



**Universidad Autónoma
de San Luis Potosí**
FACULTAD DE CIENCIAS



TESIS PROFESIONAL

Para obtener el grado de:
LICENCIADA EN FÍSICA

PRESENTA:

Rocío Reyes Ramos

Estudio de interacciones hadrónicas con $\Lambda^0\Lambda^0$ y $\Lambda^0 p\pi^-$ en el estado final

ASESOR DE TESIS

Dr. Jürgen Engelfried

Profesor Investigador del Instituto de Física

SAN LUIS POTOSÍ, S.L.P.

Enero, 2015

A mi papá por su trabajo.
A mi mamá por su fortaleza.
A mi hermana por sus porras.
Muchas gracias.

Agradecimientos

A mi asesor el **Dr. Jürgen Engelfried** por su paciencia al momento de la realización de este trabajo y al explicar una y otra vez las dudas.

A la **colaboración SELEX** que me permitió usar los datos, sin los cuales este trabajo no hubiera sido posible. Al **Instituto de Física** por permitirme realizar este trabajo de tesis en sus instalaciones. Al **Dr. Antonio Morelos Pineda** por el préstamo de las computadoras en las que se realizó el trabajo de tesis.

A mi papá, **Toño**, por sus cuidados, sus consejos y su apoyo incondicional, agradezco su trabajo duro que bien sé que es para que nosotras tres estemos bien. A mi mamá, **Pati**, que es mi fortaleza y no deja que me doble, por sus cuidados y cariño. A mi hermana, **Erandi**, por sus ánimos, porras y sus regaños. Sé que no les facilité esta etapa así que muchas gracias por su comprensión y paciencia, los quiero.

A mis abuelos, **Cato** y **Carmelú**, a mis tíos, **Came**, **René**, **Cuti**, **Lilia**, **Silvia**, **Rubén**, **Martha**, **Marco**, **Maye**, **Aris**, **Laura** y **Gerardo**, por cuidarme en todo momento, siempre están presentes cuando los necesito. A mis **primos y primas**, por las risas y las buenas atenciones que siempre me brindan.

A los técnicos académicos **L.E.S.D. José Limón Castillo** y a **I.E. Luz del Carmen Nuche**¹ por su apoyo en el laboratorio de altas energías en todo cuanto a problemas con software, hardware, miles de dudas y por su amistad en los ataques de pánico que sucedieron. A la señora **Araceli Becerra** y a **Rosa María Rodríguez**, por todas las facilidades durante mis estudios de licenciatura y durante mi estancia en el instituto.

A mis **compañeros de laboratorio** por compartir el mismo espacio, pláticas y reuniones en la duración de este trabajo. A **Nora** por sus traducciones e interpretaciones del idioma y a sus explicaciones.

A mis **profesores de la facultad y del instituto** por sus enseñanzas, que permitieron mi formación académica. En especial a los doctores **Alvarado**, **Cisneros**, **Palomares**, **Mariana**, **Faustino**, **Guirado**, **Villaseñor**, **José Manuel García** a quienes agradezco su forma respetuosa de enseñar a los alumnos y del apoyo moral que recibí de ellos.

¹como noche pero con u

A mis amigas **Ericka** y **Katia** por escucharme, por sus consejos acompañados por cafecito para hacerme más ligera esta etapa y por muchas otras cosas. A **Thalía**, por sus mensajes con chistes que nadie más entiende, pero que relajan en un día de estrés.

Y a la **duración de esta tesis** que me permitió conocer mucha gente, como **Loki**, por algo suceden las cosas, quiero creer.

RESUMEN

Se estudian interacciones entre partículas hadrónicas. Se utilizó la información obtenida del experimento de blanco fijo SELEX, que se localizó en Fermilab, el cual recopiló datos en los años 1996 y 1997. Era un espectrómetro multi-etapas con una alta aceptación ($x_f > 0.1$). Se diseñó para el estudio sistemático con alta estadística de los mecanismos de producción de materia hadrónica. Utilizó un haz de Σ^- y π^- de 600 GeV/c y un haz de protones de 540 GeV/c, los cuales interactuaron con blancos de cobre y carbón. Se encontró una acumulación en la masa invariante de $\Lambda^0\Lambda^0$ en 2340 MeV/c² con un ancho de 14 MeV; y para $\Lambda^0 p\pi^-$ una acumulación en 2215 MeV/c² con un ancho de 1.9 MeV. Tales acumulaciones indican la primera observación de una partícula nueva y su estado excitado.

ABSTRACT

Interactions between hadronic particles are studied. Data collected by the fixed target experiment SELEX at Fermilab, which took data in 1996-97, were used. It was a multi-stage spectrometer, with high forward acceptance ($x_f > 0.1$). It was designed for high statistic systematic studies of the production mechanisms of hadrons. It used a Σ^- y π^- beam of 600 GeV/c and a proton beam of 540 GeV/c, interacting with copper and carbon targets. We found an accumulation in the invariant mass of $\Lambda^0\Lambda^0$ at 2340 MeV/c² with a width of 14 MeV; and in $\Lambda^0 p\pi^-$ a 2215 MeV/c² accumulation with a width of 1.9 MeV. Those accumulations indicate the first observation of a new particle and its excited state.

Índice general

Motivación	1
Introducción	5
1 Experimento SELEX	7
1.1 Descripción de SELEX	9
1.2 Espectrómetro del haz	9
1.2.1 Haz de Hiperones	10
1.2.2 Detector de radiación de transición del haz (BTRD)	12
1.2.3 Detectores de la trayectoria del haz (BSSD)	12
1.3 Espectrómetro del vértice	13
1.3.1 Blancos de producción	14
1.3.2 Detector de tiras de silicio del vértice (VSSD)	14
1.4 Espectrómetro M1	15
1.4.1 Cámaras proporcionales de alambre (PWC)	15
1.4.2 Cámara de arrastre (DC)	16
1.4.3 Detectores de silicio de ángulo grande (LASD)	16
1.5 Espectrómetro M2	16
1.5.1 Cámaras proporcionales multialambre (MPWC)	17
1.5.2 Hodoscopios	17
1.5.3 Detectores de Radiación de Transición de los Electrones (ETRD)	18
1.5.4 Ring Imaging Cherenkov Counter (RICH)	18
1.5.5 Cámaras de arrastre vectorial (VDC)	19
1.6 Espectrómetro M3	20
1.7 Imanes	20
1.8 Sistema disparador (Trigger)	21
1.9 Filtro en línea	22
1.10 Datos del experimento SELEX	22
1.11 Procesamiento de datos	24
1.11.1 SOAP	24
1.11.2 Pass	28
1.11.3 Strip	28
1.12 Simulación	29
1.13 Physics Analysis Workstation (PAW)	30
2 Análisis	31

2.1	Partícula Λ^0	33
2.2	Decaimiento en $\Lambda^0\Lambda^0$	36
2.3	Decaimiento en $\Lambda^0 p\pi^-$	41
3	Estudios Sistemáticos	53
3.1	Comprobación mediante simulación	53
3.2	Estabilidad del pico contra diferentes cortes de L/σ	55
3.3	Revisión de eventos repetidos	58
3.4	Estudio del ruido con event-mixing	58
4	Conclusiones	67
A	Colaboración SELEX	69
B	Programa <i>recon_recon.F</i>	73
C	Programa <i>anal.F</i>	89
D	Programa <i>sc.kumac</i>	99
E	Programa <i>mix.kumac</i>	107
	Bibliografía	119

Índice de figuras

1.1	Esquema del experimento SELEX	8
1.2	Esquema de los espectrómetros de SELEX	10
1.3	Sistema de coordenadas de SELEX	11
1.4	Esquema del espectrómetro del Haz.	11
1.5	Estaciones del detector de la trayectoria del haz.	13
1.6	Región del vértice.	13
1.7	Esquema del espectrómetro M1.	15
1.8	Esquema del espectrómetro M2.	17
1.9	Cámaras de alambre en M2.	18
1.10	Detector RICH.	19
1.11	Espectrometro M3.	20
1.12	Elementos que participan en el trigger.	21
1.13	Toma de datos.	23
1.14	Tabla <i>recdf.ocs</i>	27
2.1	Esquema de la reconstrucción de una Λ^0	34
2.2	Masa invariante de p y π^-	35
2.3	Distribución point back para $\Lambda^0\Lambda^0$	37
2.4	Masa invariante de dos partículas Λ^0 en el mismo evento	38
2.5	Ajuste de una gaussiana para dos partículas Λ^0	39
2.6	Masa invariante de p y π^- contra la masa invariante de p y π^-	40
2.7	Masa invariante de $p\pi^-$, sin el paquete de datos pp	41
2.8	Partícula H^0	42
2.9	Cortes en L/σ para $p\pi^-$ de 1.8 a 2.2	43
2.10	Cortes en L/σ para $p\pi^-$ de 1.8 a 2.2	44
2.11	Corte en L/σ para $p\pi^-$ de 1.9	45
2.12	Ajuste de una gaussiana para la masa invariante de $\Lambda^0 p\pi^-$	46
2.13	Masa invariante de p y π^- (Λ^0) y $p\pi^-$	48
2.14	Distribucion de χ^2 para el vértice secundario	49
2.15	Cortes en L/σ para $\Lambda^0 p\pi^-$ de 2.7 a 3	50
2.16	Masa invariante de dos partículas Λ^0 en el mismo evento	51
2.17	Corte en L/σ para $p\pi^-$ de 1.9	51
3.1	Masa invariante del decaimiento $H^0 \rightarrow \Lambda^0 p\pi^-$ encontrada mediante simulación	54
3.2	Ajuste de una función gaussiana para cortes de L/σ de 1.8 a 2.2	55
3.3	Ajuste de una función gaussiana para cortes de L/σ de 2.3 a 2.9	56
3.4	Gráficas de los parámetros del ajuste de una Gaussiana conforme cambia L/σ	57
3.5	Eventmixing para las entradas con rango de masa desde 2.19 a 2.24 GeV/c ²	61
3.6	Eventmixing para SUM con rango de masa desde 2.19 a 2.24 GeV/c ²	62
3.7	Eventmixing para las entradas con rango de masa desde 2.19 a 2.29 GeV/c ²	63

3.8	Eventmixing para SUM con rango de masa desde 2.19 a 2.29 GeV/c ²	64
3.9	Eventmixing para las entradas con rango de masa desde 2.19 a 4 GeV/c ² . .	65
3.10	Eventmixing para SUM con rango de masa desde 2.19 a 4 GeV/c ²	66

Índice de cuadros

1.1	Tabla de los blancos.	14
1.2	Información de los tres imanes.	20
1.3	Datos tomados de SELEX utilizados en el análisis de datos.	23

Motivación

Para la física de partículas, los años 60 del siglo XX fueron confusos ya que se conocían cientos de partículas, incluidos varios hadrones.

En 1964 los físicos Gell-Man y Zweig, cada uno por separado, introdujeron un modelo en el que los hadrones no eran partículas elementales, más bien eran partículas que se formaban por combinación de quarks y antiquarks. Su modelo presentaba tres *sabores*, o tipos, de quarks: *up* (u), *down* (d) y *strange* (s). Desde entonces los quarks sólo se han observado en estado ligado con otros quarks, lo que permite la formación de hadrones, los cuales se clasifican en:

1. Mesones: formados por estados ligados de quark-antiquark
2. Bariones: formados por estados ligados de tres quarks.

Una manera formal de llamarle a este modelo es $SU(3)$ *flavor symmetry* ($SU(3)_f$). Por ejemplo: para los mesones formados con pares de los tres quarks (u, d, s) y tres antiquarks ($\bar{u}, \bar{d}, \bar{s}$) conocidos hasta ese momento, se obtienen nueve combinaciones, que pueden ser nueve combinaciones con spin 0 y nueve combinaciones con spin 1; estas combinaciones se agrupan en un octete y un singlete. Esto coincide con los mesones que se han observado. Sin embargo, debido a la diferencia de masa entre los quarks, la simetría $SU(3)_f$ no es exacta.

Por esos años los quarks sólo eran modelos matemáticos utilizados para describir cosas, aún con la observación de los partones en la difracción profunda, es hasta 1974 cuando se detectó al quark charm (c) que todo cambió. Al incluir este nuevo quark, $SU(3)_f$ cambia a $SU(4)_f$, siendo de igual manera una simetría inexacta y sólo se utiliza para clasificar las partículas compuestas por la combinación de los cuatro quarks conocidos (u, d, s y c).

²*flavor*: sabor

Después por los años 70 del siglo pasado surgieron dos problemas, primero era necesario explicar porqué los quarks se mantenían unidos dentro de los hadrones y el otro problema que surgió fue el buscar un nuevo número cuántico.

Para solucionar el primer problema surgió la fuerza fuerte la cual es la responsable de mantener unidos a los quarks dentro de los hadrones y es transmitida entre dos quarks por partículas sin masa llamadas gluones. Esta fuerza es parte de otras tres interacciones fundamentales que son la fuerza débil, la fuerza electromagnética y la gravitacional.

La necesidad de otro número cuántico surgió con partículas como Ω^- ó Δ^{++} las cuales están formadas por tres quarks s y u , respectivamente y tienen spines paralelos. Como los quarks son fermiones esta combinación viola el principio de exclusión de Pauli. Para evitar esto, se agrega la propiedad llamada carga o *color*. El número cuántico *color* puede asumir tres valores diferentes: *rojo*, *azul* y *verde*³. Con esto podemos diferenciar a las tres s o a las tres u , que aunque sean del mismo sabor su color será diferente. De esta manera se cumple el principio de Pauli.

Ahora tenemos tres quarks u (u_r, u_g, u_b), tres quarks s , etcétera, es decir los quarks pueden ser de cualquiera de los tres colores así como los antiquarks de los tres anticolores. Los gluones llevan color y anticolor simultáneamente. Para los gluones se tienen nueve combinaciones que forman dos estados: un octete y un singlete. Donde el singlete se construye simétricamente con los tres colores y los tres anticolores dando como resultado un blanco, del cual su carga neta de color es igual a cero.

Por ejemplo, para el barión Δ^{++} sus tres quarks u siempre están cambiando su color mediante emisión y absorción de gluones para ser distintos entre sí y obtener una carga neta de color igual a cero, con lo que se obtiene un singlete. Otra forma del singlete es el color y anticolor que forman un mesón. Con esto se dice que todos los hadrones poseen una carga de color neta igual a cero, siendo singletes de color.

Todo lo anterior se resume con la cromodinámica cuántica (QCD⁴), la cual afirma que las combinaciones quark-antiquark, tres quarks o tres antiquarks, adentro de un hadrón dan como resultado un singlete de color del grupo $SU(3)_c$. La fuerza fuerte, los gluones y la carga de color ayudan a la explicación de porque sólo se han observado quarks en estados ligados y no libres.

³*Red, blue, green (r, b, g)*

⁴*Quantum Chromodynamics*

En la actualidad se conocen seis sabores de quarks: *up* (u), *down* (d), *strange* (s), *charm* (c), *bottom* (b) y *top* (t) además de seis antiquarks, pero sólo se han observado mesones y bariones, es decir, no se han encontrado partículas con estados ligados de más de tres quarks.

Sin embargo existen hipótesis para producir hadrones formados por más de tres quarks, ya que se pueden formar singletes de color con más de tres quarks. Un ejemplo son los pentaquarks formados por cuatro quarks y un antiquark[4] ó las partículas formadas por seis quarks, como el dibarión H^0 que predijo Jaffe en 1977[5]; las cuales no se han observado experimentalmente.

De ser posible encontrar H^0 se estaría demostrando que los quarks pueden juntarse en grupos de más de tres quarks, y quedaría como trabajo a futuro saber porque no se han encontrado partículas formadas por cuatro o cinco quarks.

En 1996 surgió el experimento SELEX⁵ en Fermilab, el cual era un experimento de blanco fijo que investigó, entre otras cosas, la producciones de partículas hadrónicas, como las partículas Λ^0 junto con sus modos de decaimiento. Entre sus principales logros están: la primera observación del barión *double charmed* Ξ_{cc}^+ [1][2] y la primera observación del mesón angosto *charm-strange* D_{sJ}^+ [3].

Para este trabajo es necesario producir hiperones⁶ y el experimento SELEX ofrece esta ventaja al contar con un haz (Σ^-), el cual favorece a la formación de partículas con strangeness o “extrañas” (Lambda (Λ), Sigma (Σ), Xi (Ξ), Omega (Ω) y tal vez la partícula H^0), ya que las partículas “extrañas” tienen un tiempo de vida $\sim 10^{-11}$ s y es necesario producirlas en ese momento.

El presente trabajo de tesis tiene como objetivo buscar el dibarión H^0 , el cual no ha sido observado hasta ahora, para esto:

1. Se utilizan los datos que provienen del experimento SELEX para estudiar las interacciones hadrónicas con $\Lambda^0\Lambda^0$ y $\Lambda^0 p\pi^-$ en el estado final.
2. Se comparan estas interacciones con lo publicado por Jaffe.

⁵*Segmented Large x_F Baryon Spectrometer*

⁶bariones con uno o más quarks s

Introduccion

Los quarks forman partículas llamadas hadrones y éstas se dividen en dos: bariones, formados por tres quarks, y mesones, formados de un par quark-antiquark.

Sin embargo han surgido hipótesis sobre hadrones formados por estados ligados de más de tres quarks. Un ejemplo es el dibarión H^0 , que predijo Jaffe utilizando el modelo de hadrones como bolsa en 1977. El dibarión H^0 es una partícula formada de seis quarks $uuddss$, con ⁷ $J^P=0^+$ y con una masa de 2150 MeV/c², además de un estado excitado con $J^P=1^+$ y una masa de 2335 MeV/c² [5].

El modelo de Jaffe menciona que el singlete de color para seis quarks se encuentra con una masa de 2150 MeV/c², y los que componen al octete se encuentran con masa de 2335 MeV/c², 2395-2465, 2220-2230 y 2480-2505 MeV/c².

Jaffe en su artículo menciona los siguientes rangos de masa para el estado base:

1. $M(H^0) < 2055 \text{ MeV/c}^2$
2. $2055 \text{ MeV/c}^2 < M(H^0) < 2230 \text{ MeV/c}^2$
3. $2230 \text{ MeV/c}^2 < M(H^0) < 2380 \text{ MeV/c}^2$
4. $M(H^0) > 2380 \text{ MeV/c}^2$

El autor descartó los casos 1 y 4, el primero por no ser favorecido por sus cálculos y el cuarto por estar encima del umbral del acoplamiento de dos bariones. Se centró en el segundo y tercer caso, que se resume como sigue:

⁷J: momento angular total, P: paridad

- Si la masa $H^0 > 2230 \text{ MeV}/c^2$, la partícula H^0 tendrá un decaimiento fuerte en dos partículas Λ^0 ($H^0 \rightarrow \Lambda^0 \Lambda^0$). Esto significa que los seis quarks $uuddss$ se rearreglan de la forma $uds - uds$, y decaerá en dos Λ^0 (uds) ya que le sobra energía y la fuerza fuerte es la única que interviene. Al ser decaimiento donde interviene la fuerza fuerte se estima un tiempo de vida de $10^{-23}\text{s} \sim 10^{-24}\text{s}$.
- Si la masa $H^0 < 2230 \text{ MeV}/c^2$, entonces H^0 no puede decaer fuerte, ya que la partícula Λ^0 es el barión más ligero con un quark s , es decir no va a decaer sólo por el arreglo de quarks. Por lo tanto actúa la fuerza débil, que es capaz de cambiar quarks, cambiando al quark s para que la partícula pueda decaer. El quark s cambiará a un quark u y en una partícula⁸ W^- , el cual a su vez decae en dos quarks: d y $\bar{u}$. Lo que nos deja con los quarks $uudd\bar{u}s$, estos se reacomodan para obtener otras partículas como: una Λ^0 (uds), un protón (uud) y un π^- ($d\bar{u}$). Lo que nos deja el siguiente decaimiento débil: $H^0 \rightarrow \Lambda^0 p \pi^-$.

Al ser un decaimiento fuerte, en el caso donde $M(H^0) > 2230 \text{ MeV}/c^2$, su tiempo de vida es corto (10^{-24}s), pero al ser la masa sólo un poco mayor a $2230 \text{ MeV}/c^2$, se espera que su tiempo de vida sea más largo (10^{-23}s), formando un “pico” menos ancho en la distribución de masa invariante y por lo tanto sea posible observarlo.

Para el decaimiento débil: $H^0 \rightarrow \Lambda^0 p \pi^-$, esta combinación de partículas, Λ^0 , un protón y π^- , es la más ligera y por lo tanto es la que se eligió para este trabajo. Por lo tanto se espera un tiempo de vida de $\sim 10^{-10}\text{s}$, similar al tiempo de vida de los hiperones.

Aún cuando el modelo de Jaffe no ha podido ser demostrado experimentalmente, el caso teórico de la partícula H^0 continúa vigente. Se observaron algunas $\Lambda^0 \Lambda^0$ virtuales, relacionadas con el dibarión H^0 registradas en el laboratorio KEK [6], pero otras búsquedas produjeron resultados negativos [7],[8],[9] [10].

En este trabajo de tesis se utilizó la información obtenida por el experimento SELEX para buscar los productos de los decaimientos ya descritos y se espera observar por primera vez a la partícula H^0 , en cualquiera de los dos decaimientos planteados:

- $\Lambda^0 \Lambda^0$
- $\Lambda^0 p \pi^-$

⁸ $W^\pm$ y Z^0 : son las partículas encargadas de mediar la fuerza débil

Capítulo 1

Experimento SELEX

El experimento SELEX (*Segmented Large x_F Baryon Spectrometer*, también conocido como E781), se localizó en el Fermilab¹, con él se recopilaron datos en los años 1996 y 1997. Fué un experimento de blanco fijo diseñado para realizar estudios con alta estadística de los mecanismos de producción y decaimiento de los bariones *Charm* [22]. Del experimento SELEX, (Figura 1.1), se obtuvieron los datos para la realización de este trabajo de tesis.

El experimento funcionó de manera general de la siguiente forma: un haz de protones proveniente del Tevatrón (acelerador circular), era dirigido hacia el experimento SELEX hacia el blanco de producción de hiperones para tener un haz de hiperones y después éste colisionaba con los blancos (de carbón y cobre). Las partículas que se producían de esta colisión viajaban en la dirección del haz hasta llegar al final del aparato. SELEX contó con espectrómetros en varias etapas, cada uno cumplía con una función diferente. Los imanes entre cada espectrómetro deflectaban a las partículas cargadas para obtener los momentos. La posición de las partículas se obtuvo de la combinación de la información obtenida de las cámaras de alambre y de los detectores de silicio. Y para la identificación de las partículas se utilizó el detector RICH². El experimento también contó con un filtro en línea.

¹Fermi National Accelerator Laboratory

²Ring Imaging Cherenkov Counter

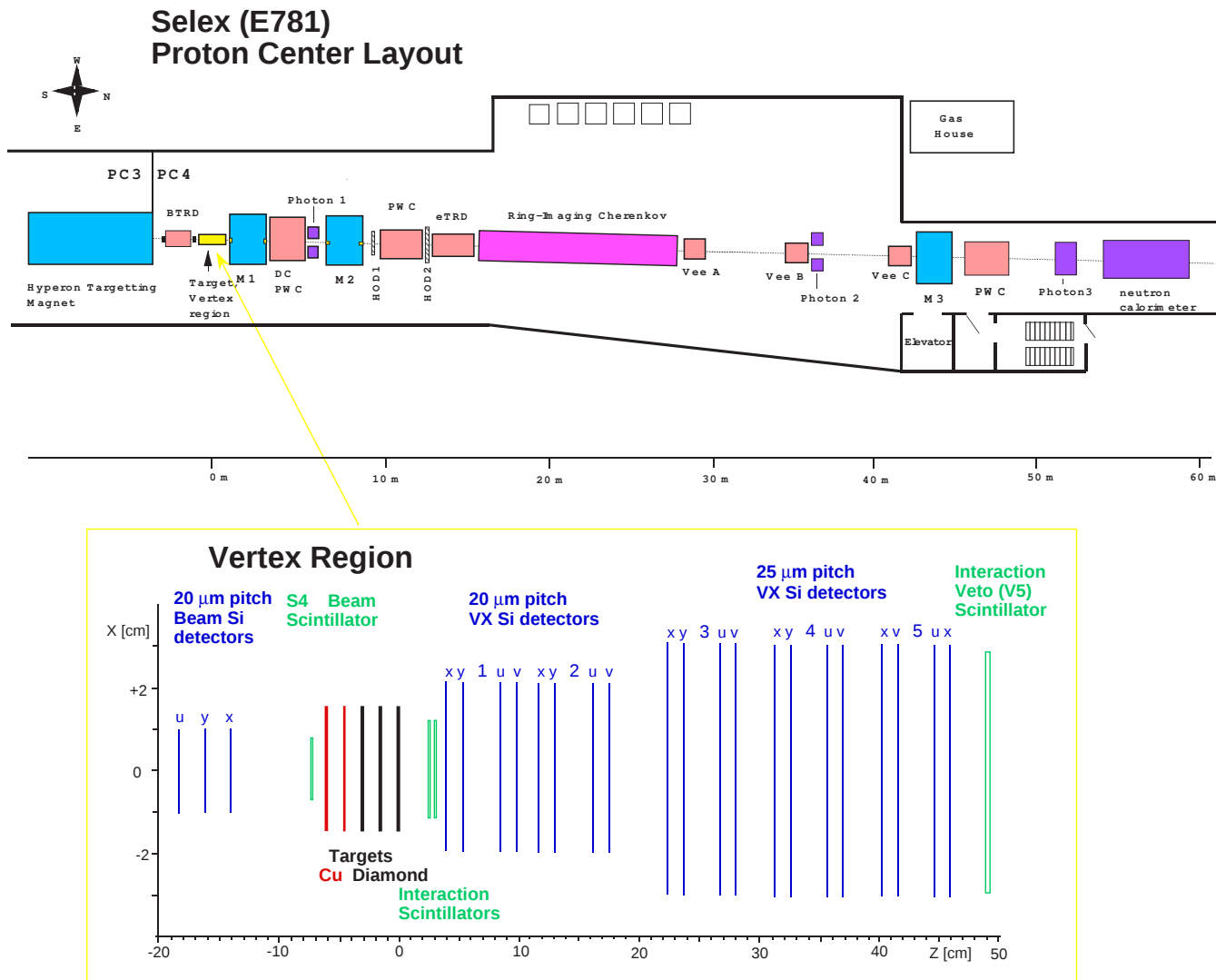


Figura 1.1: Esquema del experimento SELEX

1.1. Descripción de SELEX

El espectrómetro SELEX tuvo una longitud de 60 m, y en los hechos estuvo formado por cinco espectrómetros:

- El espectrómetro del haz
- El espectrómetro de vértices
- El espectrómetro M1
- El espectrómetro M2
- El espectrómetro M3

Cada espectrómetro, con excepción del espectrómetro de vértices, incluía detectores de partículas y un imán, con lo que fue posible calcular el momento de las partículas cargadas. La parte del vértice carecía de un campo magnético por que su diseño fue para una alta resolución de rastreo cerca del blanco de interacción (Figura 1.2).

