnis

- [Am79] E.Amaldi, S.Fubini, and G.Furlan, Pion-Electroproduction, Springer Tracts in Modern Physics #83 (1979)
- [Be98] V. Bernard, N. Kaiser and Ulf-G. Meißner, Threshold neutral pion electroproduction in heavy baryon chiral perturbation theory, Nucl. Phys. A 607 (1996) 379-401, Erratum Nucl. Phys. A 633 (4) (1998) 695-697
- [Bec90] R.Beck et al., Measurement of the $p(\gamma, \pi^0)$ cross section at threshold, Phys. Rev. Lett. 65 (1990) 1841
- [Bec96] R.Beck et al., Neutral pion photoproduction from the proton near threshold, Physics Letters B 368 (1996) 20-25
- [Ber74] J.Bernstein, Spontaneous symmetry breaking, gauge theories, the Higgs mechanism and all that, Rev. Mod. Phys. 46 (1974) 7-48
- [Ber94] A.M.Bernstein, H.Merkel, D.Newton, Threshold π^0 Electroproduction, Proposal for an Experiment
- [Ber97] A. M. Bernstein et al., Observation of a unitary cusp in the threshold $\gamma p \rightarrow \pi^0 p$ reaction, Phys. Rev. C 55 (1997) 1509
- [Blo98] K. I. Blomqvist et al., The three-spectrometer facility at the Mainz microtron MAMI, Nucl. Instr. and Meth. A 403 (1998) 263-301
- [Bri95] H.B. van den Brink, Phys. Rev. Lett. 74 (1995) 3561
- [Dis01] Michael O. Distler, Harald Merkel, Markus Weis, Data Acquisition and Analysis for the 3-Spectrometer-Setup at MAMI, Proceedings of the 12th IEEE Real Time Congress on Nuclear and Plasma Sciences, Valencia, Juni 2001
- [Dis90] Michael O. Distler, Aufbau und Test einer vertikalen Driftkammer, Diplomarbeit, Institut für Kernphysik der Universität Mainz, 1990
- [Dis96] Michael O. Distler, Elektroproduktion von neutralen Pionen am Wasserstoff an der Schwelle, Doktorarbeit, Institut für Kernphysik der Universität Mainz, 1996
- [DMT01] S.S. Kamalov, S.N. Yang, D. Drechsel, L. Tiator, A Dynamical Model for Pion Photo- and Electroproduction on the Nucleon, Phys. Rev. C 64 (2001) 032201
- [Dre92] D.Drechsel, L. Tiator, Threshold pion photoproduction on nucleons, J.Phys.G: Nucl. Part. Phys. 18(1992) 449-497

- [Ewa96] Ingo Ewald, Entwicklung und Erprobung einer langen, dünnen Flüssig-Wasserstoff-Targetzelle, Diplomarbeit, Institut für Kernphysik der Universität Mainz, 1996
- [Jan98] Werner Jansen, Wolfgang Blome, Interbus, Verlag Moderne Industrie, Bibliothek der Technik, Band 162, Landsberg/Lech 1998
- [Jen03] Peter Jennewein, Kompensation für den Kurzschluß in Dipol 1 von Spektrometer C, private Mitteilung, Institut für Kernphysik der Universität Mainz, 2003
- [Kor94] Manfred Korn, Entwicklung des Bahnrückverfolgungsverfahrens für die Drei-Spektrometer-Anlage und experimentelle Bestimmung der Abbildungseigenschaften der Spektrometer A und B mit elastischer Elektronstreuung, Doktorarbeit, Institut für Kernphysik der Universität Mainz, 1994
- [Kra95] Helmut Kramer, Grundlagen für das Steuerungs- und Überwachungssystem der Drei-Spektrometer-Anlage am Elektronenbeschleuniger MAMI. Doktorarbeit, Institut für Kernphysik der Universität Mainz, 1995
- [Kry96] Klaus Werner Krygier, Konzeption und Realisierung eines Datenerfassungssystems für die Experimente am Mainzer Mikrotron und Inbetriebnahme für Koinzidenzexperimente mit virtuellen Photonen an der Drei-Spektrometer-Anlage, Doktorarbeit, Institut für Kernphysik der Universität Mainz, 1996
- [Kun96] Volker Kunde, Aufbau des Experimentsteuerungssystems der Dreispektrometieranlage und Messung von $^{12}\text{C}(e, e'x)$ mit der BGO-Kristallkugel. Doktorarbeit, Institut für Kernphysik der Universität Mainz, 1996
- [MAI00] D. Drechsel, O. Hanstein, S.S. Kamalov, L. Tiator, A Unitary Isobar Model for Pion Photo- and Electroproduction on the Nucleon, Nucl. Phys. A645 (1999) 145-174
- [Mar93] Christoph Martin, Implementation und Test der A1-Targetsystemsoftware und Integration in das Gesamtsystem für die Kontrolle von (e, e'p)-Messungen. Diplomarbeit, Institut für Kernphysik der Universität Mainz, 1993
- [Mam94] J. Ahrens, H. Backe, D. von Harrach, K.-H. Kaiser, F. Klein, R. Neuhausen, Th. Walcher, The Mainz Microtron MAMI: A facility portrait with a glimpse at first results, Nuclear Physics News, Vol.4, No.2, (1994)
- [Maz86] E.Mazzucato et al., Phys. Rev. Lett. 57 (1986) 3144