Para el sistema de coordenadas el eje de las z se seleccionó a lo largo de la dirección del haz, el eje de las y verticalmente hacia arriba, y el eje de las x completaba el sistema de coordenadas con el uso de la regla de la mano derecha. El origen del sistema de coordenadas se encontraba al final del último blanco (Figura 1.3).

A su vez, cada espectrómetro incluyó su propio sistema de coordenadas; estos sistemas estaban alineados en posición y ángulos con el sistema de coordenadas global. Sólo el origen sobre el eje de las z cambió; este nuevo origen se encontró en el centro magnético del imán de cada espectrómetro.

1.2. Espectrómetro del haz

El espectrómetro del haz se conformó de todo lo que se encontraba en la dirección del haz hasta los blancos de producción (Figura: 1.4).

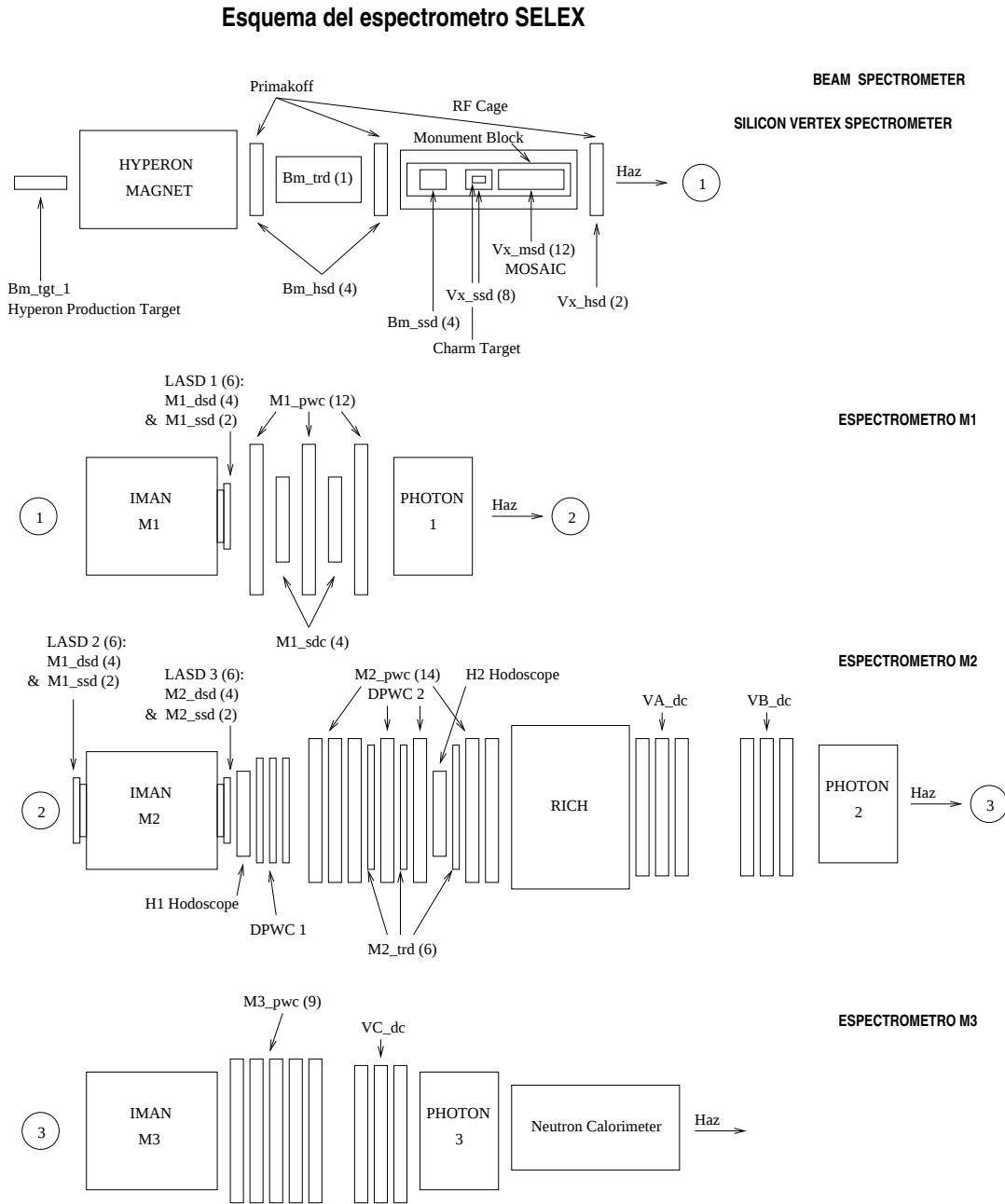


Figura 1.2: Esquema de los espectrómetros de SELEX

1.2.1. Haz de Hiperones

Del acelerador Tevatrón salía un haz de protones de 800 GeV/c, el cual era dirigido hacia un blanco de Berilio, que se encontraba a la entrada del canal de hiperones. Las partículas

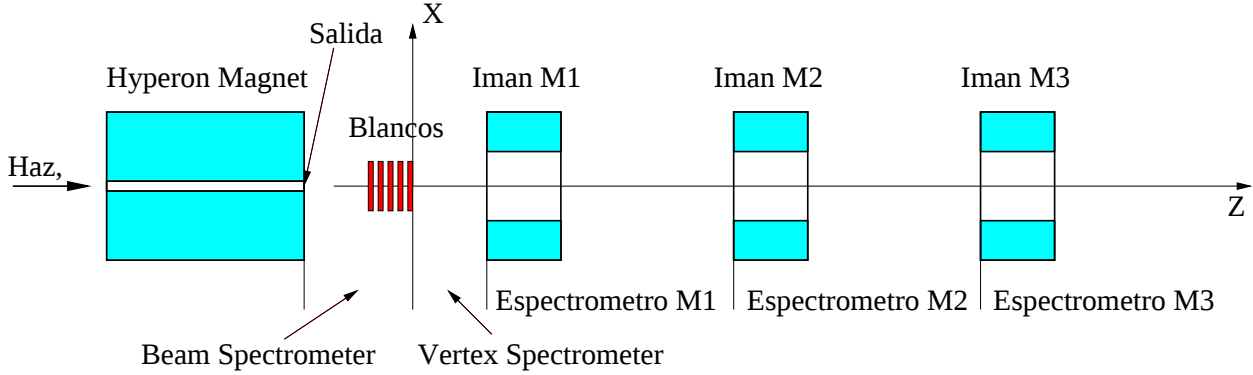


Figura 1.3: Sistema de coordenadas de SELEX

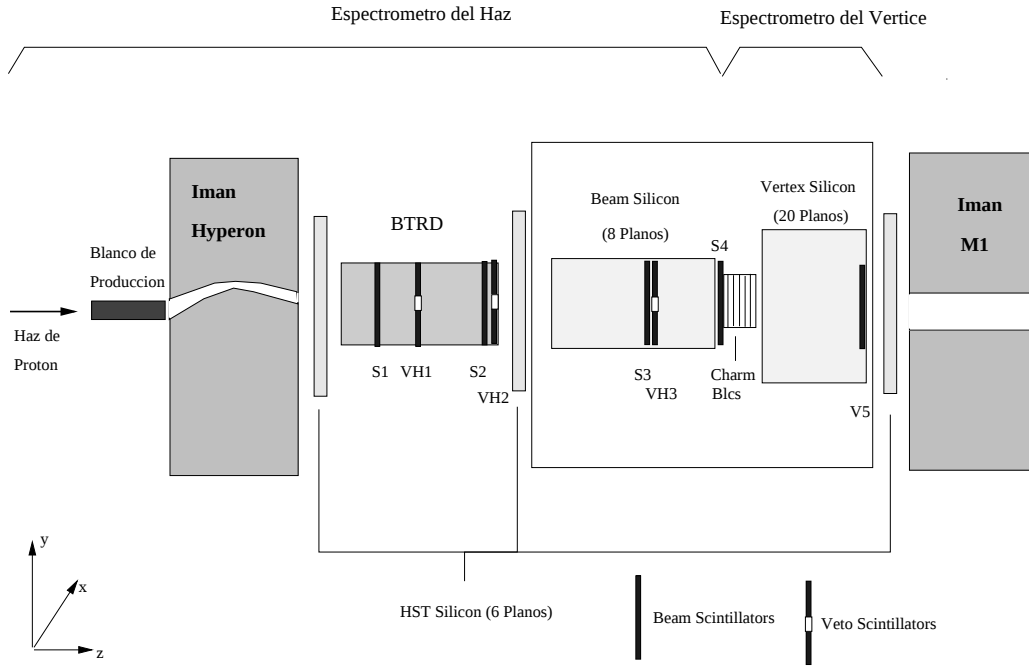


Figura 1.4: Esquema del espectrómetro del Haz.

que se producían conformaban un haz secundario que entraba al canal de hiperones. Las paredes del canal de hiperones eran de tungsteno; este elemento absorbió las partículas que no interactuaron con el blanco de Berilio y sólo partículas con altos momentos, 600 ± 50 GeV/c, pasaban a través del imán. Se necesitó de momentos altos para que la fracción relativa de hiperones en el haz secundario creciera junto con el momento. Este canal estrecho de 7.3 m de largo estaba ubicado adentro de un campo magnético de 3.5 Teslas, producido por el imán de hiperones.[11] La composición esperada del haz de acuerdo a un estudio que se realizó en el experimento de E761 con una energía de 600 GeV/c es: Para el haz negativo:

$\approx 53\%$ de π^- , $\approx 1.6\%$ de K^- , $\approx 43\%$ de Σ^- , $\approx 1.6\%$ de Ξ^- , $\approx 0.04\%$ de Ω^- , $\approx 0.04\%$ de $\bar{p}$. Para el haz positivo: $\approx 2.8\%$ de π^+ , $\approx 0.8\%$ de K^+ , $\approx 2.8\%$ de Σ^+ , $\approx 94\%$ de p . [12][13]. El haz de hiperones de SELEX ofreció la posibilidad única de medir la sección eficaz para protones, π^- y Σ^- en un ambiente poco contaminado.[14]

1.2.2. Detector de radiación de transición del haz (BTRD)

El detector de radiación de transición (BTRD³) detectó una forma de radiación electromagnética llamada radiación de transición. Esta radiación se produce en el momento en que las partículas cargadas cruzan la frontera entre dos medios con diferentes propiedades dieléctricas. La probabilidad de radiación es proporcional al factor γ de Lorentz de la partícula. Por lo que el mesón, π^- , produjo más señales en el BTRD que un barión, Σ^- ; ambas partículas tienen el mismo momento y se diferencian por su masa: π^- tienen menos masa que Σ^- . Las partículas que conformaron el haz de hiperones fueron etiquetadas en los 10 módulos idénticos del detector. Cada módulo contuvo un radiador de 200 láminas de polipropileno de $17\ \mu\text{m}$ de ancho y espaciadas a $500\ \mu\text{m}$, seguidas por tres cámaras proporcionales de alambre (PWC), llenados con una mezcla gaseosa de $\text{Xe} + 30\% \text{CH}_4$ que detectaban la radiación de transición. Este gas ayudó a maximizar la absorción de fotones en la radiación.[15] Estas cámaras de alambre daban una señal de salida cuando detectaban energía por encima de un umbral fijo. El número de planos activados en el BTRD es igual a la suma de las señales de salida de todas las cámaras de alambre. Con esto el número total de señales en el BTRD fueron utilizadas para separar el componente barión del componente mesón del haz.

1.2.3. Detectores de la trayectoria del haz (BSSD)

El detector de tiras de silicio del haz (BSSD⁴) se integró de ocho planos con tiras de silicio en un solo lado. Las tiras sirvieron para conocer la posición de la trayectoria del haz y la dirección del movimiento. Los planos estaban distribuidos en tres bloques de 3, 2 y 3 planos cada uno. Cada detector estaba formado por 1024 tiras de $20\ \mu\text{m}$ de separación entre ellas, y un área sensitiva de $4\ \text{cm}^2$. La posición de la trayectoria del haz en los blancos de producción fue medida con una resolución de $4\ \mu\text{m}$ aproximadamente. La señal de salida de las tiras fue extraída por medio de chips (Figura 1.5).

³Beam Transition Radiation Detector

⁴Beam Silicon Strip Detector

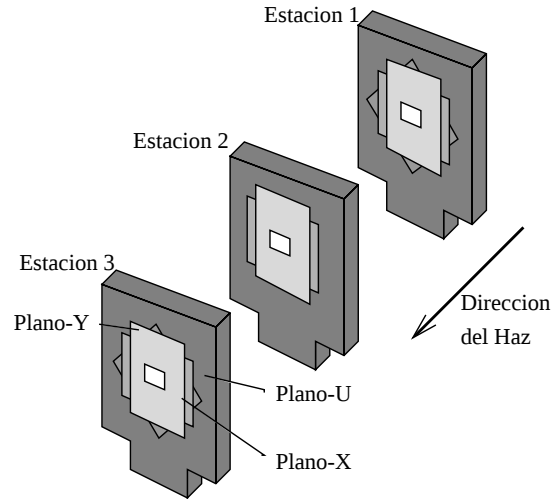


Figura 1.5: Estaciones del detector de la trayectoria del haz.

1.3. Espectrómetro del vértice

Este espectrómetro (Figura 1.6), se encontraba justo al terminar el espectrómetro del haz y terminaba a la mitad del imán M1. El espectrómetro estaba formado por dos partes principales: los blancos de producción y el detector de tiras de silicio del vértice.

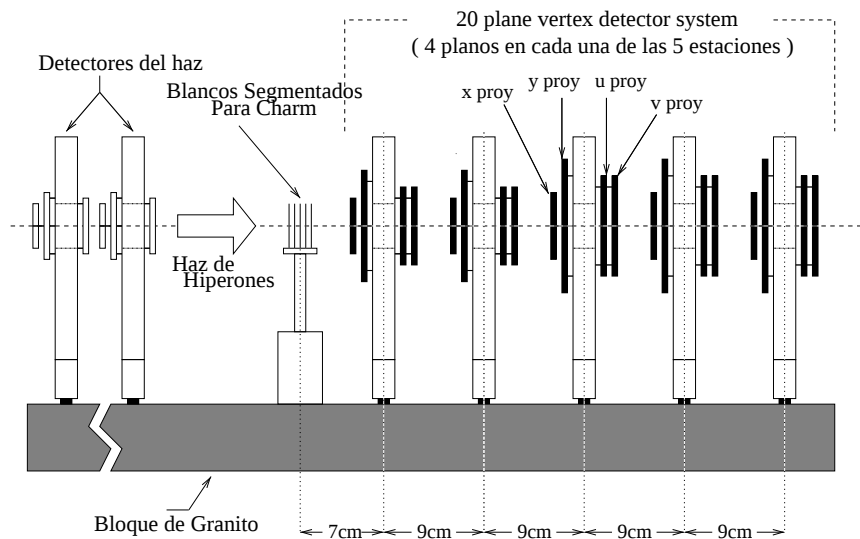


Figura 1.6: Región del vértice.

Blanco	Material	Anchura (mm)	Posición en el eje z	Número Atómico A	Densidad. g/cm ³	Rango de interac- ción. %
1	cobre	1.6	-6.13	63.5	8.96	1.06
2	cobre	1.016	-4.62	63.5	8.96	0.76
3	diamante	2.2	-3.10	12	3.20	0.82
4	diamante	2.2	-1.61	12	3.20	0.82
5	diamante	2.2	-0.11	12	3.20	0.82

Cuadro 1.1: Tabla de los blancos.

1.3.1. Blancos de producción

Las partículas del haz interactuaron con uno de los cinco blancos, con una longitud de interacción combinada de 4.3 % casi 1 % de cada blanco, divididos de la siguiente manera: dos de cobre (Cu) y tres de diamante (C), la separación entre ellos fué de 1.5 cm (Cuadro 1.1)[11]. Se eligió usar varios blancos del mismo material no muy anchos, en lugar de usar sólo uno muy ancho, para evitar que el segundo vértice se confunda del producto de la interacción con el material con el decaimiento de las partículas.

1.3.2. Detector de tiras de silicio del vértice (VSSD)

El detector VSSD⁵ consistió de 20 planos montados en cinco estaciones; estos detectaron trayectorias secundarias con alta resolución espacial. Los primeros ocho detectores de las dos primeras estaciones tenían una separación entre tiras de 20 μm , con un área efectiva de 5.1 x 5.0 cm². Los últimos 12 detectores (detectores mosaico) tenían una separación entre tiras de 25 μm y una región activa de 8.3 x 9.6 cm². Los detectores estaban montados en una base óptica especial de granito. Cada estación tiene cuatro planos x , y , u y v , los últimos dos rotados 45°. Cada detector tenía una eficiencia mayor a 98 % y una resolución espacial de 6.5 μm .

⁵Vertex Silicon Strip Detector

1.4. Espectrómetro M1

Estaba conformado por el imán M1, y los detectores que se encontraban entre los imanes M1 Y M2. Fué diseñado para analizar partículas con momentos de 2.5 a 1.5 GeV/c (Figura 1.7). Este espectrómetro contó con tres cámaras de alambre (PWC⁶), dos cámaras de arrastre (DC⁷) y dos detectores de silicio de ángulo grande (LASD⁸).

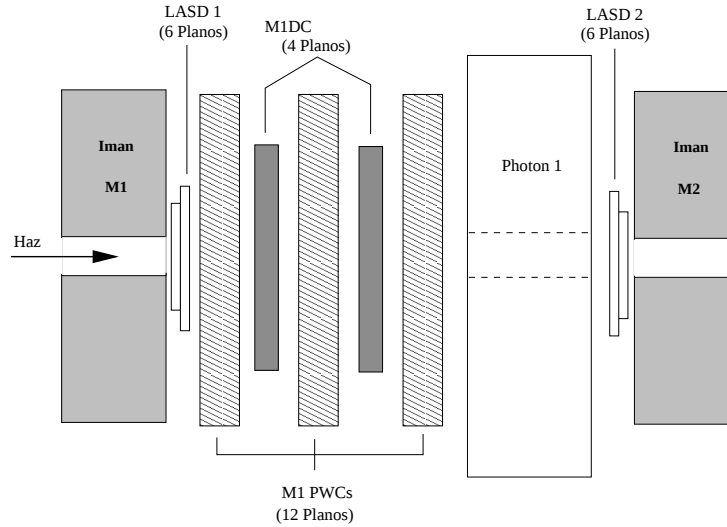


Figura 1.7: Esquema del espectrómetro M1.

1.4.1. Cámaras proporcionales de alambre (PWC)

Los alambres que formaban éstas cámaras tenían una separación de 3 mm entre ellos, que funcionaron como ánodos, y estaban centrados entre dos planos que actuaban como cátodos. Tenían una área efectiva de $2 \times 2 \text{ m}^2$. Las cámaras tenían una eficiencia de 90 % y una resolución espacial de 0.9 mm. Cada una era llenada con una mezcla de gas que se ionizaba cuando una partícula cargada pasaba por éste. Los iones cargados positivamente eran arrastrados por el campo eléctrico hacia el cátodo y los electrones hacia el ánodo. Cuando los electrones estaban cerca del ánodo producían un efecto “avalancha”, lo cual incrementaba la señal recolectada por el alambre. Esta señal era leída y así se determinaba la posición del alambre por la cual había pasado una partícula cargada. Contaban con cuatro planos sensitivos para las proyecciones x , y , u y v .

⁶Proportional Wire Chamber

⁷Drift Chambers

⁸Large Area Silicon Detectors

1.4.2. Cámara de arrastre (DC)

Para las cámaras de arrastre se mantuvo la geometría básica de las PWC, pero la distancia entre los alambres fué mayor (cm), y se utilizó el tiempo de arrastre de los electrones al alambre como información adicional. La velocidad de arrastre es constante en el gas bajo un campo eléctrico homogéneo y se mide en $\text{cm}/\mu\text{s}$ [17]. El espectrómetro M1 tenía dos DC y éstos tenían dos planos sensitivos. Medían las señales de salida en la proyección x con un 80 % de eficiencia y 0.7 mm de resolución. Estas cámaras utilizaban los valores de la velocidad de arrastre y el tiempo que tarda la partícula en atravesar, entonces se podía obtener un valor mas fino para la posición[18].

1.4.3. Detectores de silicio de ángulo grande (LASD)

Los LASD se utilizaron para obtener una alta resolución en el rastreo de partículas de altas energías en la región central del haz[14]. SELEX contaba con tres estaciones LASD: una se encontraba al final del imán M1 y las otras dos al principio y final del imán M2. Cada estación tenía dos detectores doble cara (DSD) con una distancia entre las tiras de $50 \mu\text{m}$ y un área sensitiva de $3.2 \times 2.6 \text{ cm}^2$, estos planos medían en las proyecciones x e y . También contaba con dos planos de una sola cara (SSD), con una distancia entre las tiras de $50 \mu\text{m}$ y un área sensitiva de $3.2 \times 3.2 \text{ cm}^2$, que medían las proyecciones u y v . La eficiencia para detectar los hits fue de 95 - 99 %. Cada LASD tenía una resolución de $14 \mu\text{m}$.

1.5. Espectrómetro M2

Estaba conformado por el iman M2 y todos los detectores que se encontraban entre los imanes M2 y M3. El iman M2 desviaba a las partículas con momentos menores a 15 GeV del vértice primario. Los detectores que forman a M2 son: un tercer módulo de detectores de silicio de ángulo grande (LASD3), siete cámaras proporcionales de multialambre (MPWC⁹), dos hodoscopios, un detector de luz Cherenkov (RICH¹⁰), seis planos de detectores de radiación de transición de electrones (ETRD¹¹), y dos estaciones de cámaras de arrastre vectoriales (VDC¹²) (Figura 1.8).

⁹Multiwire Proportional Chambers

¹⁰Ring Imaging Cherenkov Counter

¹¹Electron Transition Radiation Detector

¹²Vector Drift Chambers

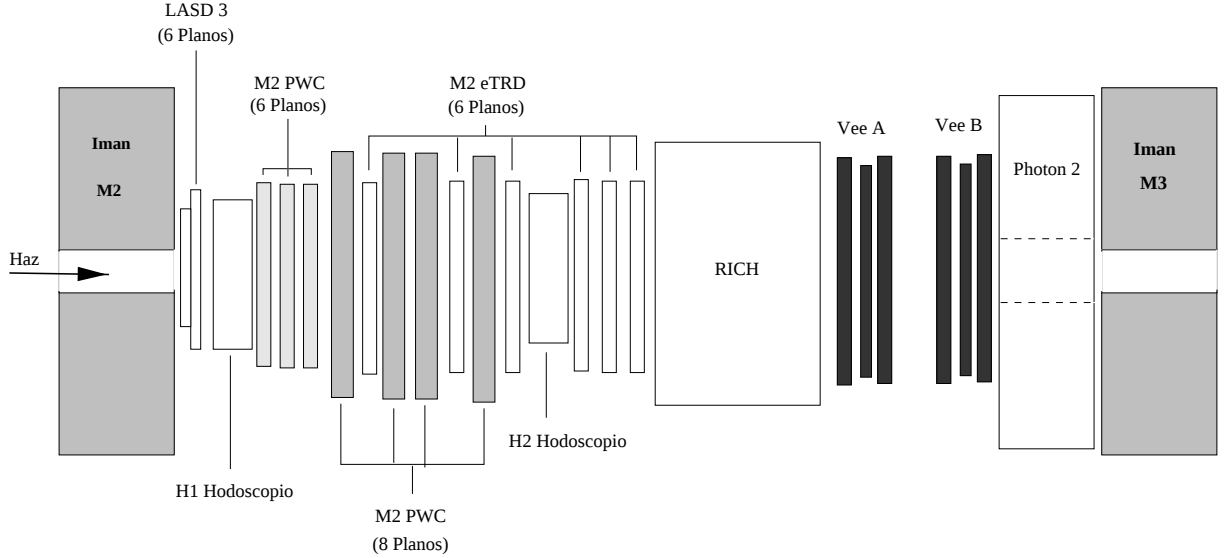


Figura 1.8: Esquema del espectrómetro M2.

1.5.1. Cámaras proporcionales multialambre (MPWC)

Las siete MPWC contaban con 2 mm de espacio entre alambres. Tres de las cámaras tenían una apertura de $60 \times 60 \text{ cm}^2$ y las otras cuatro tenían una apertura de $60 \times 100 \text{ cm}^2$. Cada una de las cámaras tenía dos planos sensitivos en proyecciones ortogonales. Tenían una eficiencia de detección del 95 % y una resolución espacial de 0.6 mm (Figura 1.9).

1.5.2. Hodoscopios

El objetivo de los dos hodoscopios H1 y H2 era el de eliminar eventos con varias trayectorias de manera rápida y así mandar una señal al hardware del mecanismo disparador[19]. Dichos hodoscopios usaron contadores de centelleo los cuales son usados para obtener el número de partículas y su momento.

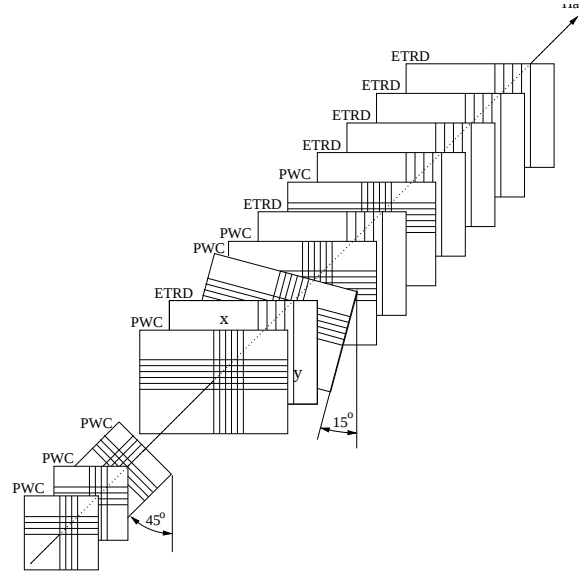


Figura 1.9: Cámaras de alambre en M2.

1.5.3. Detectores de Radiación de Transición de los Electrones (ETRD)

Los seis módulos de ETRD tenían como función separar a los electrones de los hadrones. Enfrente de cada módulo estaban localizadas 200 láminas de polipropileno de $17 \mu\text{m}$ de grueso las cuales generaban la radiación de transición. La radiación era detectada por cámaras de alambre (MPWC) con un área de $103 \times 63 \text{ cm}^2$ con espacio entre sus alambre de 4 mm y una mezcla de gas formado de xenón y metano.[20][21]

1.5.4. Ring Imaging Cherenkov Counter (RICH)

La radiación Cherenkov ocurre cuando una partícula cargada con una velocidad v , mayor que la velocidad de la luz en un medio con un índice de refracción (c/n) emite luz en un ángulo fijo dado por la ecuación 1.1.

$$\cos\theta_c = \frac{1}{\frac{v}{c}n} = \frac{1}{\beta n} \quad (1.1)$$

El detector RICH estaba formado por tres partes[22]:

- Vasija: medía 10 m de largo y 2.4 m de diámetro, la vasija se llenó con gas neón puro (a 1 atm), por el cual las partículas cruzaban y al viajar más rápido que la luz en ese medio emitían luz Cherenkov.
- Espejos: el sistema consistía de 16 espejos esféricos de forma hexagonal, con un radio de curvatura de 20 m. Los espejos reflejaban la luz Cherenkov hacia una matriz de 2848 tubos fotomultiplicadores.
- Detección de fotones: esta parte estaba formada por una placa de soporte, tubos fotomultiplicadores y el equipo para la señal de salida.

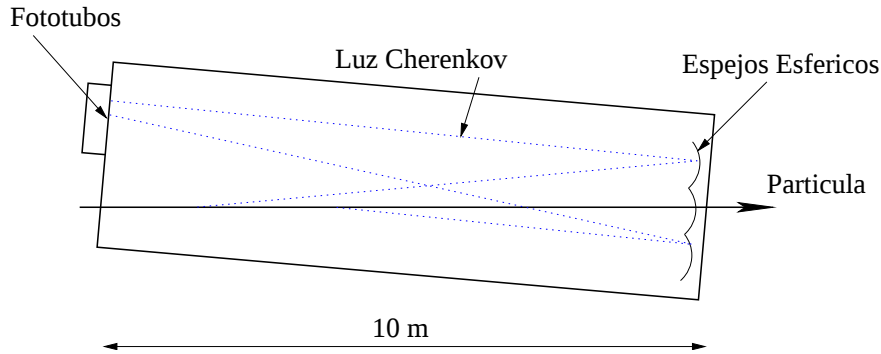


Figura 1.10: Detector RICH.

La luz Cherenkov, que se emitía cuando la partícula entraba a la vasija, era reflejada en los espejos al final del tanque y éstos reflejaban la luz hacia los fotomultiplicadores (Figura 1.10). La luz Cherenkov reflejada formaba anillos en la superficie de los fotomultiplicadores. El radio de los anillos formados crecían con la velocidad de la partícula. Una partícula con $\beta = 1$ generaba un anillo de radio 11.5 cm con 13.6 hits en el anillo, cada anillo fué medido con una resolución espacial de 5.5 mm [22][23]. La eficiencia del detector es bastante alta, sin embargo, disminuye a momentos grandes ya que el radio del anillo tiende a ser el mismo

1.5.5. Cámaras de arrastre vectorial (VDC)

El experimento contaba con nueve VDC las cuales estaban distribuidas en tres estaciones, dos de ellas en M2 (VeeA y VeeB) y la otra en M3 (VeeC). Las estaciones medían trayectorias en las proyecciones x, y, u ó x, y, v , tenían una apertura de $116 \times 116 \text{ cm}^2$ con una eficiencia de 90 % y una resolución de $100 \mu\text{m}$. Los planos midieron el vector de la trayectoria y no sólo la posición, cada cámara tenía ocho planos sensitivos en las celdas finas de la región central de la cámara y seis planos sensitivos en las celdas gruesas.[24]

1.6. Espectrómetro M3

El espectrómetro M3 estaba formado por el imán M3 y los detectores después de éste. Los detectores eran: tres MPWC (dos medían $64 \times 64 \text{ cm}^2$ y el otro $115 \times 89 \text{ cm}^2$), un VDC (VeeC), un tercer detector de fotones y el calorímetro de neutrones (Figura 1.11). El propósito de este espectrómetro era el de medir los productos de los decaimientos que tardaban en ocurrir.

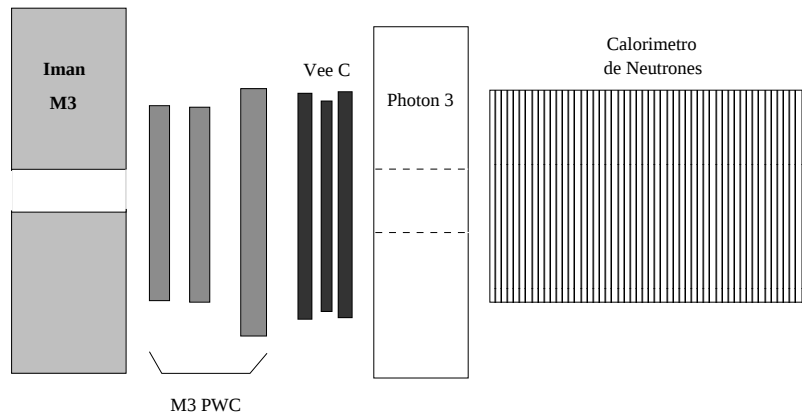


Figura 1.11: Espectrometro M3.

1.7. Imanes

Algunas características de los imanes M1, M2 y M3 estan resumidas en el Cuadro 1.2 [11].

Iman	z (center pos.) (cm)	B(T)	p_T -kick (GeV/c)	Longitud (cm)	Anchura (cm)
M1	190	1.35	0.7371	262.50	50.00
M2	745	1.54	0.8285	231.73	85.09
M3	4240	1.30	0.4170	231.73	85.09

Cuadro 1.2: Información de los tres imanes utilizados: posición, campo magnético, momento transversal kick, longitud y anchura.

Uno de los valores importantes es el p_T -kick, que es el valor del momento transversal adicional que se obtiene mientras que la partícula pasa por el iman. Se le denomina kick¹³ por el aparentemente cambio de dirección que presenta y que sucedió en algún punto al pasar por el iman y que no se puede ver por falta de detectores adentro del iman.

1.8. Sistema disparador (Trigger)

El sistema disparador se utilizó para señalar cuándo se producía una interacción preseleccionada de las partículas con los blancos que produjeran o decayeran con un quark Charm. Si el evento era aceptado, se registraba y pasaba al filtro en línea, y se rechazaban todos los demás eventos. Este mecanismo estaba formado por un set de contadores de centelleo (S1-S4), contadores veto (VH1,VH2), contadores de interacción (IC) y dos hodoscopios (H1,H2) (Figura 1.12). El sistema contaba con cuatro niveles lógicos: los primeros tres niveles, T0, T1, y T2, estaban implementados en hardware, mientras que el cuarto T3 estaba implementado en software y en el filtro en tiempo real. Cada nivel era activado cuando el nivel anterior mandaba una señal positiva [25]

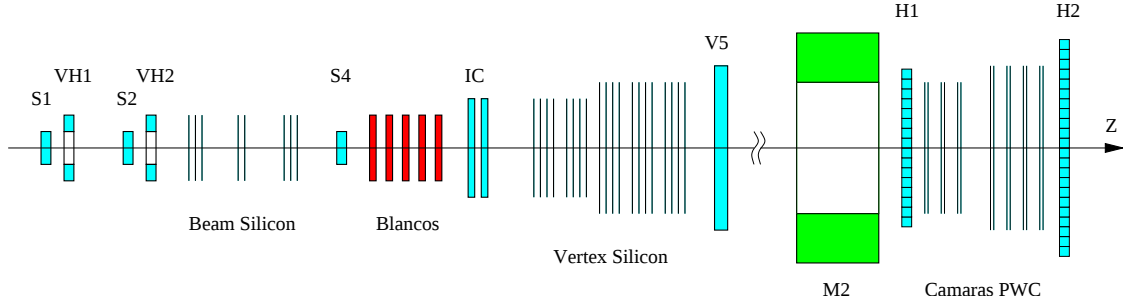


Figura 1.12: Elementos que participan en el trigger.

En el nivel T0 se define a una partícula del haz como la coincidencia de las señales de los contadores de centelleo, S1, S2 y S4, sin tener señal en los contadores veto, VH1 y VH2. Al tener una señal positiva se activaba el nivel T1. Este nivel utilizaba la información obtenida de los dos contadores de interacción IC, seleccionando eventos que tuvieran interacción en los blancos y produjeran un quark Charm. T1 utilizaba también la información del BTRD para identificar las partículas del haz, y por último las señales de los dos hodoscopios se utilizaban para seleccionar eventos cuyas trayectorias tuvieran su origen cerca de los blancos de producción. Los contadores IC generaban una señal de salida con la amplitud proporcional al número de partículas que los cruzaban. Esta señal era tomada como señal de interacción en los blancos siempre y cuando fuera mayor a la señal de tres partículas de mínima ionización.

¹³kick patada

La señal del BTRD no se utilizó en las primeras corridas. Nuevamente una señal positiva de T1 activaba el nivel T2, éste utilizaba la información de los detectores centelladores Vee para seleccionar un decaimiento de una Λ^0 .

1.9. Filtro en línea

El filtro en línea corría a la par del experimento para realizar un análisis en tiempo real de los datos obtenidos. Su principal función era la de filtrar eventos que no tuvieran evidencia de un vértice secundario. Primero se reconstruían trayectorias con momentos mayores a 15 GeV/c ya que son las trayectorias que lograban atravesar los imanes y llegar a las cámaras de alambre (PWC) que se encontraban en el espectrómetro M2. Las trayectorias reconstruidas eran extrapoladas desde las PWC al espectrómetro del vértice, después un programa de reconstrucción buscaba segmentos de las trayectorias en la región del vértice. Finalmente la trayectoria del haz y las trayectorias reconstruidas de las cámaras y de la región del vértice fueron ajustadas a un vértice primario. Si el ajuste tenía una χ^2 aceptable, significaba que todas las trayectorias eran utilizadas y que todas provenían de un vértice común, por lo que esos eventos no se conservaban. Si por el contrario una o dos trayectorias apuntaban a un vértice diferente, éste evento era guardado por el filtro. El filtro en línea disminuyó el ruido en un factor ocho y fué 50 % eficiente para una señal típica de charm.

1.10. Datos del experimento SELEX

El experimento SELEX comenzó a tomar datos en julio de 1996. Las primeras corridas tuvieron los siguientes objetivos: verificar el disparador (trigger), revisar el aparato en general, establecer la eficiencia de las cámaras, calibrar el detector de fotones, optimizar el filtro en línea así como otras tareas similares, cuyo propósito era lograr una alta calidad en los datos que se escribirían en las cintas magnéticas.

La primera parte de los datos obtenidos se realizaron sin utilizar la señal del BTRD para el nivel T1 del trigger. Por lo que las interacciones de ambos π^- y Σ^- no fueron diferenciadas y se guardaron en las cintas (grupos pb y pc del Cuadro 1.3). En la segunda parte se utilizó la señal del BTRD en T1, obteniendo así la mayoría de las interacciones con Σ^- (grupos pd, pe, pf, pg, ph, px y pz). Por último, se invirtió la polaridad del haz de hiperones y se obtuvo un haz en su mayoría de protones (grupo pp). El total de las interacciones obtenidas se pueden

Conjunto de datos	Corrida
pb1	5587 - 6312
pb2	6313 - 7012
pc1	7018 - 7381
pd	7382 - 8079
pe	8087 - 8673
pf	8683 - 9046
pg	9057 - 9411
ph	9502 - 9789
px	9806 - 10243
pz	10244 - 10858
pp	10876 - 11313

Cuadro 1.3: Datos tomados de SELEX utilizados en el análisis de datos.

ver en la Figura 1.13. La cantidad de interacciones almacenadas fué de 1×10^9 de los 15.2×10^9 de las interacciones totales¹⁴.

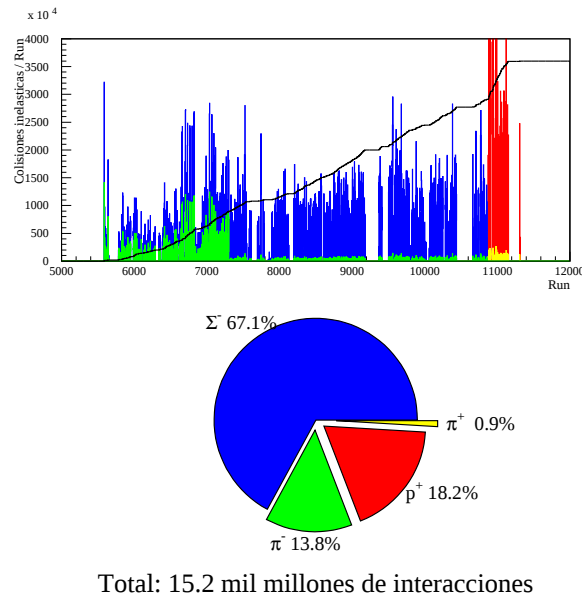


Figura 1.13: Toma de datos.

¹⁴Total de interacciones: en EE.UU. 15.2 billions, en México 15.2 mil millones

1.11. Procesamiento de datos

1.11.1. SOAP

Se utilizó un programa de procesamiento de datos llamado SOAP¹⁵ para realizar el análisis de todos los datos obtenidos por el experimento SELEX. Éste programa, como otros que se utilizan en este análisis, fué escrito por los investigadores que participaron en el proyecto, no son programas que hayan sido comprados. Este programa tiene 220861 líneas, está escrito en su mayoría en lenguaje de programación FORTRAN, y se divide en otros cinco paquetes, que se describen a continuación:

- UNPACK: El propósito de este paquete es desempacar los datos que están grabados en las cintas magnéticas. Las señales que se generaban en cada detector por algún cable, las tiras de silicio o los fotomultiplicadores, eran guardadas en una lista que creaba el hardware. Esta lista de señales es convertida por el software en una nueva lista de posiciones en el sistema de coordenadas local de cada detector.
- TRACKING: La finalidad de TRACKING es reconstruir las trayectorias que siguieron las partículas en su paso por los detectores. Para hacerlo el programa busca y mide los parámetros siguientes: tres coordenadas espaciales del punto donde fue observado por primera vez, tres vectores momento, el identificador de la partícula y la coordenada z donde fue vista por última vez.

El proceso que utiliza el programa es el siguiente:

1. Se prepara una lista de las coordenadas de las señales que se obtuvieron de cada espectrómetro y se prueba el hecho de que pueden ajustarse a una línea. Si el ajuste es bueno, las señales se guardan en una nueva lista como segmentos de la trayectoria que recorrió la partícula a través de cada espectrómetro
2. A los segmentos de cada espectrómetro se les une entre sí para formar un candidato a trayectoria entre detectores. El algoritmo utilizado para esta parte es similar al del paso 1. Los segmentos son ajustados por pares de espectrómetros a una trayectoria continua. Los segmentos pares que tengan un χ^2 aceptable son asignados como parte de la trayectoria.
3. Por último se calcula el momento de la trayectoria obtenida. Se realiza un ajuste a una función que describe la trayectoria de la partícula que atraviesa el campo magnético de los imanes y las PWC de los espectrómetros; como parámetro del ajuste se utiliza la curvatura de la trayectoria.

¹⁵SELEX Off/Line Analysis Physics

- PARTID¹⁶: La información obtenida con TRACKING, de los detectores como BTRD, ETRD y RICH, es extrapolada a PARTID para la identificación de partículas. Para la presente tesis se utilizó la información obtenida de BTRD y RICH. Toda la información que identifica a las partículas se guarda en una tabla llamada *recpid*, ésta se encuentra en el archivo llamado *recdf.ocs*, del que se hablará más adelante. La información en esta tabla es: el nombre abreviado de la partícula, el sistema usado para su identificación (RICH, BTRD), los cortes en likelihood, los momentos mínimo y máximo, la distancia mínima y máxima, el mínimo y el máximo del momento transversal de la partícula. Estos sólo son cortes suaves para reducir la señal de salida y reducir el tiempo de cálculo de las computadoras, después se aplicarán cortes más duros con la ayuda de otros programas.
- VERTEX: El siguiente paso es encontrar las intersecciones de las trayectorias, esta intersección es llamada vértice. VERTEX encuentra las coordenadas de los vértices primarios y secundarios. El vértice primario indica la posición de la interacción del haz y alguno de los blancos y las trayectorias que se producen en este choque. Los vértices secundarios son aquellos que muestran la posición del decaimiento de la partícula madre. Para encontrar estos vértices se cuenta con tres paquetes: *v1*, *v2* y *candidate*. Para este trabajo se utilizaron *v2* y *candidate*:
 - *v2*: este paquete hace una lista de todas las posibles trayectorias. Después hace un ajuste de las trayectorias junto con la del haz a un punto, este ajuste debe tener χ^2 menor a 3.5 para poder designar el punto como vértice primario. Las trayectorias que no pertenecen al vértice primario se toman para revisar si forman parte de un vértice secundario.
 - *candidate(c)*: este paquete se basa en una lista de datos que crea el usuario con las especificaciones para el tipo de partículas que busca y no requiere que χ^2 sea mayor a 3.5,. Candidate revisa la información de PID y busca eventos que tengan las partículas solicitadas y revisa si proceden de algún punto común mediante un ajuste como la rutina anterior. Si el ajuste es bueno, guarda este punto como un vértice secundario. Para el vértice primario utiliza todas las trayectorias que no se utilizaron para el secundario.
- RECON: La finalidad de RECON es proporcionar un código estandar que se maneja con un lenguaje de alto nivel, con el cual se pueden reconstruir las partículas hipotéticas utilizando la información de los paquetes anteriores TRACKING, VERTEX y PID. Una hipótesis de una partícula reconstruida es el conjunto de trayectorias que se identificaron antes y que vienen de un punto común, es decir de un vértice. Los parámetros que se utilizan son: las coordenadas de los vértices primario y secundario con sus errores asociados (σ), la separación entre éstos vértices (L), el cuadrimomento (p_μ), la masa (m) y error, el identificador de las trayectorias y la carga (q). RECON ejecuta

¹⁶Particle Identification

un programa de alto nivel el cual esta almacenado en una tabla OCS¹⁷, en este trabajo la tabla se encuentra en el archivo llamado *recdf.ocs* (Figura 1.14). Con los parámetros de esta tabla el programa busca los eventos que tienen una reconstrucción de una partícula que coincida con el criterio de búsqueda. El significado de cada línea de la tabla se explica a continuación [26]:

- id : número de identificación único para la reconstrucción.
- name : nombre de la reconstrucción.
- from : tipo de búsqueda.
- pr : número de prongs en la reconstrucción.
- q : carga de la partícula reconstruida.
- pid : partículas producto.
- ls_min/max : rango de L/σ .
- mass_min/max : rango de masa.
- out: bits para los datos de salida.

Se ve en la tabla *recdf.ocs* que para las partículas con id de 801 a 805, en la opción de **from** aparece **rr**, que quiere decir *recon_recon*. Este es parte de RECON aquí se hacen dos reconstrucciones de partículas por separado (utiliza *v1*, *v2* o *c*) y después las combina para obtener una reconstrucción completa.

Para controlar el uso de los paquetes y rutinas vistos anteriormente, se utiliza un archivo llamado *test.cmd*. En este archivo se encienden o apagan los paquetes por separado, también se activan o desactivan los diferentes cortes que contiene cada paquete. Además contiene la ubicación de los archivos a correr y la cantidad de eventos que se quiere analizar. Se puede ver un ejemplo de este para usar el paquete de datos pb1:

```
in soap_data_pass2.cmd.part

exec tracking vertex partid unpack recon

set off recon ftup1 ftup2 ftup3
set on recon ftup4

set off recon rec_refit2
set on recon rec_refit1

exec user

disk in l/strip-pass2.lambda.p2b01.charm_run005587_006021.out1
```

¹⁷Open Constant System

```

disk in l/strip-pass2.lambda.p2b01.charm_run006022_006171.out1
disk in l/strip-pass2.lambda.p2b01.charm_run006172_006312.out1

#recdf      0      0 fill anal v04.12 Jan 19 1998 14:31 psc ! pass11 mode recon list
! pass11 production - change from _11: add vx vees to charm recons
! id name      from pr q pid      ls_min ls_max mass_min mass_max      out[]
!
! Strange states
2 10          v2 2 0 p+i-      5.0 800.0      1.090      1.140      $000008
3 alam        v2 2 0 p-i+      5. 800.      1.090      1.140      $100008
4 l2          c 2 0 p+i-      0. 0.      1.060      1.112      $000008
5 al2         c 2 0 p-i+      0. 0.      1.060      1.112      $000008
!
801 dilambda  rr 4 0 l0l0      0. 0.      2.000      4.000      $000008
802 adilambda rr 4 0 alamalam  0. 0.      2.000      4.000      $000008
!
803 wrong     rr 4 0 l0alam     0. 0.      2.000      4.000      $000008
804 H         rr 4 0 l0l2      0. 0.      2.000      4.000      $000008
805 aH        rr 4 0 alamal2    0. 0.      2.000      4.000      $000008
!
*end
!
#recpid      0      0 fill anal v04.2 21-Nov-1997 23:00 psc ! Particle id list
!
! name system cut      pmin pmax mdistmin mdistmax ptmin ptmax spectro logic
!
e etrd e>0.5 0. 0. 0. 0. 0. 0. m2 none ! default electron
i rich i<0.1 0. 0. 0. 0. 0. 0. any not ! not non-pi if in rich
il none none 0. 0. 0. 0. 0. 0. any none ! not non-pi if in rich
k rich k/i>=1 0. 0. 0. 0. 0. 0. m2 none ! default Kaon
p rich p/i>=1 0. 0. 0. 0. 0. 0. m2 none ! default proton
e1 etrd e>0.5 0. 0. 0.0020 0.0500 0.300 2.000 vxm2 none ! non-pvtx electron
k1 none none 0. 0. 0. 0. 0. 0. any none ! any track
k2 rich k/i>=1 0. 0. 0. 0. 0. 0. any not ! any track not a K
k3 rich k<0.1 0. 0. 0. 0. 0. 0. any not ! not non-k if in rich
p1 rich p/i>2. 100. 600. 0. 0. 0. 0. any none !
p2 rich p/i>2. 0. 0. 0. 0. 0. 0. m1m2 none ! lambda daughter proton
p3 none none 0. 0. 0. 0. 0. 0. any none ! any track
s none none 30. 600. 0. 0. 0. 0. vxm1nm2 none ! m1_kink
x none none 30. 600. 0. 0. 0. 0. vxm1nm2 none ! m1_kink
o none none 30. 600. 0. 0. 0. 0. vxm1nm2 none ! m1_kink
gg2 phot none 2. 8. 0. 0. 0. 0. m1m2m3 none ! any
gg3 phot none 2. 8. 0. 0. 0. 0. m2m3 none
!
*end

```

Figura 1.14: Tabla *recdf.ocs*.