- [Me96] Dispersion-Theoretical Analysis of the Nucleon Electromagnetic Formfactors, P. Mergell, Ulf-G. Meißner, D. Drechsel, Nucl.Phys. A596 (1996) 367-396
- [Mer02] Harald Merkel, Neutral Pion Threshold Production at $Q^2 = 0.05\text{GeV}^2/c^2$ and Chiral Perturbation Theory, Physical Review Letters, 2002, 012301 1-4
- [Neu89] Rainer Neuhausen, Meßmethoden der Physik, Vorlesungsskript, Institut für Kernphysik der Universität Mainz, WS 1988/89
- [Pos00] Thomas Pospischil, Aufbau und Inbetriebnahme eines Protonen-Polarimeters an MAMI und Messung der Proton-Polarisation in der Reaktion $p(\vec{e}, e'\vec{p})\pi^0$ in paralleler Kinematik im Bereich der $\Delta(1232)$ -Resonanz. Doktorarbeit, Institut für Kernphysik der Universität Mainz, 2000
- [Pos02a] Th. Pospischil et al, The focal plane proton-polarimeter of the 3-spectrometer setup at MAMI, Nucl. Instr. and Meth. A 483 (2002) 713-725
- [Pos02b] Th. Pospischil et al, The horizontal drift chambers for the focal plane proton-polarimeter of the 3-spectrometer setup at MAMI, Nucl. Instr. and Meth. A 483 (2002) 726-733
- [Sch91] S. Scherer, J.H. Koch, Pion electroproduction at threshold and PCAC, Nuclear Physics A534 (1991) 461-477
- [Sch93] S. Scherer, J.H. Koch, J.L.Friar, Systematics of low-energy theorems for pion photo- and electroproduction, Nuclear Physics A552 (1993) 515-522
- [Ste93] Stefan Steffens, Entwicklung und Erprobung einer „X-Windows“-Benutzeroberfläche zur rechnergesteuerten und -kontrollierten Durchführung von Koinzidenzexperimenten an MAMI. Diplomarbeit, Institut für Kernphysik der Universität Mainz, 1993
- [Wel92] T.P. Welch et al, Phys. Rev Lett 69 (1992) 2761

Danksagung

An dieser Stelle möchte ich mich bei all denen bedanken, die zum Gelingen dieser umfangreichen Arbeit beigetragen haben.

Herrn Prof. Dr. Th. Walcher danke ich für die interessante und abwechslungsreiche Themenstellung sowie den großen Freiraum, den er mir bei der Umsetzung der Arbeit gab.

Den Containerkindern Daniel Elsner, Peter Bartsch und Andre Suele danke ich für das familiäre Klima und die lustigen Stunden die ich mit ihnen verbracht habe.

Harald Merkel und Ulrich Müller bin ich dankbar für die Hilfe beim experimentellen und apparativen Teil meiner Dissertation.

Ein besonderer Dank geht an die MAMI-Gruppe und die Werkstätten im Institut, ohne die die Durchführung dieser Arbeit nicht möglich gewesen wäre.

Großer Dank gebührt meinen Eltern für ihre Unterstützung bei meinem Studium und ganz speziell meiner lieben Michaela für die Motivationsschübe, die vorliegende Arbeit endlich aufs Papier zu bringen.

Lebenslauf

Name: Markus Weis

Anschrift: Lessingstraße 2a
33428 Harsewinkel

Geburtsdatum: 29.03.1970
Geburtsort: Mainz-Mombach

Eltern: Franz-Josef Weis
Maria Elisabeth Weis, geb. Kurz

Schulbildung:
1976-1980: Grund- und Hauptschule am Lemmchen
Mainz-Mombach
1980-1990 Willigis-Gymnasium Mainz
25.05.1990 Abitur

Wehrdienst:
01.06.1990-30.09.1991 Kassel/Fritzlar

Studiengang:
1991-1997 Studium der Physik an der Johannes-
Gutenberg Universität Mainz
07.04.1997 Diplom in Physik

Promotion:
seit 04.1997 Wissenschaftlicher Mitarbeiter am Institut für
Kernphysik der Universität Mainz