1.11.2. Pass

Pass es el procesamiento de los datos crudos utilizando SOAP, que mediante cortes suaves y algoritmos de reconstrucción redujo el tamaño de los datos y con esto se facilitó el análisis posterior. Los datos del experimento se corrieron dos veces, cada uno tardó en correr medio año:

- pass1: este es el primer procesamiento que se realizó. Se generaron cinco archivos donde se guardaron las reconstrucciones:

1. Mesones con quarks Charm.
2. Bariones con quarks Charm.
3. Decaimientos semileptónicos.
4. Estados extraños.
5. Partículas exóticas.

si alguna reconstrucción se repetía en dos o más grupos de la lista, esta era guardada en cada uno de los grupos a los que pertenecía, lo cual provocaba que el tamaño de los archivos fuera mayor. La información obtenida se guardó en archivos llamados *ftuples*, los cuales tienen un tamaño fijo. En los *ftuples* se guardó la información en 82 variables, en las cuales encontramos información acerca de los vértices, trayectorias, masa, momento, etc. Los datos procesados con pass1 fueron llamados como a los datos brutos (pb01, pb02, etc)

- pass2: alguna de las diferencias con el procesamiento anterior es la manera como se guardó la información, los archivos que se generaron fueron llamados *vtuples* que son archivos de tamaño variable. Estos pueden guardar un número no constante de variables y esto ayuda a ahorrar espacio, además guardaban información extra como de los vees y kins. Se incluyeron más tipos de reconstrucciones. Si al menos una reconstrucción de la lista fue encontrada el evento fue guardado en un archivo de salida llamado *out1* y aquellas reconstrucciones que no aparecían en la lista fueron guardadas en otro archivo de salida llamado *out2*. Para nombrar a los datos procesados con pass2 se cambió p por p2. Por ejemplo de pb01 pasó a ser p2b01.

1.11.3. Strip

Otra etapa es el strip, que siguió después del pass, que se usa cuando se necesita calcular mejor, por ejemplo algunas veces usando los *vtuples* o *ftuples* la información puede no servir o ser insuficiente.

El control de todo el proceso en el SOAP se realiza por medio de un archivo de comandos (strip). Cualquier paquete de SOAP puede ser ejecutado o no ejecutado utilizando este strip, así como también se puede usar para activar o desactivar diferentes opciones de cada paquete y además puede usarse para aplicar cortes sobre diferentes variables en cada paquete.[12]

Por ejemplo para buscar eventos con una partícula Λ^0 , no era conveniente correr todos los datos. Para esto se escribió un programa de análisis especial que buscara partículas Λ^0 y escribiera estos eventos en un archivo aparte, guardando el número de corrida (run number) y el número de evento (event number). Se tiene entonces una lista con todos los eventos que tienen una partícula Λ^0 , a este archivo se le conoce como ftupselect. Para que lea la lista se coloca un switch en el archivo .cmd. Si el archivo de salida es muy grande se abre otro archivo. Para este trabajo se utilizó la información guardada en la dirección ../strip-pass2.lambda.p2b01.charm_run005587_006021.out1, con los paquetes desde p2b01 hasta p2z01.

1.12. Simulación

El principal objetivo de la simulación es revisar la aceptación del aparato SELEX, así como para revisar que los programas funcionen. Para este trabajo se utilizó la simulación para obtener más información de la partícula a estudiar. La simulación puede ser dividida en dos pasos: generar los eventos y la incrustación de éstos en rutinas de SELEX. Para la producción de eventos se utiliza el generador de eventos Embedding (EDG¹⁸) de SELEX [31] llamado QQ.

La producción de eventos se especifica en un archivo de control que contiene la siguiente información: número de eventos a generar, identificación de la partícula a generar, identificación de la partícula del haz, momento de la partícula del haz, forma de la distribución x_F y forma de la distribución p_T . La forma de las distribuciones de x_F y p_T se especifican por medio de los parámetros n y b de la ecuación (1.2)

$$\frac{d\sigma}{f x_f dp_T^2} = A * (1 - x_F)^n * \exp(-bp_T^2) \quad (1.2)$$

Donde b y n son parámetros libres, p_T es la magnitud del momento transversal definido por $p_T = \sqrt{p_x^2 + p_y^2}$ y x_F es la variable de escalamiento de Feynman que se define como $x_F = p_z/p_{max}$, p_{max} es el momento máximo que puede tener la partícula producida.

¹⁸Embedded Data Generator

El archivo que se obtiene es un archivo con extensión *.emb*, el cual contiene información de los productos del decaimiento de los eventos que se generaron. La información la presenta en forma de tabla: en la primera línea se especifica el número de partículas y la carga eléctrica de cada una. Las siguientes líneas se dividen en ocho columnas: en las primeras tres columnas, siendo el vértice primario en (0,0,0) se obtiene el punto donde se generó la partícula con las tres coordenadas (x,y,z) , las siguientes tres columnas son los cosenos directores (α, β, γ) y el momento y masa de la partícula son las últimas dos columnas. Después se utiliza el programa llamado Embedding que se encuentra integrado en SOAP. Este programa simula eventos en todo el aparato SELEX [27]. Embedding utiliza los eventos que se generaron con EDG y encuentra un vértice primario de un evento real. Después utiliza el vértice primario (0,0,0) y las trayectorias del evento generado y realiza una traslación al vértice primario del evento real. Y así es como se realiza la simulación con Embedding.

1.13. Physics Analysis Workstation (PAW)

Una herramienta de análisis que desarrolló el CERN¹⁹ y que se utiliza en este trabajo de tesis fué PAW. PAW es una herramienta interactiva que permite el análisis, visualización y presentación de datos experimentales en la computadora.

Este utiliza los archivos de salida, con extensión *.hbk*, que se producen al analizar los *ftuples*. Los archivos con extensión *.hbk*, contienen la lista de todos los histogramas generados. PAW permite visualizar estos histogramas, y además cuenta con herramientas que permite su manipulación. Los comandos para manipular a los histogramas pueden ser especificados en un archivo *.kumac* que puede ser ejecutado en PAW. Algunas de las aplicaciones de las herramientas de PAW son: operaciones aritméticas con los histogramas, ajuste de funciones a las distribuciones estadísticas, gráficas de funciones, gráficas de resultados para usarlos en presentaciones o publicaciones etc, [28][29]

¹⁹European Organization for Nuclear Research

Capítulo 2

Análisis

En la sección 1.11.1 del capítulo 1 se describió el programa SOAP, así como los diferentes paquetes que lo forman, los cuales se utilizaron para el procesamiento de los datos obtenidos del experimento SELEX.

Uno de los paquetes de SOAP, RECON, incluye el programa *recon_recon.F*. Este programa realiza dos reconstrucciones de partículas, sólo para partículas diferentes entre sí, cada una por separado, ya sea utilizando *v2* o *c*, y después las combina para obtener una reconstrucción completa.

Para este trabajo de tesis en el que se buscaron las partículas del decaimiento $H^0 \rightarrow \Lambda^0 \Lambda^0$, era necesario que el programa encontrara dos partículas iguales en el mismo evento, por lo que se estudió todo el programa *recon_recon.F* para poder realizar una mejora. Finalmente se mejoró agregando unos renglones al programa, se puede ver el programa *recon_recon.F* completo con esta mejora en el apéndice B. Los renglones agregados permiten reconstruir dos partículas del mismo tipo:

```
If (pid_double(irecdf)) Then
  If (nirecn.lt.nr(irecdf)) Then ! nothing found
    Goto 900
  EndIf
Else
  If (No_recn) Then      ! nothing found
    Goto 900
  EndIf
EndIf
```

Como se mencionó anteriormente estos programas fueron escritos por los mismos investigadores que participaron en el experimento SELEX. El programa RECON está escrito de una manera general, lo que lo hace flexible para poderlo modificar y así poder analizar las partículas.

También se elaboró la tabla *recdf.ocs*. Se crearon las siguientes partículas (Figura 1.14):

- Λ^0 (10): su reconstrucción se llevó a cabo con el paquete de VERTEX *v2*, con las partículas de decaimiento p y π^- . Con un rango de L/σ de 5 a 800 y con un rango en la masa invariante de 1.09 a 1.140 GeV/c² (la masa invariante de Λ^0 es de 1.1156 GeV/c²). La reconstrucción de esta partícula Λ^0 se utilizó para la reconstrucción de la partícula $\Lambda^0\Lambda^0$ y $\Lambda^0p\pi^-$.
- $p\pi^-$ (12): para su reconstrucción se utilizó el paquete *c* de VERTEX, de igual manera con las partículas de decaimiento p y π^- . En esta ocasión no se dió un rango para L/σ y se le dió una masa de 1.06 a 1.112 GeV/c², ligeramente por debajo del valor conocido de la masa invariante de Λ^0 .
- $\Lambda^0\Lambda^0$ (dilambda): aquí se utiliza el programa mejorado *recon_recon.F*, que utiliza la reconstrucción de dos partículas Λ^0 (10), sin rango en L/σ y rango en masa de 2 a 4 GeV/c².
- $\Lambda^0p\pi^-$ (H): igualmente utiliza el programa mejorado *recon_recon.F*, utilizando la reconstrucción de las partículas Λ^0 (10) y $p\pi^-$ (12). Nuevamente sin rango en L/σ y rango en masa de 2 a 4 GeV/c².

La partícula Λ^0 tiene mucho tiempo de vida y con el boost de Lorentz, una partícula Λ^0 típica, puede volar en promedio 8 m. Sin embargo ésta partícula se reconstruye con el paquete *v2* y éste sólo encuentra partículas en 12 cm, es decir sólo encuentra como 1 % de todas las Λ^0 que se generan en el experimento, por lo que para el decaimiento $\Lambda^0\Lambda^0$ sólo encontrará el cuadrado de 1 %. Para corregir esto se necesitan de paquetes más avanzados y quedará como trabajo a futuro.

Se creó un subdirectorío por cada paquete de datos desde p2b01 hasta p2z01 y se corrió el programa SOAP en cada subdirectorío para evitar que se sobrescribiera la información. Se utilizó el archivo *test.cmd* (página 26). Los archivos de salida que se obtuvieron, se guardaron juntos en una carpeta para el análisis siguiente, con nombres como: recon.ftup4.p2b01, etc.

Los datos que se obtuvieron se analizaron con un programa llamado *anal.F* (Apéndice C), así como los programas que fueron escritos por los investigadores. El programa *anal.F* fue desarrollado por la autora de este trabajo de tesis, por lo que se puede modificar para que sea específico para cada análisis que se necesite. En éste programa se utilizan algunas de las 82 variables de los *ftuples* para realizar cortes más específicos. Las variables sobre las cuales se realizaron los cortes y algunas especificaciones son las siguientes:

L/σ : la partícula Λ^0 tiene un tiempo de vida largo, 2.63×10^{-10} s y el ruido es generado por partículas con tiempo de vida corto, las cuales decaen inmediatamente por la fuerza fuerte cerca del vértice primario. Por lo tanto el corte L/σ es uno de los cortes más importantes debido a que ayuda a reducir el ruido, este se compone de:

- L : es la distancia que existe entre el vértice primario (donde se genera la partícula madre) y el vértice secundario (donde ocurre el decaimiento) sobre el eje z ($L = z_{sec} - z_{prim}$).
- σ : es la combinación de errores del ajuste de los vértices: $\sigma = \sqrt{\sigma_p^2 + \sigma_s^2}$, donde σ_p^2 y σ_s^2 miden la incertidumbre de la medición de la posición de los vértices primario y secundario.

Pointback (*pvtx*): Es una medida de que tan bien el vector momento de la partícula madre apunta hacia el vértice primario. Esto se calcula sumando los vectores momento de las partículas hijas generando un vector resultante, el cual se extrapola hasta el plano donde se encuentra el vértice primario. Después se mide la distancia entre la extrapolación y el vértice primario. El pointback es esta distancia al cuadrado dividida por su error.

χ^2 : Un valor pequeño de esta variable, indica que el ajuste realizado es bueno.

Masa invariante: cuando una partícula decae en varias partículas (como en este caso), el cuadrado del cuadrimomento correspondiente a la partícula madre se calcula de la siguiente manera

$$P^2 = P^\mu P_\mu = \frac{E^2}{c^2} - \vec{p}^2 = m^2 c^2$$

entonces:

$$P^2 = m^2 c^2$$

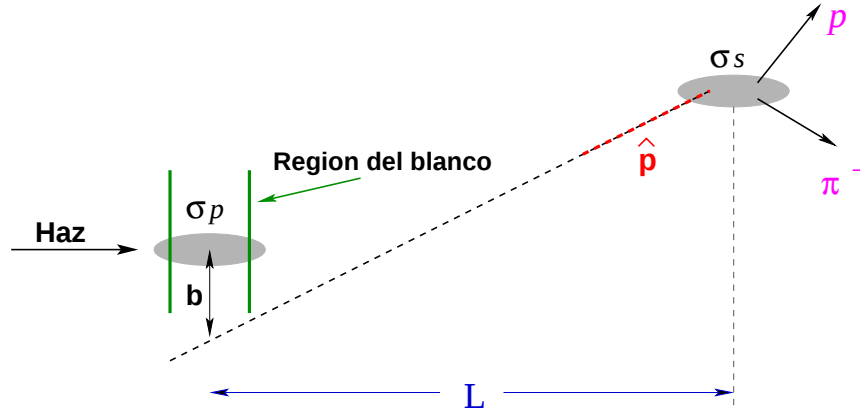
Finalmente, se despeja m y queda:

$$m = \frac{\sqrt{P^2}}{c}$$

esta m simboliza la *masa invariante*. La variable P^2 es la suma del cuadrimomento de todas las partículas hijas. La información de las partículas hijas se obtienen de los diferentes detectores de SELEX, por ejemplo para el momento se utiliza el detector de partículas y los campos magnéticos.

2.1. Partícula Λ^0

Para reconstruir a la partícula Λ^0 se usan las partículas de su decaimiento p y π^- (Figura 2.1), estas partículas son seleccionadas con las condiciones que se dieron en la tabla *recdf*: con el paquete RECON, usando $v2$ y un corte de $L/\sigma > 5$.

Figura 2.1: Esquema de la reconstrucción de una Λ [30]

Al graficar la masa invariante de los candidatos a Λ^0 se obtiene una distribución centrada alrededor del valor 1115.6 MeV/c² con un ruido a los costados del pico que se comporta como una recta (polinomio de orden 1). Para conocer el número de partículas se realiza el ajuste de una función gaussiana más una recta (Ecuación (2.1)).

$$gauss(x1) = \frac{b}{\sqrt{2\pi}} \frac{p_1}{p_3} \exp\left(\frac{-1}{2} \left(\frac{x1}{p_3}\right)^2\right) + p_4 + p_5 x1 \quad (2.1)$$

- $x1 = x - p_2$ es una traslación a dónde se encuentra el máximo de la gaussiana
 - x : masa invariante de la partícula
 - p_2 : es el punto máximo de la gaussiana, el cual se espera sea igual a la masa invariante de la partícula
- b : ancho del bin¹
- p_1 : número de eventos debajo de la curva gaussiana y arriba del ruido
- p_3 : es σ , donde su valor al cuadrado es la varianza de la gaussiana
- p_4 y p_5 son los parámetros de la recta
 - p_4 : es el valor donde la recta cruza al eje vertical, es decir, nos da la altura del ruido y con la traslación se tiene este valor en el punto máximo de la gaussiana
 - p_5 : es la pendiente

La función está normalizada al número de eventos en la señal, p_1 . Al realizar este ajuste tenemos que el número de partículas Λ^0 es: 1.4 millones (Figura 2.2).

¹Bin son los intervalos en los que se divide el eje x . El ancho del bin es el ancho de estos intervalos

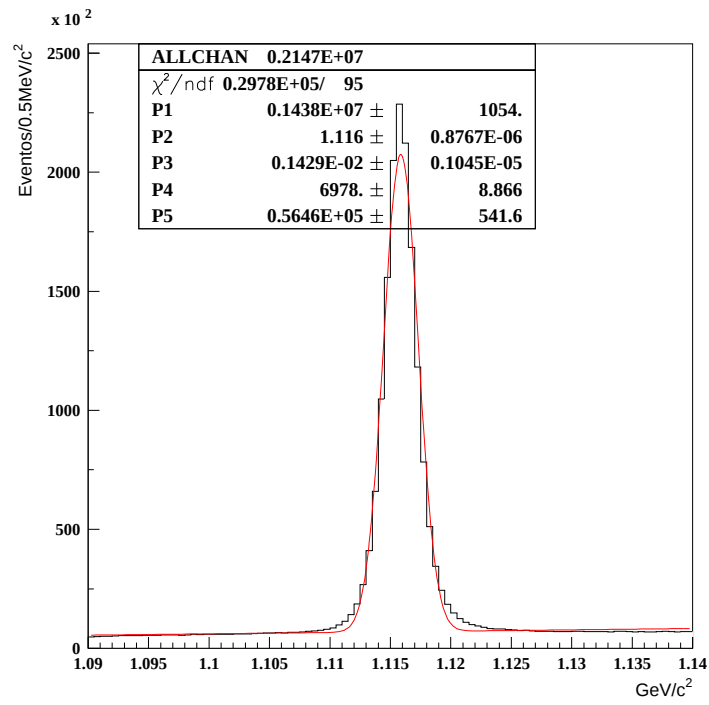


Figura 2.2: Masa invariante de p y π^- , la acumulación en el valor $1.115 \text{ GeV}/c^2$ es la partícula Λ^0

2.2. Decaimiento en $\Lambda^0\Lambda^0$

De acuerdo con la predicción de Jaffe, para la partícula H^0 , si la masa de la partícula $H^0 > 2230$ MeV/c² el decaimiento que se presenta es fuerte. El decaimiento fuerte sería $H^0 \rightarrow \Lambda^0\Lambda^0$. Para esto se buscó a dos partículas Λ^0 que se encontraran en el mismo evento.

Con un proceso de análisis similar al que se utilizó para la masa invariante de $p\pi^-$ se busca la masa invariante para dos partículas Λ^0 de un mismo evento, con las condiciones que se dieron en la tabla *recdf*: con el paquete RECON, usando el mejorada *recon_recon.F* y sin corte en L/σ ya que es un decaimiento fuerte, tiene poco tiempo de vida y este corte no permitiría verlo.

Para este análisis fue necesario recalcular el pointback, ya que el valor de pointback que se obtiene de los *ftuples* se calcula sin verificar que las partículas hijas apunten a un mismo vértice primario.

Para asegurar que las partículas vengan del mismo vértice, se realiza lo siguiente: se guarda la información del vértice primario (pvx) de la partícula madre en una nueva variable (lpv). También se guarda la información de los vértices secundarios (svx) de las partículas hijas en nuevas variables (lsv). Y así, junto con los errores de posición de los vértices (svx_s) y los momentos (p), se realiza el cálculo para el nuevo pointback (lpvtx). Después se grafica la distribución de este nuevo pointback (Figura 2.3).

Con esto se obtiene el corte: $Pvtx < 5$.

Si la diferencia entre el valor para la masa de Λ^0 entre el obtenido en el análisis y el valor conocido (1.1156 GeV/c²) es menor a 0.005, se usará para los siguientes cálculos el valor 1.1156 GeV/c² (Figura: 2.6).

Por último se calculó la *masa efectiva*, su cálculo es: $E^2 = p^2 + m^2$, al despejar se obtiene $m^2 = E^2 - p^2$, este cálculo se realiza con doble precisión ya que la diferencia entre los valores de E y p es muy pequeña y en la resta se podrían eliminar.

La gráfica de masa invariante para dos partículas Λ^0 que comparten el mismo evento se muestra en la figura 2.4. El pico representa a una partícula que decae en $\Lambda^0\Lambda^0$ y tiene una masa alrededor de 2340 MeV/c². Se aprecia que de 277 eventos sólo se tienen alrededor de 20 partículas que decaen en $\Lambda^0\Lambda^0$. Se realizó el ajuste de una función gaussiana, donde se ve que el ancho es de 14 ± 0.001973 MeV (Figura 2.5).

También se muestra una gráfica de la masa invariante de p y π^- contra la masa invariante de otros p y π^- (Figura 2.6), y de ahí tomamos los datos que se encuentran en el centro (la parte

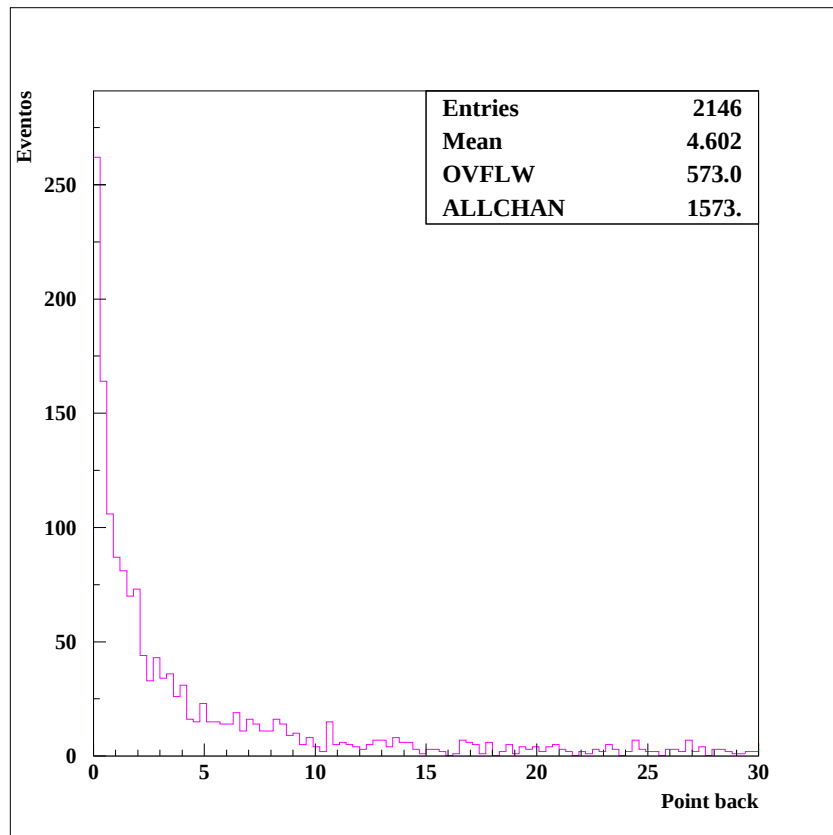


Figura 2.3: Distribución de point back de las partículas $\Lambda^0\Lambda^0$ del mismo evento

sombreada) que puede que sean partículas $\Lambda^0\Lambda^0$, y las usamos para aplicar el método de la masa invariante.

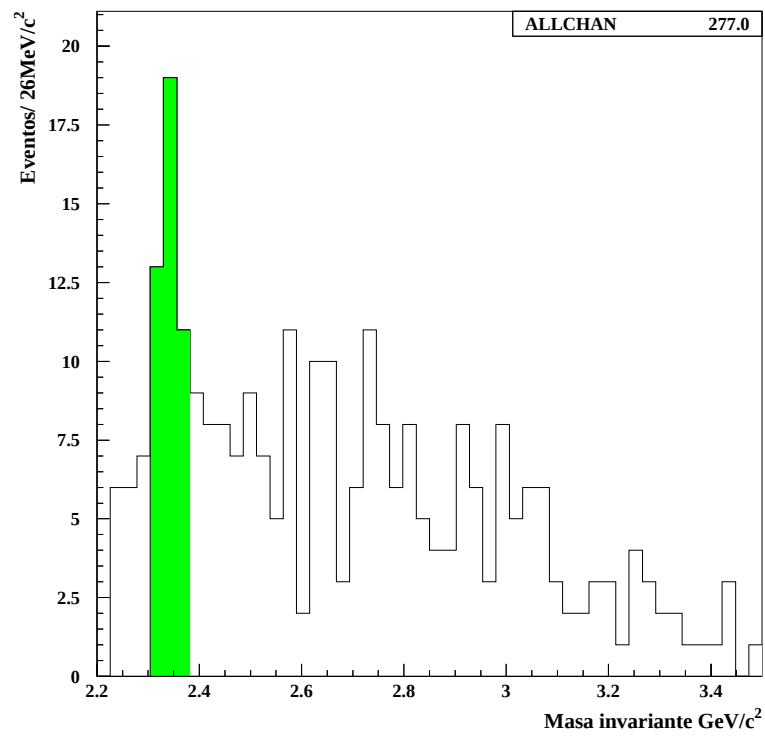


Figura 2.4: Masa invariante de dos partículas Λ^0 en el mismo evento

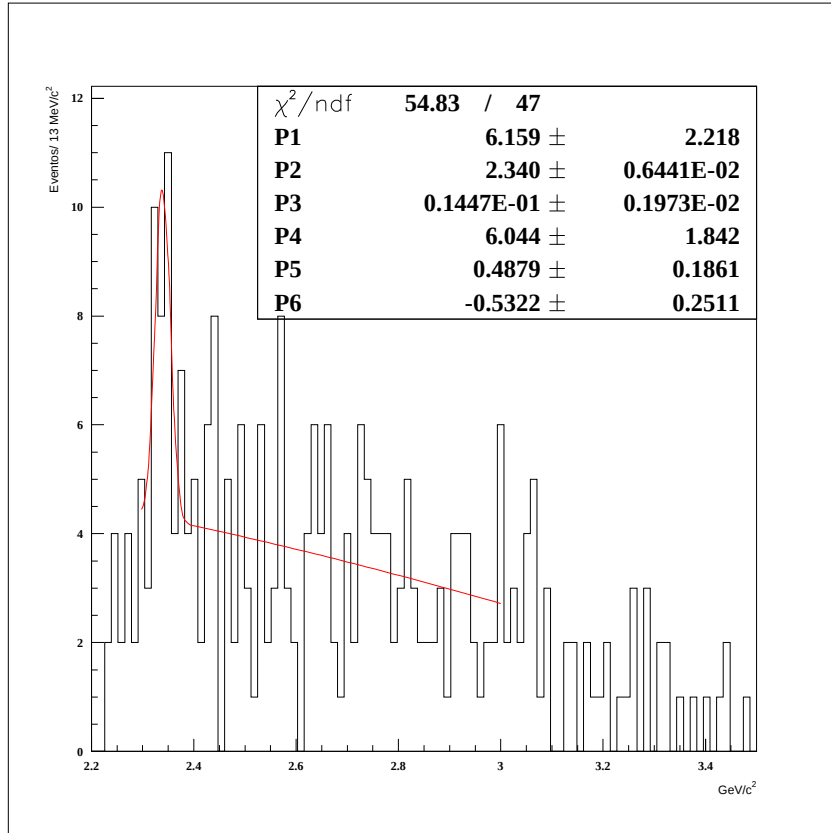


Figura 2.5: Ajuste de una gaussiana a la masa invariante de dos partículas Λ^0 en el mismo evento

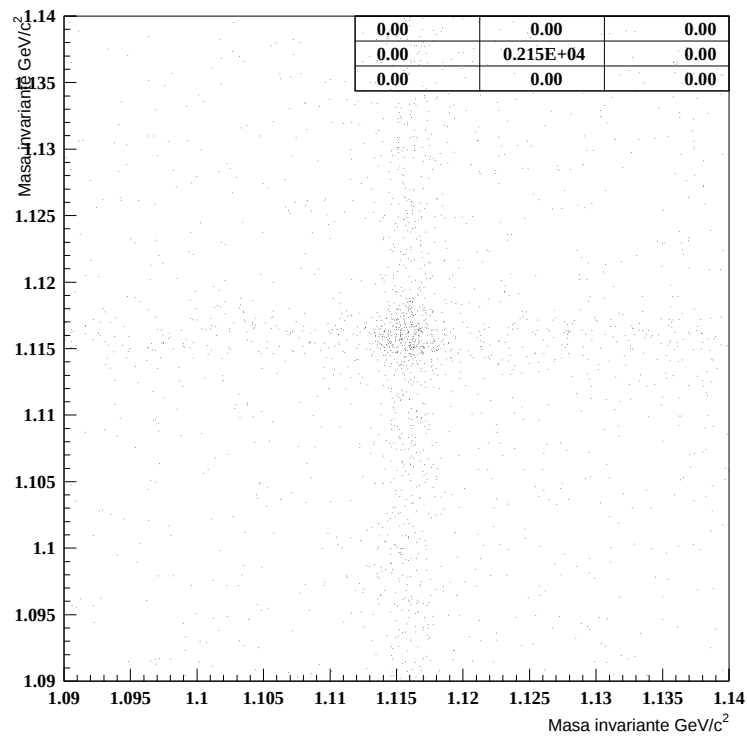


Figura 2.6: Masa invariante de p y π^- contra la masa invariante de p y π^+ , para obtener partículas $\Lambda^0\Lambda^0$

2.3. Decaimiento en $\Lambda^0 p \pi^-$

Siguiendo con la predicción de Jaffe, si la masa de la partícula $H^0 < 2230 \text{ MeV}/c^2$ esta no puede decaer fuerte. Entonces sólo puede actuar la fuerza débil la cual puede cambiar quarks. El decaimiento débil sería entonces $H^0 \rightarrow \Lambda^0 p \pi^-$.

La partícula que decae en $\Lambda^0 p \pi^-$ está formada por una partícula Λ^0 y un par de partículas $p \pi^-$, cuya distribución de masa invariante da un valor ligeramente por abajo del valor conocido de la masa invariante de Λ^0 (Figura 2.7). En la gráfica se observa un escalón el cual está dado porque en los archivos *strip* la partícula Λ^0 tiene un corte en el valor de masa invariante y al cambiar este valor para generar $p \pi^-$, con masa menor, éste corte ya implícito genera el escalón.

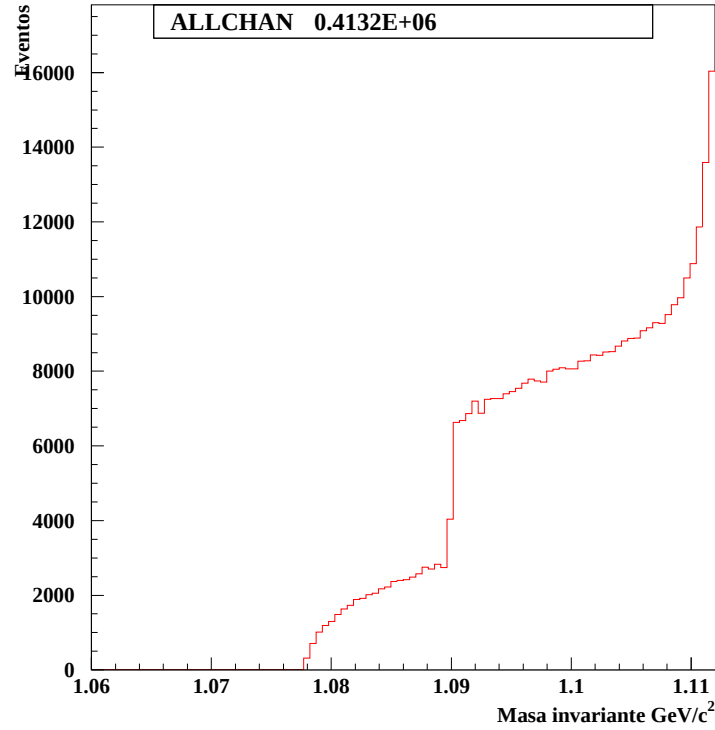


Figura 2.7: Masa invariante de $p \pi^-$, sin el paquete de datos pp

Si la diferencia entre el valor para la masa de la partícula hija Λ^0 entre el obtenido en el análisis y el valor conocido ($1.1156 \text{ GeV}/c^2$) es menor a 0.005, se usará para los siguientes cálculos el valor $1.1156 \text{ GeV}/c^2$.

En este decaimiento ($H^0 \rightarrow \Lambda^0 p \pi^-$), H^0 decae en Λ^0 , p y π^- , posteriormente Λ^0 decae en p y π^- , (Figura 2.8). Se guarda la posición de los vértices secundarios sobre el eje z (svx_z) de estas partículas hijas en nuevas variables (sv) y se hace el corte $L2 > L1$, para diferenciar las partículas hijas de H^0 , p y π^- , de las partículas hijas de Λ^0 .

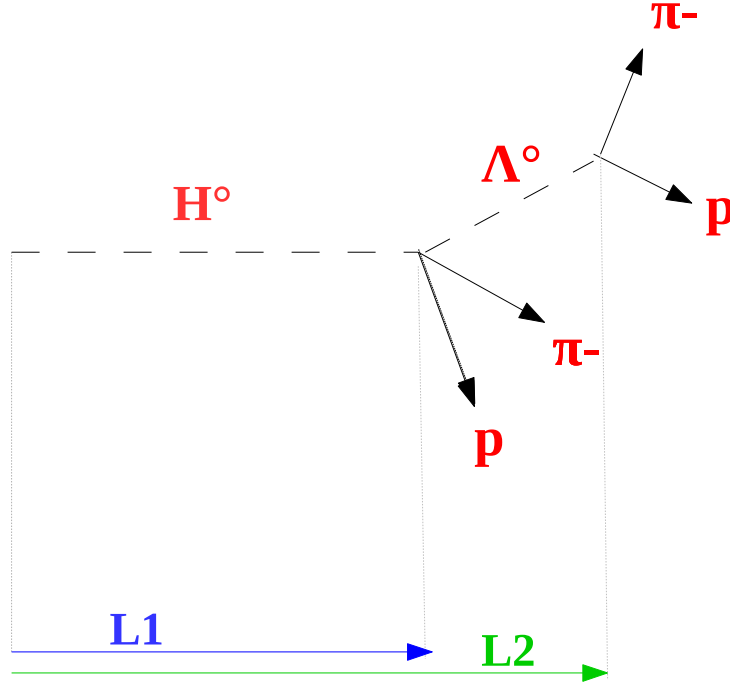


Figura 2.8: Partícula H^0

Después, de igual manera que para $\Lambda^0 \Lambda^0$, se calculó la masa efectiva.

Por último se realizó, con varios valores, el corte de L/σ sobre la partícula hija² $p\pi^-$ (Figuras 2.9 y 2.10) y fue en valor de corte $L/\sigma = 1.9$ (Figura 2.11) donde se presentó una acumulación en el valor de la masa invariante de alrededor de $2215 \text{ MeV}/c^2$. Se realizó el ajuste de una función gaussiana donde se ve que el ancho es de $1.9 \pm 0.001262 \text{ MeV}$ (Figura 2.17).

²Para la partícula hija Λ^0 ya se había realizado un corte de $L/\sigma > 5$.

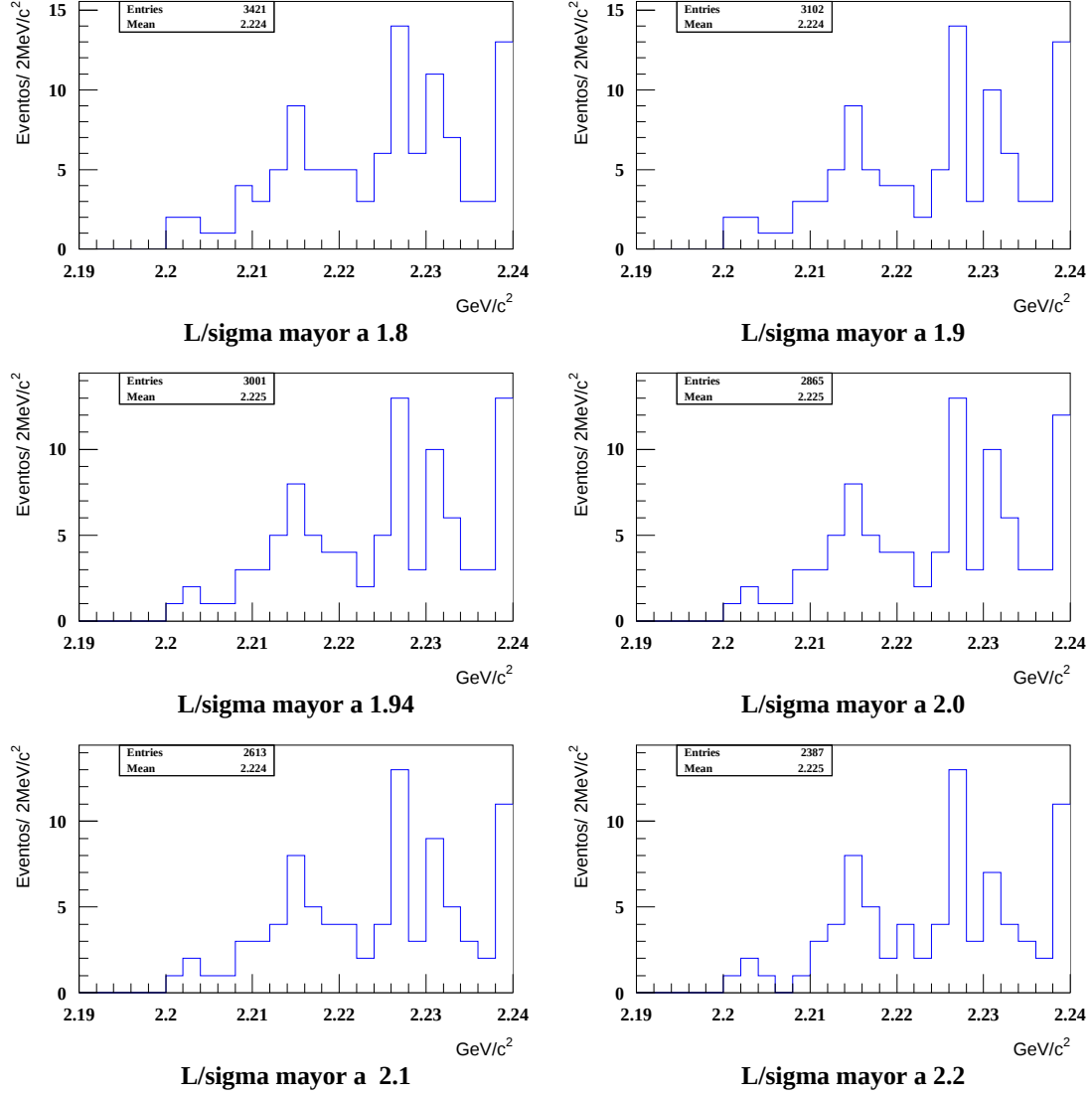


Figura 2.9: Masa invariante $\Lambda^0 p \pi^-$ con cortes en L/σ para $p \pi^-$ de 1.8 a 2.2

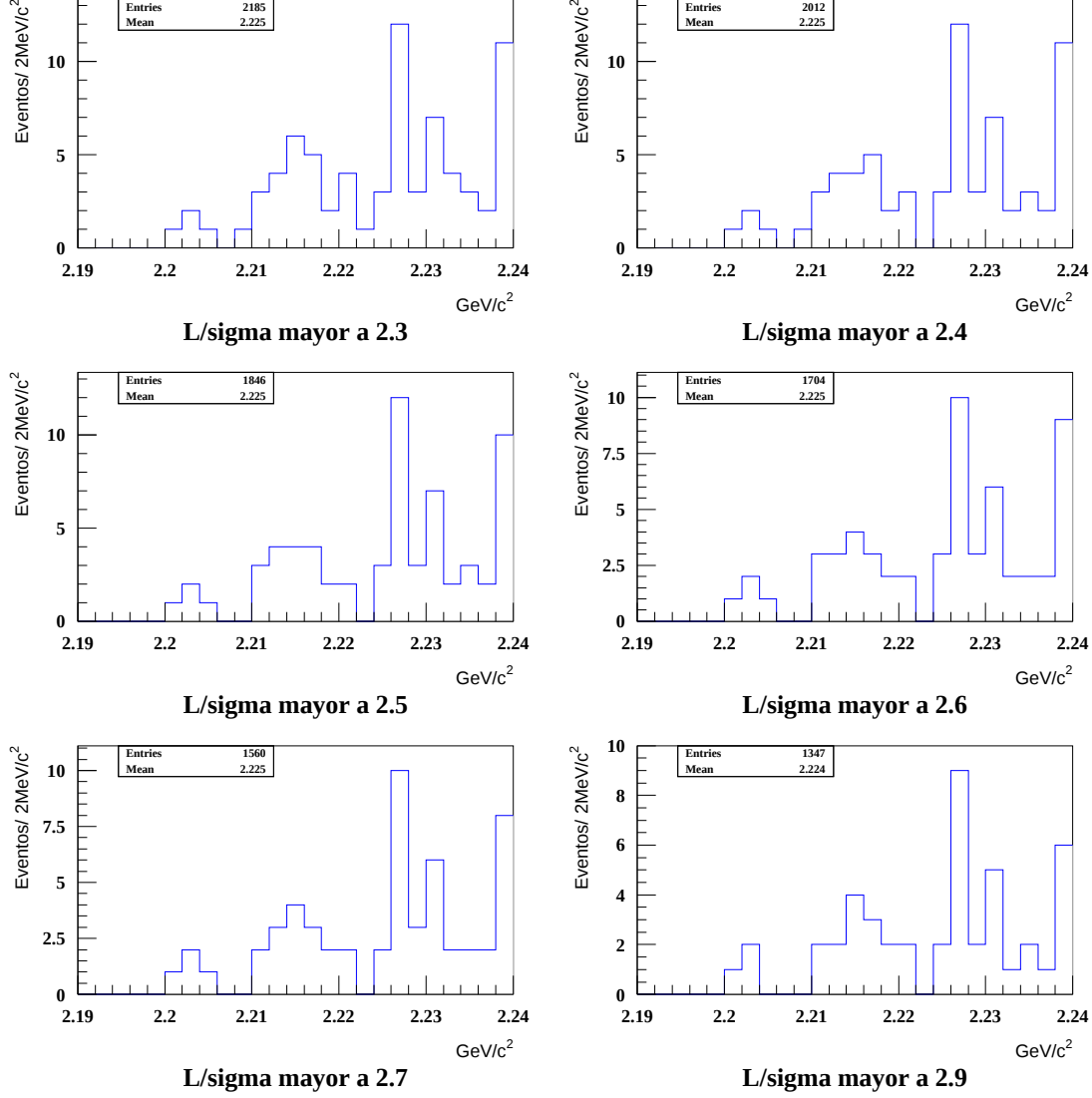


Figura 2.10: Masa invariante $\Lambda^0 p \pi^-$ con cortes en L/σ para $p \pi^-$ de 2.3 a 2.9

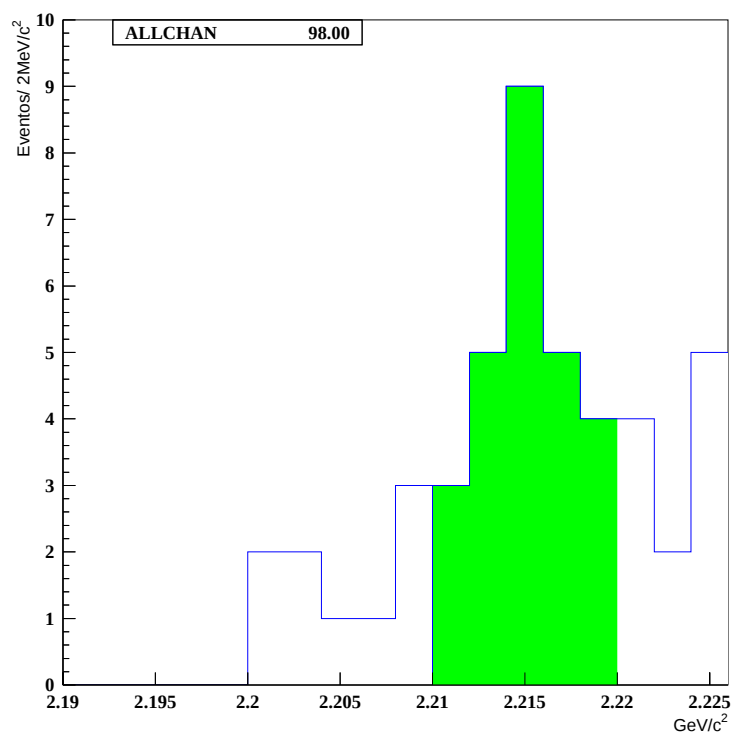


Figura 2.11: Corte en L/σ para $p\pi^-$ de 1.9

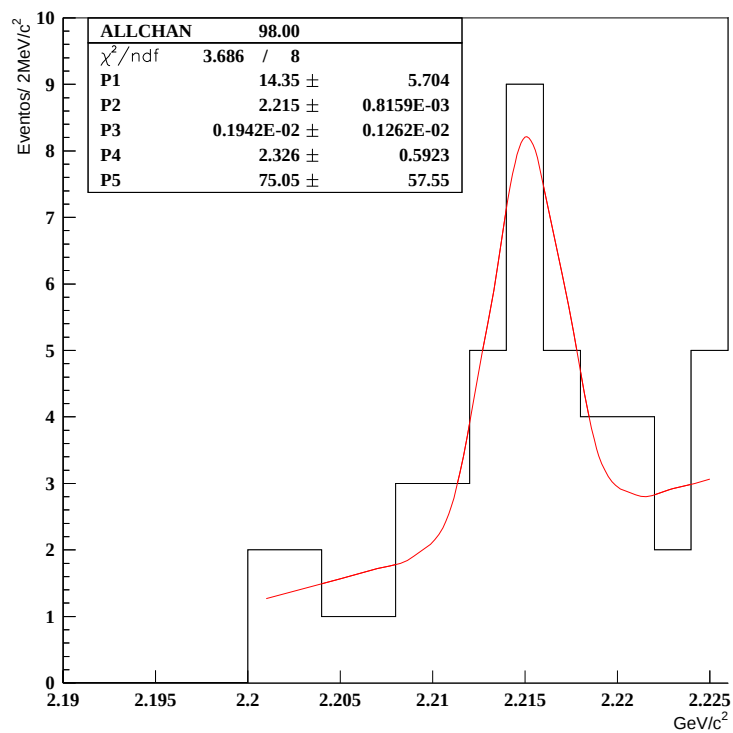


Figura 2.12: Corte en L/σ para $p\pi^-$ de 1.9, con el ajuste de la función gaussiana

Después de estos análisis, se observó en todos los cortes de L/σ una acumulación. Se eligió el valor del corte de $L/\sigma > 1.9$, porque se puede observar mejor este *pico*, el cual es una partícula con número bariónico dos (por Λ^0 y p), esto quiere decir que esa partícula tiene seis quarks como la partícula que predijo Jaffe. Como este *pico* no había sido observado nunca, es de llamar la atención, por ello se decidió correr todos los paquetes de datos nuevamente, incluyendo el paquete del haz positivo pp (Cuadro 1.3), que anteriormente no se había analizado para comprobar que el *pico* es real. Para usar el paquete pp, se cambió su archivo *test.cmd* correspondiente, ya que en el experimento al usar un haz positivo se cambió la polaridad de los imanes, y este cambio también se debe ver en el programa de análisis.

Las líneas agregadas y cambiadas en el archivo *test.cmd* para el paquete de datos pp se ven a continuación:

```
set on recon rec_refit2
set on recon rec_refit1

set cut recon cand_csec 9.

set cut tracking ptm1      -0.7371
set cut tracking ptm2      -0.8285
```

Para el resto de los paquetes de datos (pb a pz), se agregó a *test.cmd* las líneas `set on recon rec_refit2` y `set cut recon cand_csec 9`. Siendo importante la línea `set on recon rec_refit2` pues con este se hace un refit al segundo vértice de la partícula buscada y no solo al primero.

Al correr todos los datos junto con las modificaciones, el número de eventos aumenta, es decir, aumenta la estadística. Se muestran las nuevas gráficas para la masa invariante de p y pi^- y la masa invariante de $p\pi^-$ (Figura: 2.13).

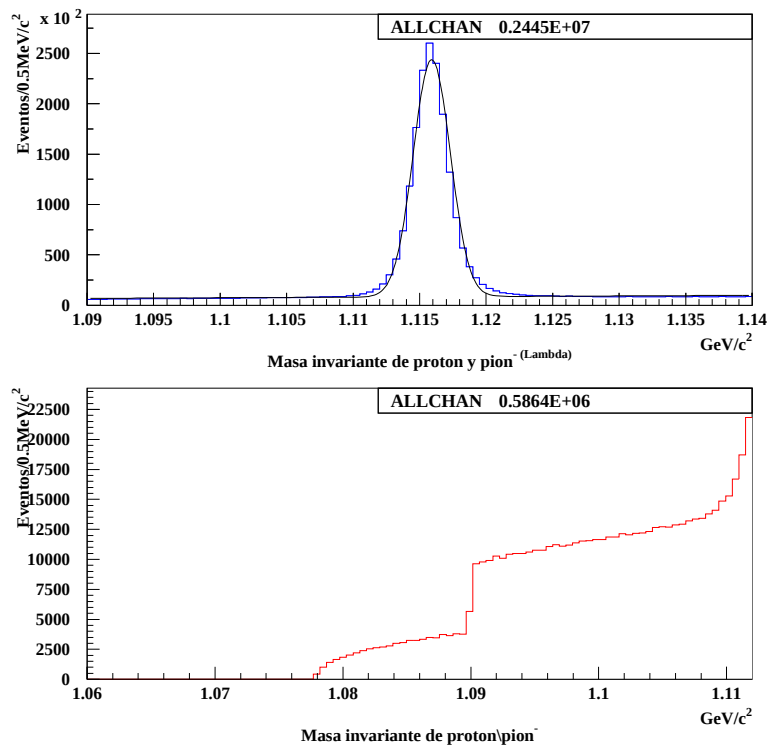


Figura 2.13: Masa invariante de p y π^- (Λ^0) y $p\pi^-$, con todos los paquetes de datos incluidos el grupo pp

Estos nuevos datos fueron analizados con ayuda de un programa *anal.F*, el cual es diferente al primero que se presentó y nuevamente fue desarrollado por la autora de este trabajo de tesis. Para la partícula $\Lambda^0 p \pi^-$ se volvieron a realizar los cortes y análisis ya explicados, sólo se agregó un nuevo corte sobre χ^2 para el vértice secundario. Se puede ver la distribución de χ^2 en la figura 2.14. El corte fué $\chi^2 < 5$.

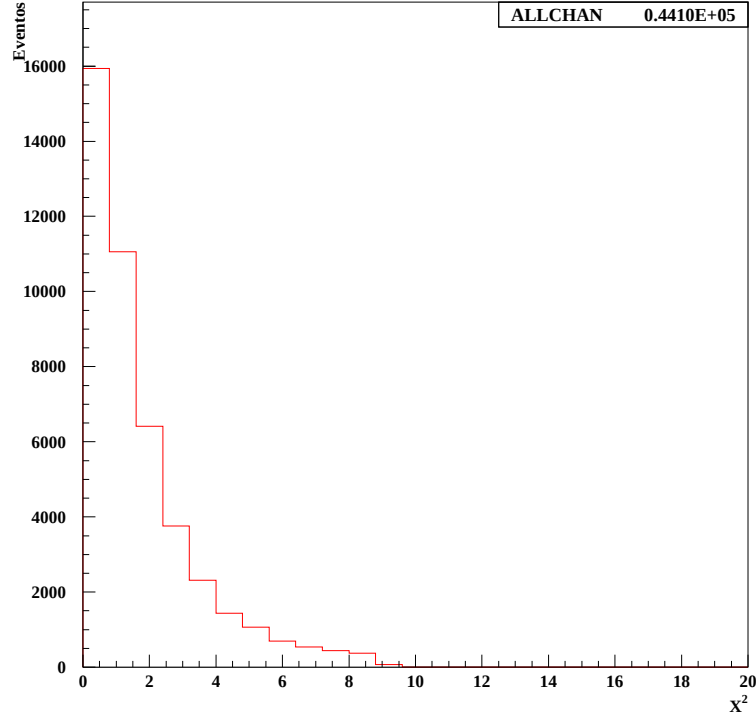


Figura 2.14: Distribucion de χ^2 para el vértice secundario

También se volvieron a utilizar los cortes L/σ . En esta ocasión es un corte sobre la partícula $\Lambda^0 p \pi$ y no sólo sobre la partícula hija $p \pi^-$ ya que al hacer el segundo refit ahora se toma en cuenta también la trayectoria de la partícula hija Λ . El valor de L cambiará muy poco pues sólo es la posición sobre el eje z , pero el valor de σ debe ser menor, el error debe ser menor.

Se muestran algunas de las gráficas con todos estos cortes en la figura 2.15.

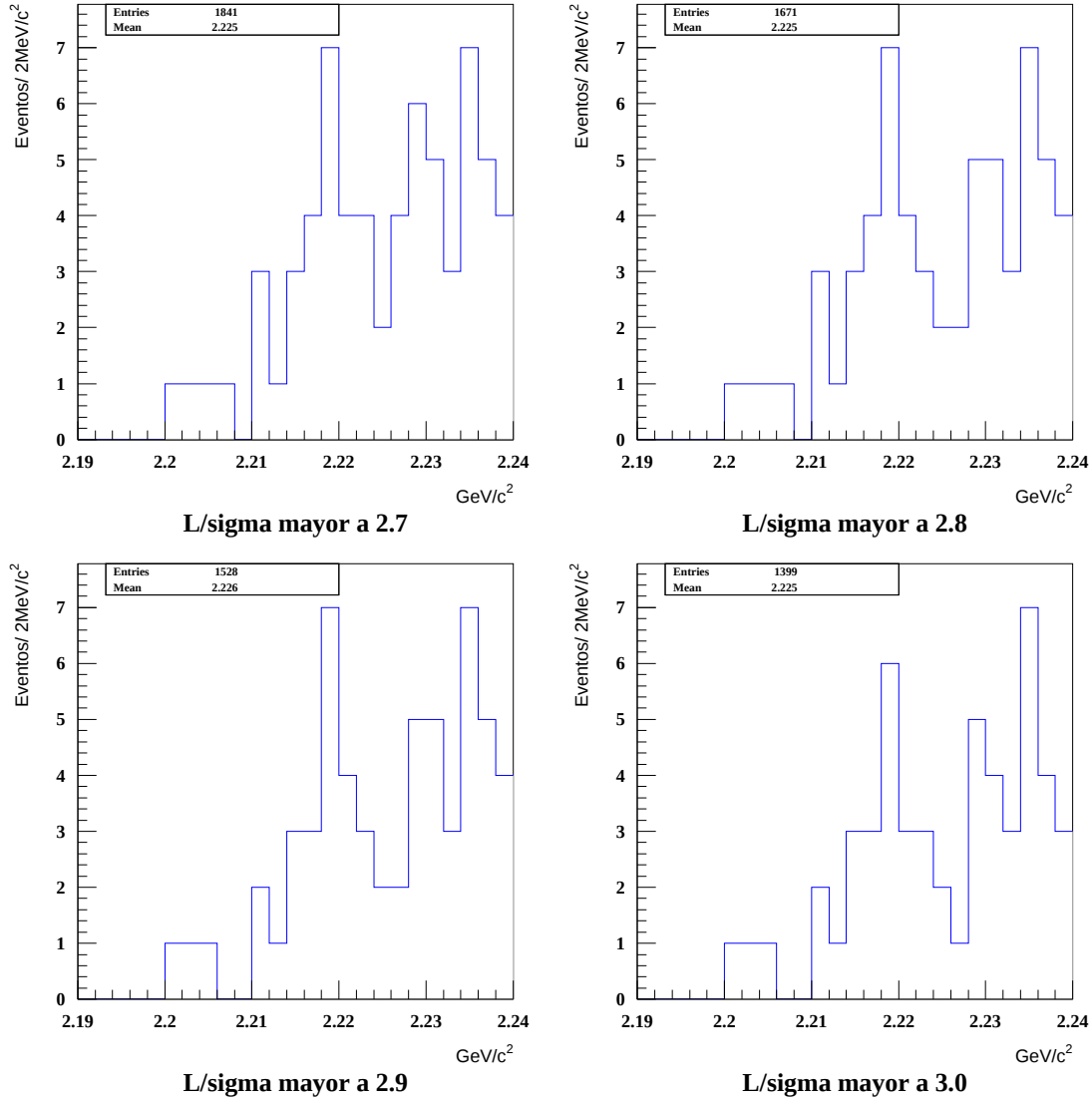


Figura 2.15: Masa invariante $\Lambda^0 p \pi^-$ con cortes en L/σ para $\Lambda^0 p \pi^-$ de 2.7 a 3

Resumiendo. Se encontraron dos acumulaciones importantes, en la figura 2.16; primero se ve una acumulación alrededor del valor de masa invariante de $2340 \text{ MeV}/c^2$, esta acumulación representa a una partícula con un decaimiento en dos partículas Λ^0 . En la figura 2.17 se ve una acumulación alrededor del valor de masa invariante de $2215 \text{ MeV}/c^2$, que representa una partícula con un decaimiento en $\Lambda^0 p \pi^0$. Es la primera vez que se observan estas acumulaciones.

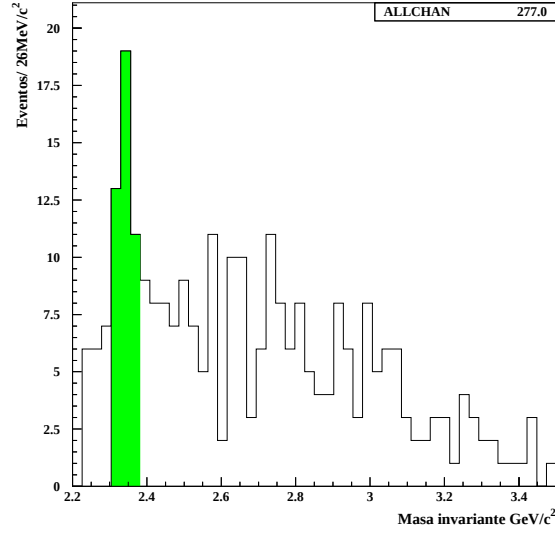


Figura 2.16: Masa invariante de dos partículas Λ^0 en el mismo evento

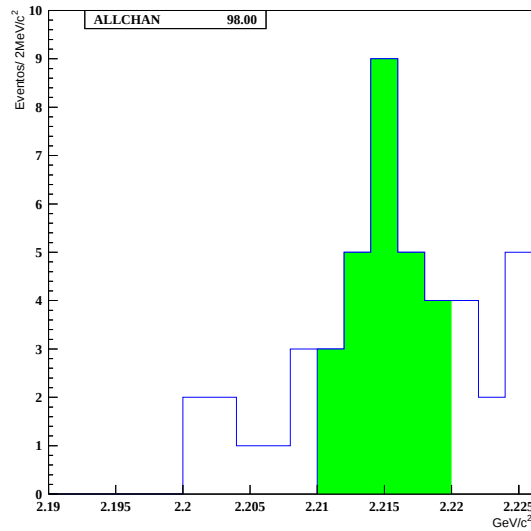


Figura 2.17: Corte en L/σ para $p\pi^-$ de 1.9

Capítulo 3

Estudios Sistemáticos

En el capítulo anterior, en la gráfica de la masa invariante de $\Lambda^0 p \pi^-$ se encontró un “pico” relevante al realizar el corte $L/\sigma > 1.9$ para $p \pi^-$ (Figura 2.11). Para demostrar que esta acumulación no es una manifestación de ruido o algún otro error, se hicieron varios estudios sistemáticos.

3.1. Comprobación mediante simulación

La simulación se realiza porque al tener varios picos similares en el histograma se desconoce cual es el pico que representa a la partícula que se está buscando ó el ancho de este pico¹. Se simuló el decaimiento $\Lambda^0 p \pi^-$ con embedding (ver 1.12), usando en la fórmula para $n = 4.3$ y para $b = 1/\frac{GeV}{c^2}$, que son los valores típicos en la reconstrucción de partículas. Se reconstruyó por el método de masa invariante y se obtuvo un “pico” que se obtendría si esta partícula existiera, se realizó un ajuste y se obtuvo un ancho de 2.9 MeV (Figura 3.1, parámetro P3), lo que justifica la selección del ancho de 2 MeV para el decaimiento $\Lambda^0 p \pi^-$ (Figura 2.11); ya que el ancho coincide con la resolución del experimento, se afirma que el experimento SELEX es adecuado para hacer estas mediciones.

¹El ancho es igual a la resolución

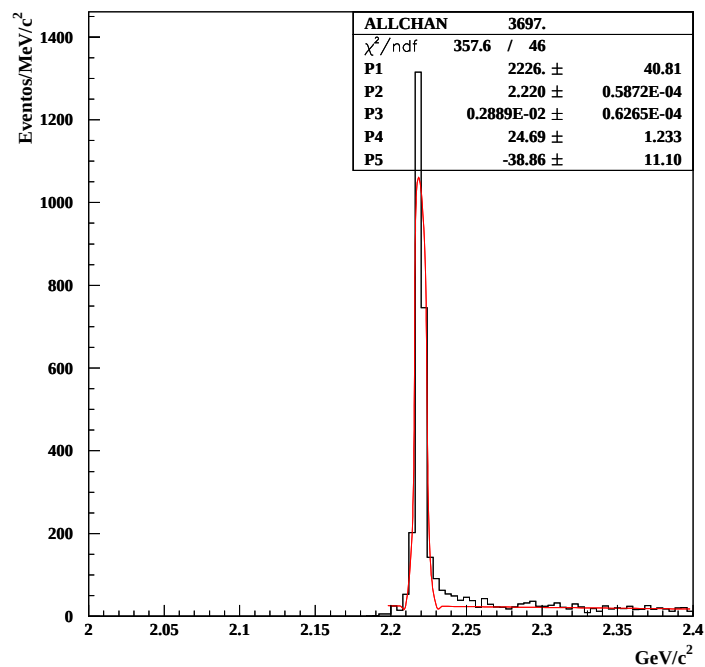


Figura 3.1: Masa invariante del decaimiento $H^0 \rightarrow \Lambda^0 p \pi^-$ encontrada mediante simulación

3.2. Estabilidad del pico contra diferentes cortes de L/σ

Se realizó el ajuste de una función gaussiana a las diferentes gráficas obtenidas (Figuras 2.9 y 2.10) y se observó el cambio en las variables del ajuste mientras cambia la variable L/σ para $p\pi^-$.

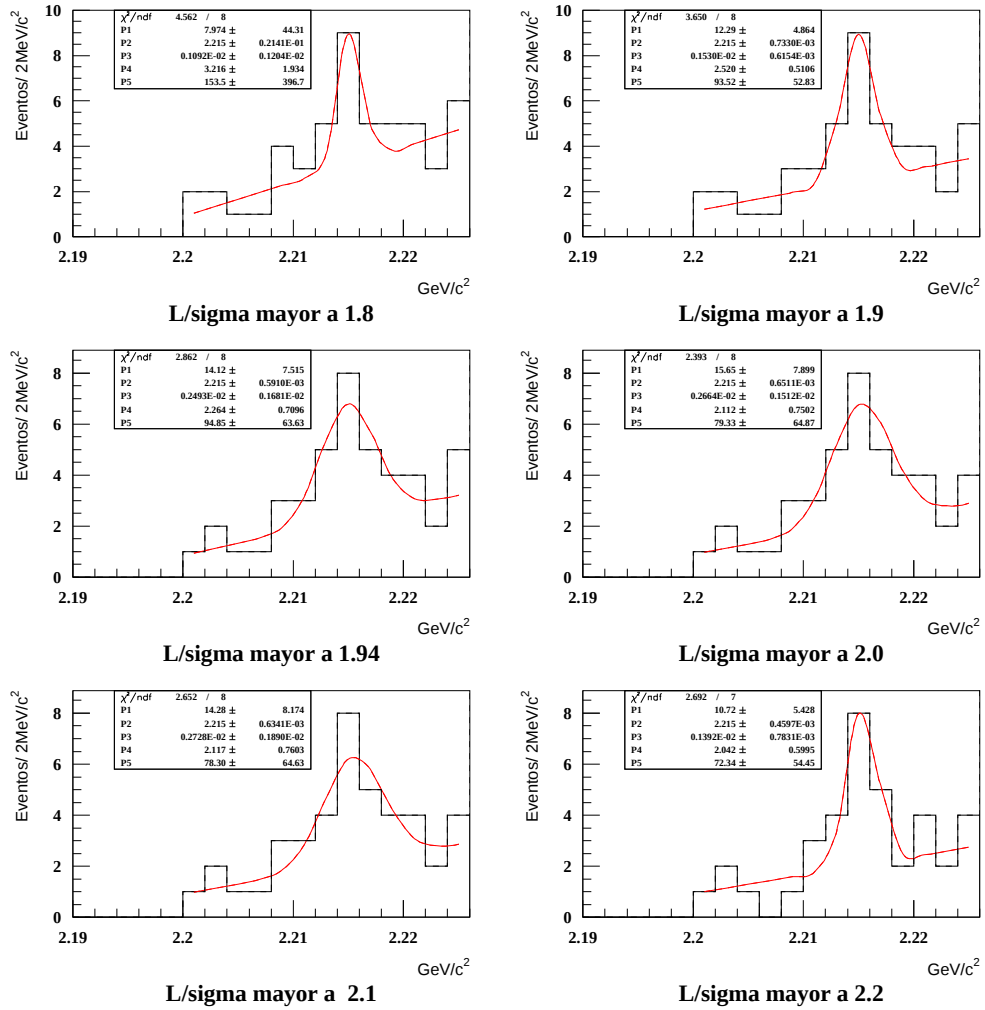


Figura 3.2: Ajuste de una función gaussiana a gráficas del decaimiento $H^0 \rightarrow \Lambda^0 p \pi^-$ con cortes en L/σ de 1.8 a 2.2 para $p\pi^-$

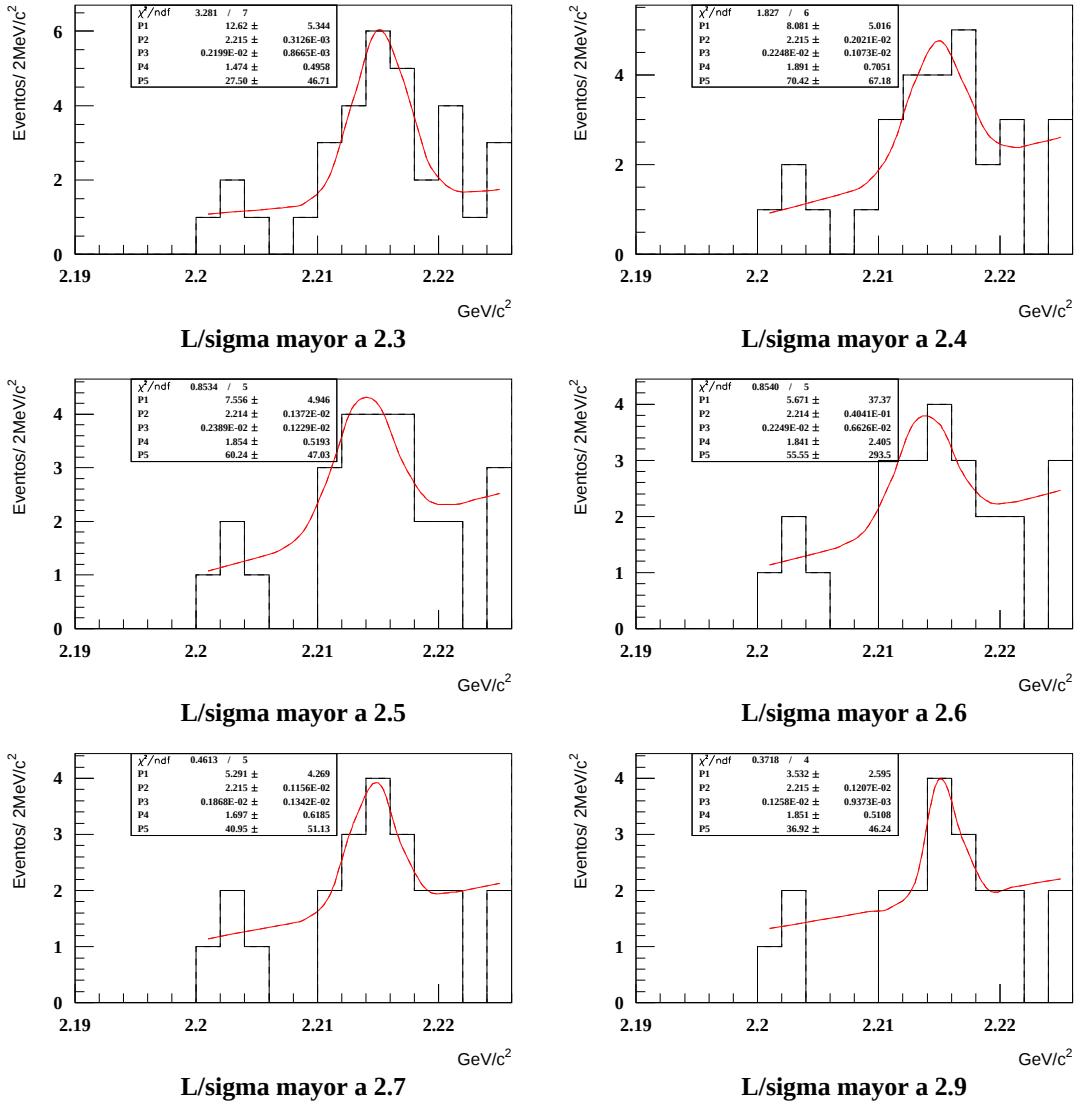


Figura 3.3: Ajuste de una función gaussiana a gráficas del decaimiento $H^0 \rightarrow \Lambda^0 p \pi^-$ con cortes en L/σ de 2.3 a 2.9 para $p\pi^-$

Cada parámetro de los ajustes (el número de partículas, la media, σ , la constante y la pendiente), fue graficado y se verificó que el valor de estas variables se mantienen constantes con los diferentes cortes en L/σ (Figura 3.4).

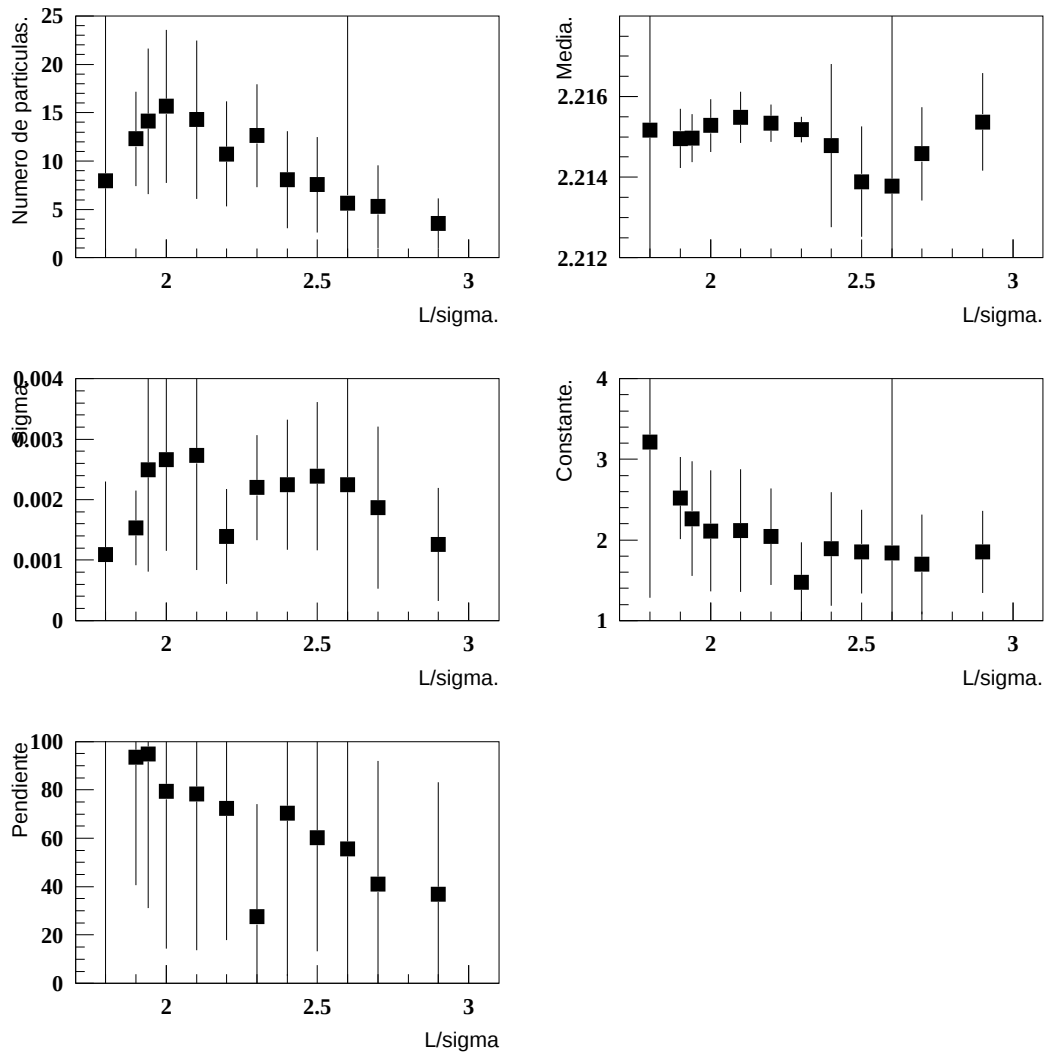


Figura 3.4: Gráficas de los parámetros del ajuste de una Gaussiana conforme cambia L/σ .

Al ver que los valores no presentan cambios abruptos, se mantienen más o menos constante, podemos decir que el pico es real.

3.3. Revisión de eventos repetidos

Hay ocasiones en que una partícula al tener poco momento (como un π^-), provoca una difracción múltiple, es decir, camina en zig-zag, este tipo de trayectoria tendrá errores de parámetros de ajuste muy grandes cuando se le ajuste una recta. Si se junta el resultado del ajuste con los errores se obtiene un cilindro muy ancho, y esto tiene como consecuencia que cualquier ajuste de vértices que se realice en éste, funcionará. Si se tienen 10 o más de estas partículas con poco momento, cuando se realice el método de la masa invariante, los valores obtenidos para cada partícula serán casi los mismos y formarán un “pico” muy claro.

Para comprobar que la acumulación encontrada en este trabajo no fuera como el caso descrito, se guardó una lista con los datos: número de corrida, número de evento, masa (se obtiene de SOAP) y masa invariante; los datos se tomaron del histograma que se generó después de todos los cálculos y cortes y se comprobó que las entradas son de eventos diferentes, lo que descartó el caso visto anteriormente. La lista tiene 3461918 entradas con 98 líneas. Se muestra sólo una parte de la lista a continuación:

5788.	1202714.	2.2253747	2.22625732
6063.	604585.	2.23559666	2.23070765
6148.	1228403.	2.20272756	2.2036202
6170.	906388.	2.2128315	2.21283627
6170.	1210637.	2.23754334	2.23296762
6418.	1152901.	2.23452878	2.23328543
6539.	903872.	2.22838426	2.22678924
6560.	403959.	2.22680163	2.22774911
6713.	604587.	2.23939824	2.23882151
6802.	1144372.	2.23804021	2.23830128

3.4. Estudio del ruido con event-mixing

Una prueba importante en este estudio es la descripción del ruido, ya que al tener pocos eventos que analizar, la estadística que se obtiene del decaimiento no es suficiente, provocando que el ruido pueda ser producto de algún error en el análisis. Para poder definir al ruido se utilizó el método *event-mixing*.

El método *event-mixing* se basa en afirmar que el ruido está formado por accidentes, es decir, partículas que no comparten el mismo vértice pero que aparecen en el histograma ó de partículas que no pertenecen al decaimiento buscado. Básicamente consiste en ignorar los datos que forman

la acumulación y concentrarse en los accidentes. Con este método se evitan dos problemas, uno es evitar la búsqueda de una función o modelo matemático que nos permita hacer un buen ajuste y el segundo es el quitar la incógnita de como se comporta el ruido adentro del “pico”, porque como no habrá “pico” se podrá ver el comportamiento del ruido en todo el rango de datos.

Por lo que primero se corre el programa SOAP y después se selecciona un evento con candidatos a $p\pi^-$ y de otro evento candidatos a Λ^0 que no tengan nada que ver uno con el otro, es decir, no deben pertenecer al decaimiento buscado $\Lambda^0 p\pi^-$ de cumplir esta condición, se guarda cuando más un candidato por evento.

Se utiliza una tabla parecida a la de embedding para reconstruir el decaimiento $\Lambda^0 p\pi^-$, pero utilizando la información de los $p\pi^-$ y Λ^0 que se seleccionaron. La diferencia entre mixing y embedding, es que mixing utiliza partículas reales, no simulaciones de estas, por lo que se esperan resultados reales.

Se corren los datos 25 veces para aumentar la cantidad de eventos para realizar un análisis estadístico más acertado, lo que da una precisión del ruido cinco veces mejor, es decir, el efecto del ruido al error total es de $1/25 \sim 4\%$, siempre y cuando el ruido sea igual a la señal. Los resultados se guardan en un histograma. Como estas partículas no pueden producir el decaimiento $\Lambda^0 p\pi^-$, producen ruido, es decir, el ruido será formado sólo por puros accidentes [3].

Estos nuevos datos son analizados con el programa *anal.F*, pero para que revise sólo los datos del ruido se debe agregar al principio del programa la siguiente línea, donde primero se genera la lista de los eventos que conforman el pico y luego se pide que no se tomen en cuenta y así tener la seguridad de que se esta analizando sólo datos que pertenecen al ruido:

```

Open(11,File='events.list')
  nb = 0
111 Continue
  nb = nb + 1
  Read(11,*,end=112) brun(nb),bevent(nb)
c    Write(*,*) nb,brun(nb),bevent(nb)
  Goto 111
112 Continue
  Close(11)
  nb = 0

  ievent = event
  irun = run

Do ib=1,nb
  if (ievent.eq.bevent(ib) .and. irun.eq.brun(ib)) Goto 10
EndDO

```

Con lo anterior tenemos dos tipos de datos: los creados con *event-mixing*, llamados **mix**, y los datos que se obtuvieron después de correr todos los paquetes de datos, incluyendo el paquete **pp** llamados **originales**.

Para el siguiente análisis se tomó la siguiente información de los histogramas:

- **ENTRIES**: El número de entradas, es la suma de todos los eventos que entran en el rango de masa invariante que se definió en el programa *anal.F* y los que no.
- **SUM** ó **ALLCHAN**: son sólo los eventos que entran en el rango de masa invariante que se definió en el programa *anal.F*.

Para realizar un estudio sistemático se llevó acabo un cálculo utilizando los datos **mix** y **originales**, para tres rangos diferentes de masa (2.19 a 2.24 GeV/c², 2.19 a 2.29 GeV/c² y de 2.19 a 4 GeV/c²), para cada corte diferente de L/σ (ningún corte, 0, 1, 2, 3 y 4) y para cada uno de los tipos de información obtenidos de los histogramas (**ENTRIES** y **SUM**).

Se ejemplifica un cálculo realizado con el rango de masa de 2.19 a 2.24 GeV/c², con corte de L/σ=0 con información de **ENTRIES**, usando los dos tipos de datos (**mix** y **originales**).

Primero se buscó el error con la fórmula 3.1.

$$error\ relativo = \frac{valor\ medido - 25 * valor\ real}{valor\ real} \quad (3.1)$$

Donde *valor medido* son los datos **mix**; y *valor real* son los datos **originales**. Se puede ver el programa *sc.kumac* con el que se realizó el cálculo en el apéndice D.

Con este número calculado, se utiliza su inverso para escalar un vector que toma la información de los histogramas generados con *event-mixing*. Se hace este cálculo por cada rango de masa, y corte en L/σ, y por tipo de información, para finalmente combinar los histogramas de los datos originales y mix (Figuras 3.5, 3.6, 3.7, 3.8, 3.9), y ver como se comporta el ruido, el ruido aparece de color rojo en las gráficas. Todo esto se realizó con otro programa llamado *mix.kumac* que se puede ver en el apéndice E.

Por lo tanto después de realizar el *event-mixing* se tiene bien descrito el ruido en todas las combinaciones de masa y rangos de ésta, y después de combinar estos resultados podemos ver el “*pico*” es real.

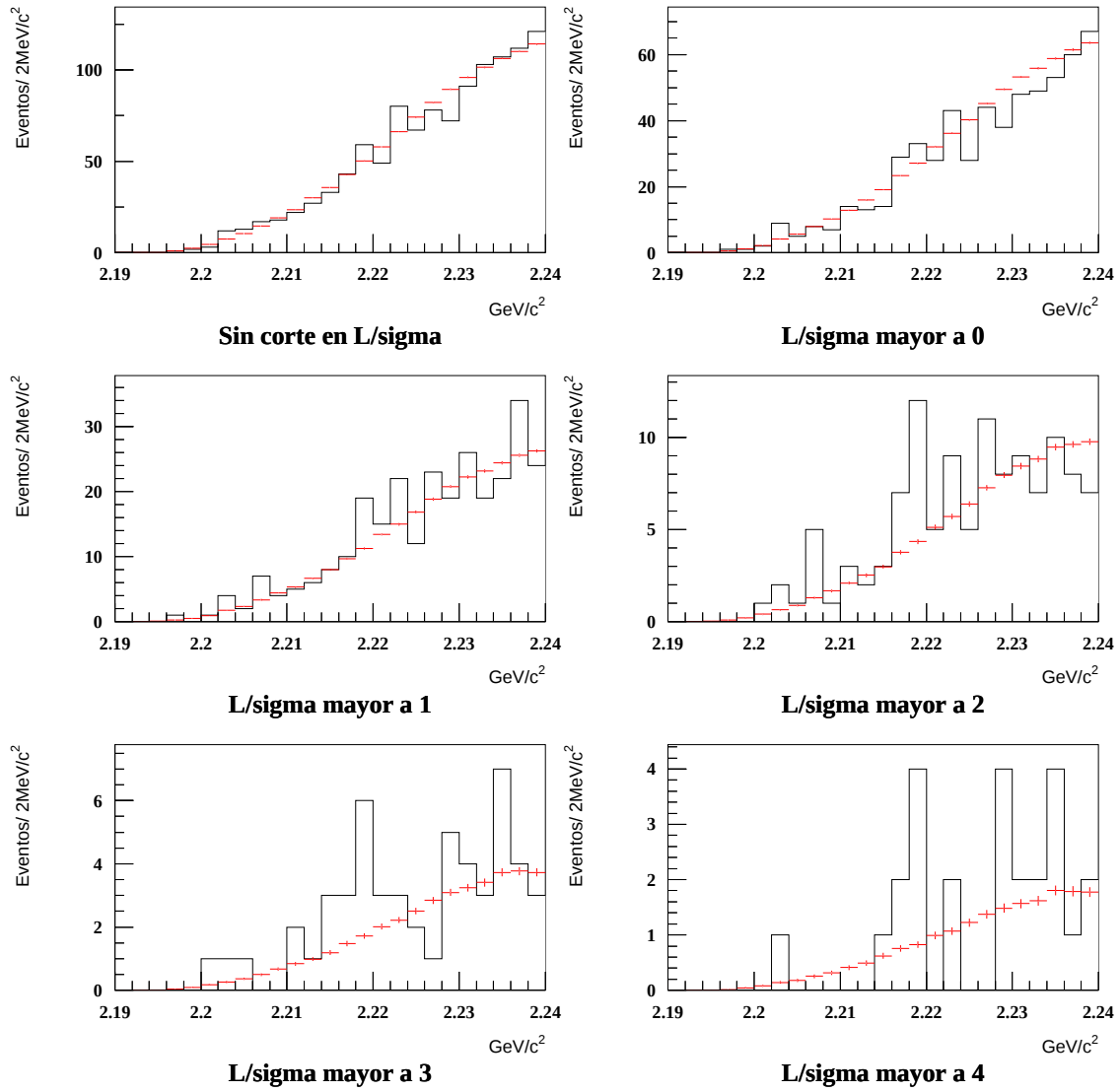


Figura 3.5: Entradas tomadas de los datos mix (rojo) y originales. Rango de masa desde 2.19 a 2.24 GeV/c^2

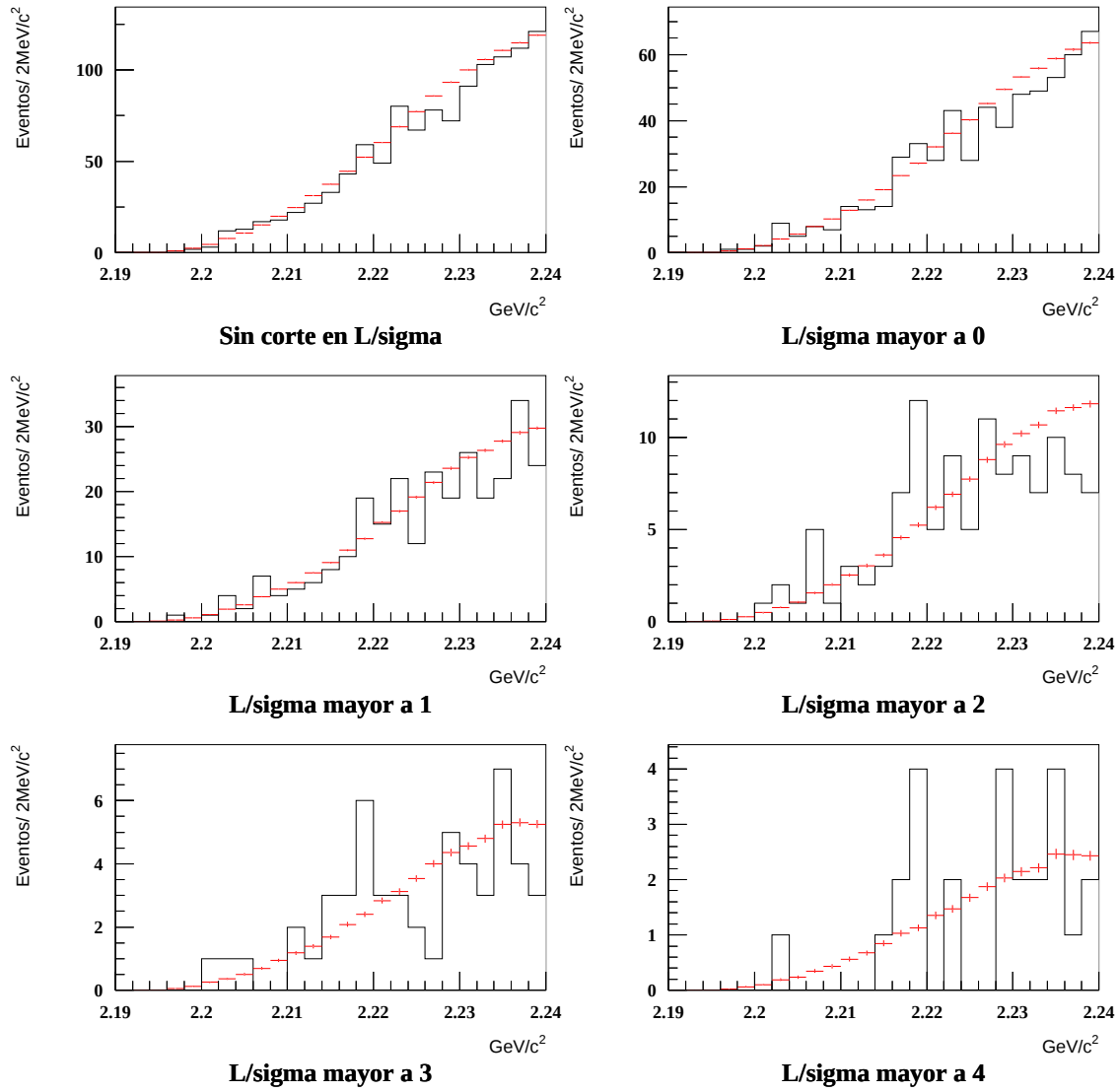


Figura 3.6: Contenidos de los histogramas (SUM) tomados de los datos mix (rojo) y originales. Rango de masa desde 2.19 a 2.24 GeV/c²

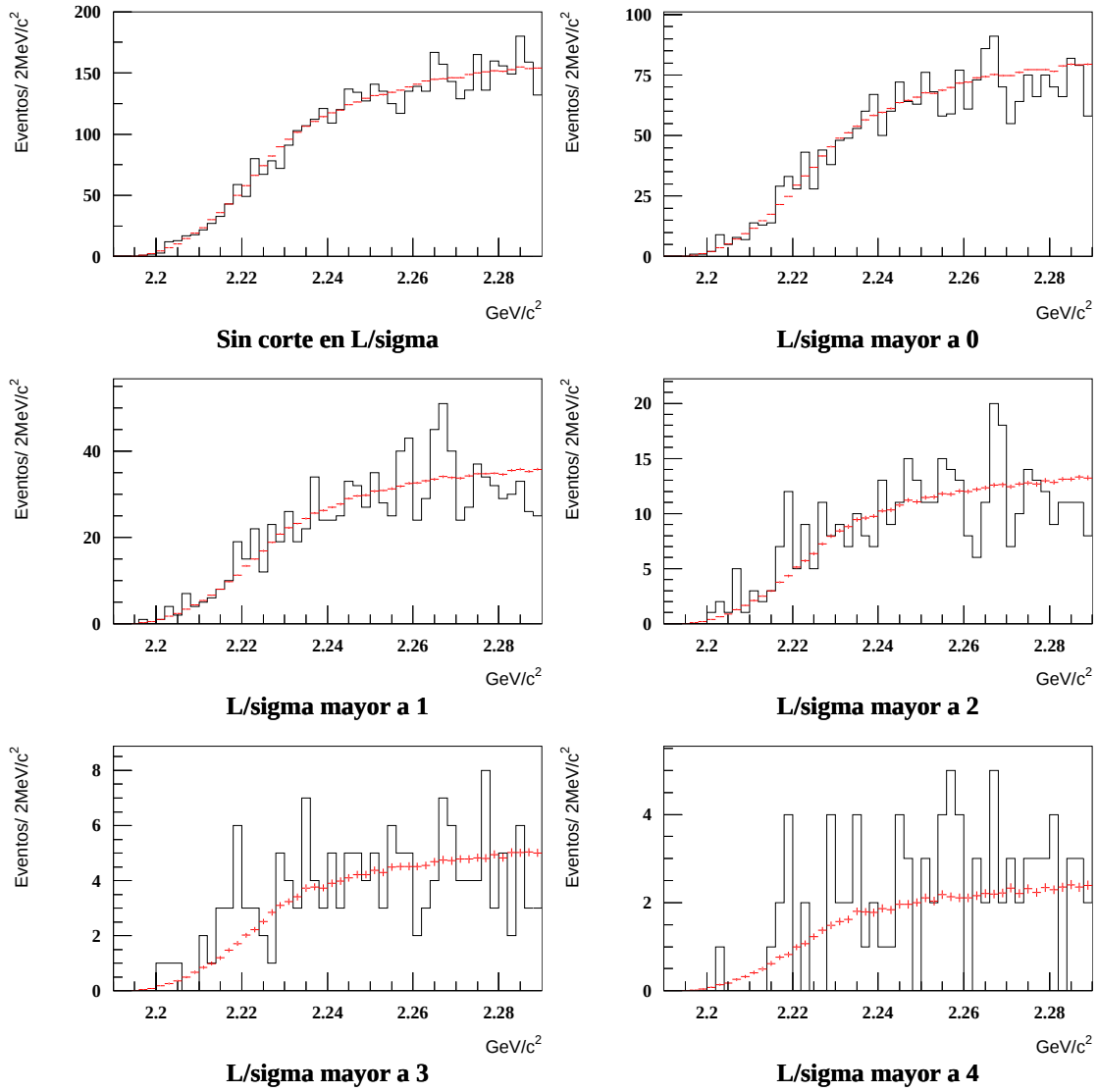


Figura 3.7: Entradas tomadas de los datos mix (rojo) y originales. Rango de masa desde 2.19 a 2.29 GeV/c^2

n

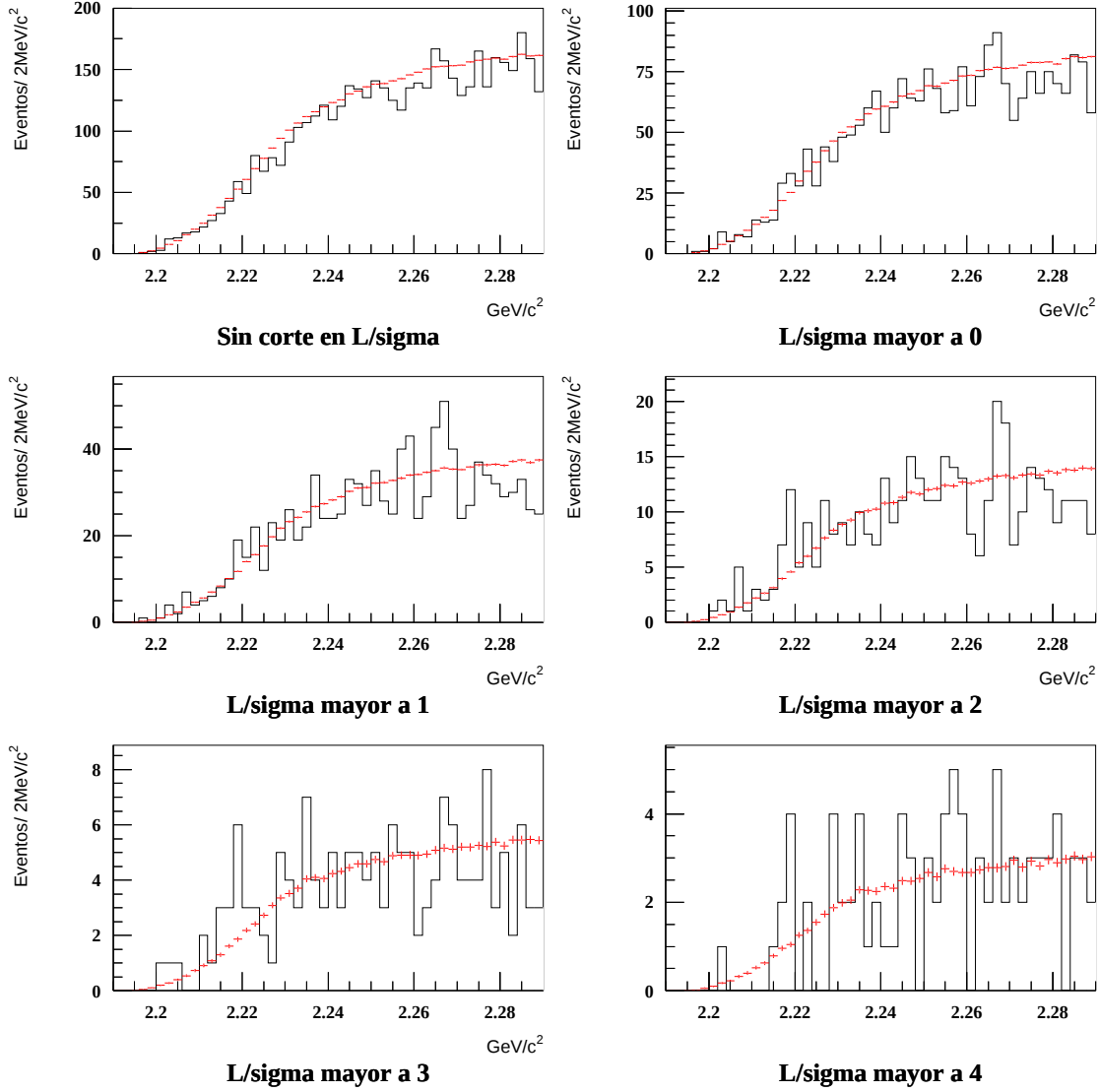


Figura 3.8: Contenidos de los histogramas (SUM) tomados de los datos mix (rojo) y originales. Rango de masa desde 2.19 a 2.29 GeV/c²

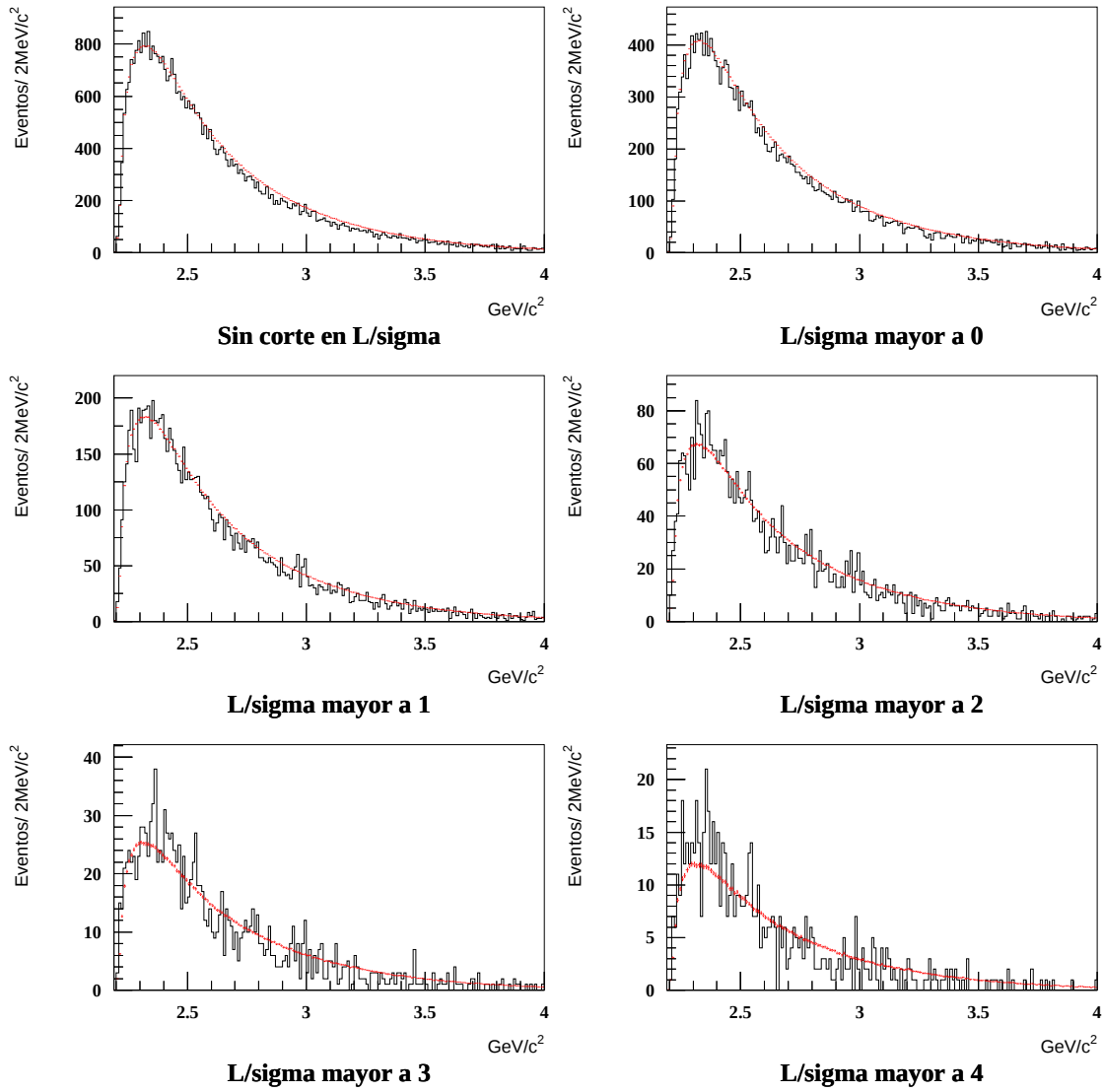


Figura 3.9: Entradas tomadas de los datos mix (rojo) y originales. Rango de masa desde 2.19 a 4 GeV/c^2

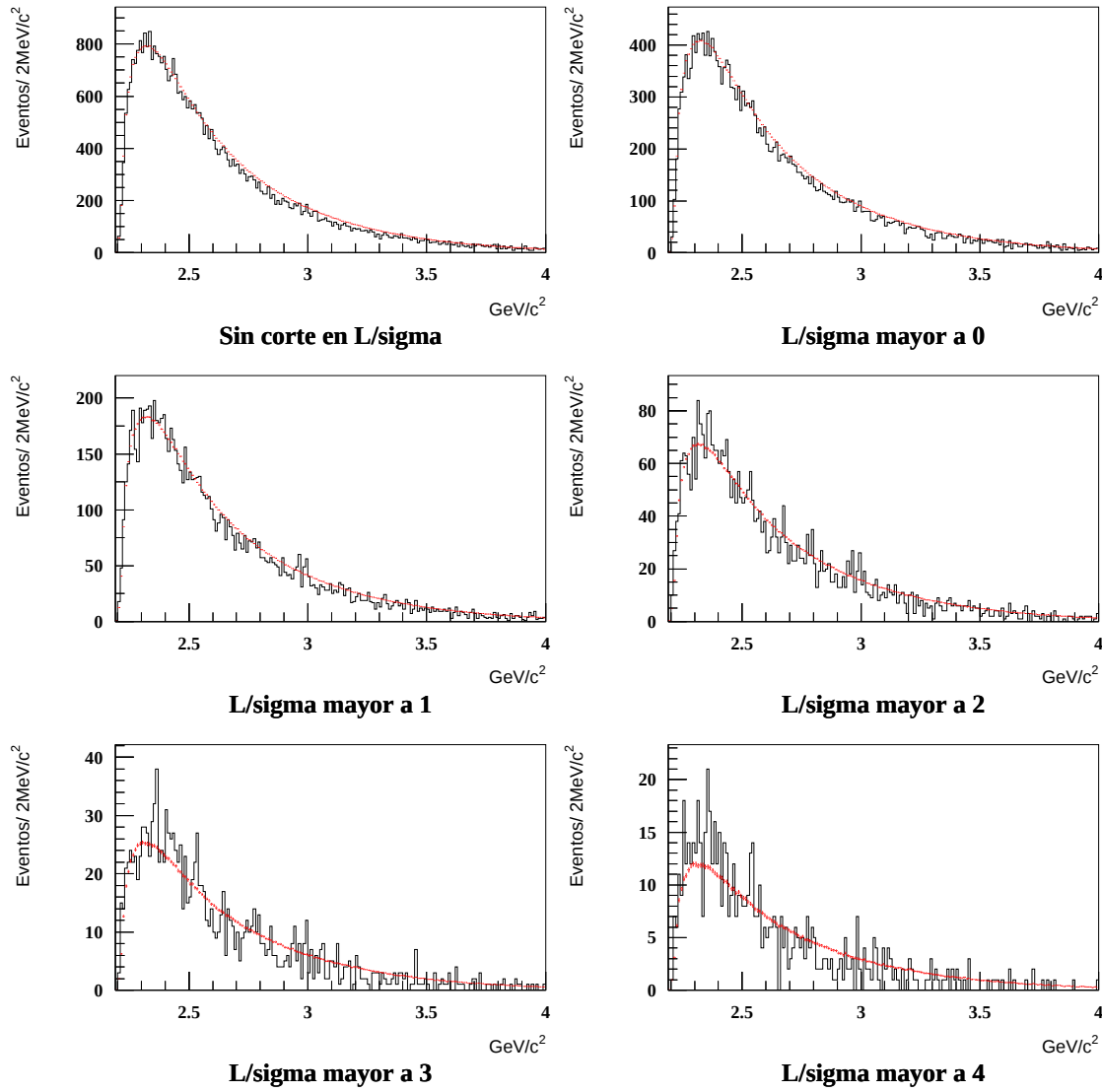


Figura 3.10: Contenidos de los histogramas (SUM) tomados de los datos mix (rojo) y originales. Rango de masa desde 2.19 a 4 GeV/c²

Capítulo 4

Conclusiones

En el presente trabajo de tesis se buscó partículas hadrónicas que decaen a su estado final en $\Lambda^0\Lambda^0$ y $\Lambda^0p\pi^-$, donde se encontraron dos acumulaciones para cada decaimiento:

Para $\Lambda^0\Lambda^0$ se buscó eventos que presentaran dos partículas Λ^0 donde se encontró una acumulación en la gráfica de la masa invariante con un valor de $2340 \text{ MeV}/c^2$ y un ancho de $14 \pm 0.001973 \text{ MeV}$; es decir, se encontró una partícula que decae en $\Lambda^0\Lambda^0$.

Para $\Lambda^0p\pi^-$ se encontró una acumulación en la gráfica de la masa invariante con un valor de $2215 \text{ MeV}/c^2$ y un ancho de $1.9 \pm 0.001262 \text{ MeV}$; es decir, se encontró una partícula que decae en $\Lambda^0p\pi^-$.

Se verificó que estas acumulaciones fueran reales y no algún error de análisis.

Estos resultados coinciden con lo que propuso Jaffe en 1977 sobre una nueva partícula que llamó H^0 formada de seis quarks ($uussdd$) y tiene una masa de $80 \text{ MeV}/c^2$ por debajo de $2m_\Lambda$, es decir, una masa alrededor de $2230 \text{ MeV}/c^2$. También predijo un estado excitado con una masa de $2335 \text{ MeV}/c^2$.

Con esto podemos decir que se encontró una nueva partícula con masa de $2215 \text{ MeV}/c^2$ y su estado excitado con masa de $2340 \text{ MeV}/c^2$, que coinciden aproximadamente con el modelo de Jaffe.

Como trabajo a futuro queda corregir algunos de los paquetes, para poder tener una cantidad más grande de datos que analizar y realizar una mejor reconstrucción de las partículas, como Λ^0 , para obtener una estadística más confiable de estos decaimientos.

Apéndice A

Colaboración SELEX

E. Gülmez

Bogazici University, Bebek 80815 Istanbul, Turkey

R. Edelstein, S.Y. Jun, A.I. Kulyavtsev¹, A. Kushnirenko², D. Mao³, P. Mathew⁴, M. Mattson,
M. Procario⁵, J. Russ, J. You¹

Carnegie-Mellon University, Pittsburgh, PA 15213, U.S.A.

A.M.F. Endler

Centro Brasileiro de Pesquisas Físicas, Rio de Janeiro, Brazil

P.S. Cooper, J. Kilmer, J. Lach, L. Stutte

Fermi National Accelerator Laboratory, Batavia, IL 60510, U.S.A.

V.P. Kubarovsky, V.F. Kurshetsov, A.P. Kozhevnikov, L.G. Landsberg⁶, V.V. Molchanov, S.B. Nurusev,
S.V. Petrenko, A.N. Vasiliev, D.V. Vavilov, V.A. Victorov

Institute for High Energy Physics, Protvino, Russia

M.Y. Balatz⁶, G.V. Davidenko, A.G. Dolgolenko, G.B. Dzyubenko⁶, A.V. Evdokimov, M.A. Kubantsev,
I. Larin, V. Matveev, A.P. Nilov⁶, V.A. Prutsko, A.I. Sitnikov, V.S. Verebryusov⁶, V.E. Vishnyakov

Institute of Theoretical and Experimental Physics, Moscow, Russia

U. Dersch⁷, I. Eschrich⁸, I. Konorov⁹, H. Krüger¹⁰, J. Simon¹¹, K. Vorwalter¹²

Max-Planck-Institut für Kernphysik, 69117 Heidelberg, Germany

I.S. Filimonov⁶, E.M. Leikin, A.V. Nemitkin, V.I. Rud
Moscow State University, Moscow, Russia

A.G. Atamantchouk⁶, G. Alkhazov, N.F. Bondar, V.L. Golovtsov, V.T. Kim, L.M. Kochenda,
A.G. Krivshich, N.P. Kuropatkin¹, V.P. Maleev, P.V. Neoustroev, B.V. Razmyslovich¹³, V. Stepanov¹³,
M. Svoiski¹³, N.K. Terentyev¹⁴, L.N. Uvarov, A.A. Vorobyov
Petersburg Nuclear Physics Institute, St. Petersburg, Russia

M.A. Moinester, A. Ocherashvili¹⁵, V. Steiner
Tel Aviv University, 69978 Ramat Aviv, Israel

J. Amaro-Reyes, A. Blanco-Covarrubias, J. Engelfried, N. Estrada, A. Flores-Castillo, G. López-Hinojosa,
A. Morelos, J.L. Sánchez-López, I. Torres¹⁶, E. Vázquez-Jáuregui¹⁷
Universidad Autónoma de San Luis Potosí, San Luis Potosí, México

V.J. Smith
University of Bristol, Bristol BS8 1TL, United Kingdom

U. Akgun, A.S. Ayan, M. Kaya¹⁸, E. McCliment, C. Newsom, Y. Onel, S. Ozkorucuklu¹⁹
University of Iowa, Iowa City, IA 52242, U.S.A.

L.J. Dauwe⁶
University of Michigan-Flint, Flint, MI 48502, U.S.A.

M. Iori
University of Rome “La Sapienza” and INFN, Rome, Italy

L. Emediato, C.O. Escobar²⁰, F.G. Garcia¹, P. Gouffon, T. Lungov, M. Srivastava, R. Zukanovich-Funchal
University of São Paulo, São Paulo, Brazil

A. Penzo
University of Trieste and INFN, Trieste, Italy

-
- ¹Now at Fermi National Accelerator Laboratory, Batavia, IL 60510, U.S.A.
- ²Now at Institute for High Energy Physics, Protvino, Russia
- ³Present address: Lucent Technologies, Naperville, IL
- ⁴Present address: Baxter Healthcare, Round Lake IL
- ⁵Present address: DOE, Germantown, MD
- ⁶deceased
- ⁷Present address: Advanced Mask Technology Center, Dresden, Germany
- ⁸Present address: University of California at Irvine, Irvine, CA 92697, USA
- ⁹Present address: Physik-Department, Technische Universität München, 85748 Garching, Germany
- ¹⁰Present address: The Boston Consulting Group, München, Germany
- ¹¹Present address: Siemens Healthcare, Erlangen, Germany
- ¹²Present address: Allianz Insurance Group IT, München, Germany
- ¹³Present address: Solidum, Ottawa, Ontario, Canada
- ¹⁴Now at Carnegie-Mellon University, Pittsburgh, PA 15213, U.S.A.
- ¹⁵Present address: NRCN, 84190 Beer-Sheva, Israel
- ¹⁶Present address: Instituto Nacional de Astrofísica, Óptica y Electrónica, Tonantzintla, Mexico
- ¹⁷Present address: SNOLAB
- ¹⁸Present address: Kafkas University, Kars, Turkey
- ¹⁹Present address: Süleyman Demirel Üniversitesi, Isparta, Turkey
- ²⁰Present address: Instituto de Física da Universidade Estadual de Campinas, UNICAMP, SP, Brazil

Apéndice B

Programa *recon_recon.F*

A continuación se presenta el programa mejorado de *recon_recon.F*.

```
subroutine recon_recon(irecdf)

*      $Id: recon_recon.F,v 1.34 2010/08/04 14:32:36 jurgen Exp $
*      $Author: jurgen $

*      Search for all instances in an event which match the several
*      already done reconstructions
*      defined in the list of reconstruction definitions (recdf).

*      from:pid = rr:recn1recn2 - add two reconstructions.
*      examples:

*#recdf      0  0  fill anal v01.0  25-Jul-1997 08:42 psc ! track recon
*      examples
*name from pr  q  pid      ls_min ls_max mass_min mass_max pvtx_min pvtx_max
*      bit
*      rr  3 +1  i+d0_kpi   0.0    0.0    1.000    4.000        0.0      0.0  $8

*      Jurgen Engelfried      19 January 1998
*      V.Matveev May 1999
*      &      ntracks, tracks(mtrk)      ,! number, track list intern
*      store negative index in track list in case of gamma

implicit none

*      Include files

#include "control.inc"
```

```

#include "drv_lun.inc"
#include "part_cbk.inc"
#include "recdf_cbk.inc"
#include "recn_bk.inc"
#include "recon_par.inc"
#include "spec_pos_cbk.inc"
#include "trk_bk.inc"
#include "tseg_bk.inc"
#include "vtx2_bk.inc"
#include "vertex_par.inc"
#include "tgt_pos_cbk.inc"
#include "tgt_sz_cbk.inc"
#include "virt_trk.inc"

*      arguments

      Integer irecdf                                ! index in recon def

*      Local declarations

      logical
&      first , bad_pid , no_recn , already_used ,
&      do_refit_prim , do_refit_sec ,
&      pid_double(mrecdf)                        ! identical ids?

      character*12
&      from                                     ,! local "from" string
&      pid                                     ! local "pid" string
      real*8
&      p(4)                                   ,! 4vector for mass calc
*      mass_result                             ! calculated (fixed) mass
      Real
&      chi2_prim ,                             ! primary vertex chi2
&      recon_pvtx                             ,! point-back rchi2
&      chi2_sec                               ! secondary vertex chi2

      integer
&      blklen                                 ,! from util
&      n                                     ,! address of recn
&      nirecn , iirecn(mrecn)                 ,! for r s
&      ntracks , tracks(mtrk)                 ,! number, track list intern
&      pid_id(max_objs_recn , mrecdf) ,        ! pointer to id-recdf
&      pid_type(0:max_objs_recn , mrecdf) ,    ! is it vk1?
&      nr(mrecn) ,                            ! how many r
&      perm_array(mrecn) ,                    ! for permutations
&      npass                                 , ! for primary fit
&      nvtxp                                 , ! location for prim fit
&      nvtxs                                 , ! location for sec fit
&      ivtx                                  , ! first prim vertex
&      tgt                                   , ! target
&      ntrackssec , trackssec(mtrk)           , ! tracks for sec.

```

```

&      ntrackssec2,trackssec2(mtrk)           , !      vertex fit.
&      i, irecn, itrk,jtrk, jrecdf, r, itgt    ! loop index
Integer
&      vtracks(mtrk),                        ! for virtual tracks
&      nptracks
Integer jprim,jprim2,i_pvtx,i_pvtx2,jsec,jsec2,isegsec,
&      ndof,i_svtx,i_svtx2
Real xsec,ysec,zsec,zsec2,r_tmp
Real pxsect,pysect,pzsect,ttx,tty,pxsec,pysec,pzsec,tx,ty,
&      param,bx,by,z1,z2,w1,wt,zt,zp,sig_zp,chi2,zd,sigma

Save first, nr, pid_id, chi2_prim, npass, chi2_sec, pid_type
Save do_refit_prim,do_refit_sec, pid_double
Data first/.True./

integer XFF00, X100, X1F, X1000000
parameter (XFF00='FF00'x, X100='100'x, X1F='1F'X,
&          X1000000='1000000'x)

If (first) Then
  First = .False.
  call vzeroi(pid_id,max_objs_recn*mrecdf)

  If (cuts(c_rec_cpri,p_recon).le.0.) Then
    If (cuts(c_vtx1,p_vertex).gt.0.) Then
      chi2_prim = cuts(c_vtx1,p_vertex)
    Else
      chi2_prim = 4.
    EndIf
  Else
    chi2_prim = cuts(c_rec_cpri,p_recon)
  EndIf
  If (cuts(c_rec_csec,p_recon).le.0.) Then
    If (cuts(c_vtx2,p_vertex).gt.0.) Then
      chi2_sec = cuts(c_vtx2,p_vertex)
    Else
      chi2_sec = 5.
    EndIf
  Else
    chi2_sec = cuts(c_rec_csec,p_recon)
  EndIf
  If (cuts(c_rec_npass,p_recon).le.0.) Then
    If (cuts(c_npass2,p_vertex).gt.0.) Then
      npass = cuts(c_npass2,p_vertex)
    Else
      npass = 5
    EndIf
  Else
    npass = cuts(c_rec_npass,p_recon)
  EndIf
  Write(LUNSOUT,1101) chi2_prim,chi2_sec,npass

```

```

Write(LUNLOG,1101) chi2_prim,chi2_sec,npass
If (control(s_rec_refit2,p_recon)) Then
  do_refit_prim = .True.
  do_refit_sec = .True.
  Write(LUNSOUT,1103) ' ','_'
  Write(LUNLOG,1103) ' ','_'
ElseIf (control(s_rec_refit1,p_recon)) Then
  do_refit_prim = .True.
  do_refit_sec = .False.
  Write(LUNSOUT,1103) ' ','_not_'
  Write(LUNLOG,1103) ' ','_not_'
Else
  do_refit_prim = .False.
  do_refit_sec = .False.
  Write(LUNSOUT,1103) ' _not_', '_not_'
  Write(LUNLOG,1103) ' _not_', '_not_'
EndIf
EndIf

ivtx = 0
If (n_dir_vtx2(1,single_vtx2).le.1) Then
  ivtx=0 ! no primary vertex
Else
  If (n_dir_vtx2(1,single_vtx2).le.2) Then
    ivtx = beg_dir_vtx2(1,single_vtx2) ! first single vertex
  Else
    ivtx = beg_dir_vtx2(1,single_vtx2)+2 ! third single vertex
  Endif
Endif
If (ivtx.gt.0) Then
  If (nt_vtx2(1,ivtx).le.0) ivtx = 0 ! no tracks in primary
Else
  ivtx = 0
EndIf

* fill recn table - apply recdf cuts
c print *, 'RR: enter recon, irecdf, nrecn=' ,irecdf, nrecn
c write(*,*) pid_id(1,irecdf)

If (pid_id(1,irecdf).eq.0) Then ! do the compilation
  Bad_pid = .False.
  from = from_recdf(1,irecdf) ! local "from" string
  pid = pid_recdf(1,irecdf) ! local "pid" string
  nr(irecdf) = blklen(from_recdf(1,irecdf))
  If (nr(irecdf).lt.2) Bad_pid = .True.
  Write(LUNSOUT,1001) id_recdf(1,irecdf),nr(irecdf)
  Write(LUNLOG,1001) id_recdf(1,irecdf),nr(irecdf)
  Do r = 1,nr(irecdf) ! compile recons
    Do jrecdf = 1,irecdf-1 ! loop over all prev. defs
      If (pid(1:blklen(name_recdf(1,jrecdf)))
& .eq. name_recdf(1,jrecdf)

```

```

&          (1:blklen(name_recdf(1,jrecdf)))) Then
pid_id(r,irecdf) = id_recdf(1,jrecdf)
Write(LUNSOUT,1002) id_recdf(1,jrecdf)
Write(LUNLOG,1002) id_recdf(1,jrecdf)
If (from_recdf(1,jrecdf)(1:3).eq.'vk1') Then
    pid_type(r,irecdf) = 1
Else
    pid_type(r,irecdf) = 0
EndIf
pid = pid(blklen(name_recdf(1,jrecdf))+1:)
Goto 10
EndIf
EndDo
10 EndDo

Write(LUNSOUT,*) '┐'
Write(LUNLOG,*) '┐'
pid_type(0,irecdf) = 0
pid_double(irecdf) = .False.
Do r = 1,nr(irecdf)
    If (pid_type(r,irecdf).eq.1) pid_type(0,irecdf) = 1
    If (pid_id(r,irecdf).eq.0) Bad_pid = .True.
    If (r.ge.2) Then
        Do i = 1,r-1
            If (pid_id(r,irecdf).eq.pid_id(i,irecdf))
&                pid_double(irecdf) = .True.
        EndDo
    EndIf
EndDo
If (Bad_pid) Then
    Write(LUNSOUT,*) 'RECON.RECON:┐Bad┐definition┐',
&        pid_recdf(1,irecdf)
    Write(LUNLOG,*) 'RECON.RECON:┐Bad┐definition┐',
&        pid_recdf(1,irecdf)
    pid_id(1,irecdf) = -1 ! bad definition
EndIf
EndIf ! compilation

c    Write(*,*) 'pid_id',(pid_id(r,irecdf),r=1,nr(irecdf))
If (pid_id(1,irecdf).eq.-1) Goto 900 ! bad definition

nirecn = 0
c    Write(*,*) 'id_recon',nrecn,(id_recn(1,irecn),irecn=1,nrecn)
c    Write(*,*) (pid_id(r,irecdf),r=1,nr(irecdf)),pid_double(irecdf)
Do r = 1,nr(irecdf)
    No_recn = .True.
    Do irecn = 1,nrecn ! find matching recon
        If (id_recn(1,irecn).eq.pid_id(r,irecdf)) then ! matching recon
            already_used = .False.
        Do i = 1,nirecn
            If (iirecn(i).eq.irecn) already_used = .True.

```

```

c      Write(*,*) i,irecn,iirecn(i)
      EndDo
      If (.not.already_used) Then
        nirecn = nirecn + 1
        iirecn(nirecn) = irecn
c      Write(*,*) 'iirecn_',r,nirecn,(iirecn(i),i=1,nirecn)
        perm_array(nirecn) = nirecn
        No_recn = .False.
      EndIf
    EndIf
  EndDo

```

La mejora en el programa fueron las siguientes lineas:

```

c      If (No_recn .and. .not.pid_double(irecdf)) Then ! nothing found
c      If (No_recn .and. .not.pid_double(irecdf)
c      &      .and. nirecn.lt.nr(irecdf)) Then ! nothing found
c      If (No_recn) Then ! nothing found

      If (pid_double(irecdf)) Then
c      If (nirecn.lt.nr(irecdf)) Then ! nothing found
        Write(*,*) 'Did_not_find_',r,pid_id(r,irecdf),already_used
        Goto 900
      EndIf
      Else
c      If (No_recn) Then ! nothing found
        Write(*,*) 'Did_not_find_',r,pid_id(r,irecdf),already_used
        Goto 900
      EndIf
    EndIf
  EndDo

c      print *, 'found_matching_recons_#', nirecn
c      Write(*,*) 'Doing_Permus'
c      Write(*,*) (iirecn(irecn),irecn=1,nirecn)
      perm_array(1) = 0
      Call Combi(perm_array,nirecn,nr(irecdf))
      Do While (perm_array(1).ne.0) ! Loop over permutation
c      Write(*,*) (perm_array(r),r=1,nr(irecdf))

      ntracks = 0
      ntrackssec = 0
      ntrackssec2 = 0
      Do r = 1,nr(irecdf)
        If (id_recn(1,iirecn(perm_array(r))).ne.
&      pid_id(r,irecdf)) Goto 150 ! wrong permu
        Do itrk = 1,pr_recdf(1,recdf_recn(1,iirecn(perm_array(r))))
          ntracks = ntracks + 1
          tracks(ntracks) =
&      iand(pmtrk_recn(itrk,iirecn(perm_array(r))),

```

```

&          XFF00)/X100
      vtracks(ntracks) = tracks(ntracks)
      if(iand(pmtrk_recn(itrk,iirecn(perm_array(r))),
&          X1000000).ne.0)          ! gamma
&          tracks(ntracks) = -tracks(ntracks) ! make neg.index
ccc get tracks from first rr
      if (r.eq.1.and.pid_type(0,iirecn(perm_array(r))).eq.0) Then
        ntrackssec = ntrackssec +1      ! first secondary
        trackssec(ntrackssec) = tracks(ntracks)
      Endif
      if (r.eq.2.and.pid_type(0,iirecn(perm_array(r))).eq.0) Then
        ntrackssec2 = ntrackssec2 +1    ! second secondary
        trackssec2(ntrackssec2) = tracks(ntracks)
      endif
ccc end get tracks from...

      EndDo
EndDo

c      Write(*,*) 'RR: _trks:',(tracks(itrk),itrk=1,ntracks)

Do itrk = 1,ntracks-1
  Do jtrk = itrk+1,ntracks
    If (tracks(itrk).eq.tracks(jtrk)) Goto 150 ! same tracks
  EndDo
EndDo

If (nrecn.ge.mrecn-nr(irecdf)) Then ! Overflow
  overflow_recn = overflow_recn + 1 + nr(irecdf)
  Goto 900 ! we are done here
Else
  n = nrecn + 1 ! Get next recn entry
EndIf
call vzeror(recn(1,n),lrecn) ! clear entry

c      calculate invariant mass (later allow for fixed mass via partnumber)
      Do r = 1,nr(irecdf)
        p(4) = pz_recn(1,iirecn(perm_array(r)))
        p(3) = py_recn(1,iirecn(perm_array(r)))
        p(2) = px_recn(1,iirecn(perm_array(r)))
c      If (id_recn(1,iirecn(perm_array(r))).ne.12) Then

        mass_result = dble(mass_recn(1,iirecn(perm_array(r))))

c      ** code for recons containing ds vees/kinks (should use for vees only)
c      if (from_recdf(1,recdf_recn(1,iirecn(perm_array(r))))(1:3)
c      &      .eq.'vk1') then
c      If (pid_type(1,iirecn(perm_array(r))).eq.1) Then ! vk1
c      ** check charm cuts on hyperon (use l/sig area in recdf)
c      if(mass_recn(1,iirecn(perm_array(r))).gt.
&      ls_max_recdf(1,recdf_recn(1,iirecn(perm_array(r))))

```

```

&          )then
      goto 150          ! fail cut-do next combination
    else if (mass_recn(1,iirecn(perm_array(r))).lt.
&          ls_min_recdf(1,recdf_recn(1,iirecn(perm_array(r))))
&          )then
      goto 150          ! fail cut-do next combination
    else
    end if

c      ** load table particle mass instead of recn mass
      if (id_recn(1,iirecn(perm_array(r))).eq.20) Then
        mass_result = mass_part(1,kaon0_long_pid)
      elseif (id_recn(1,iirecn(perm_array(r))).eq.21 .or.
*         id_recn(1,iirecn(perm_array(r))).eq.22) then
        mass_result = mass_part(1,lambda_pid)
      elseif (id_recn(1,iirecn(perm_array(r))).eq.23 .or.
*         id_recn(1,iirecn(perm_array(r))).eq.24) then
        mass_result = mass_part(1,sigma_minus_pid)
      elseif (id_recn(1,iirecn(perm_array(r))).eq.25 .or.
*         id_recn(1,iirecn(perm_array(r))).eq.26) then
        mass_result = mass_part(1,sigma_plus_pid)
      elseif (id_recn(1,iirecn(perm_array(r))).eq.27 .or.
*         id_recn(1,iirecn(perm_array(r))).eq.28) then
        mass_result = mass_part(1,xi_minus_pid)
      elseif (id_recn(1,iirecn(perm_array(r))).ge.29 .and.
*         id_recn(1,iirecn(perm_array(r))).le.34) then
        mass_result = mass_part(1,omega_pid) ! parent is omega for these
        kinks
      else
        print *, 'REC.REC: warning vee/kink recdf not #20-34'
      endif
    endif          ! vk1

      p(1) = sqrt((mass_result)**2 +
&          p(2)**2 + p(3)**2 +p(4)**2)    ! energy E=sqrt(m**2+p**2)

c      print *, 'p,mass_result:',p(1),mass_result
c      Else
c          p(1) = sqrt(1.115D0**2 +
c          &          p(2)**2 + p(3)**2 +p(4)**2)
c      EndIf
c      Do i = 1,4
c          e_recn(i,n) = e_recn(i,n) + p(i)
c      EndDo          ! doing 4-vector E=e_recn(1), px=e_recn(2), etc
      EndDo
      mass_recn(1,n) = sqrt(max(0.D0, e_recn(1,n)**2
&          - px_recn(1,n)**2
&          - py_recn(1,n)**2
&          - pz_recn(1,n)**2))

c      print *, 'n,mass:',n,mass_recn(1,n)

```

```

        if(                                     ! mass cuts
*      mass_min_recdf(1,irecdf).ge.
*      mass_max_recdf(1,irecdf) .or.      ! mass cuts enabled
*      (mass_recn(1,n) .ge. mass_min_recdf(1,irecdf) ! min mass
*      .and.
*      mass_recn(1,n) .le. mass_max_recdf(1,irecdf)) ! max mass
*      ) then                                  ! We have reconstruction
c      fill this reconstruction
        id_recn(1,n) = id_recdf(1,irecdf)

        Call ucopyii(tgt_recn(1,iirecn(perm_array(1))),
&      tgt_recn(1,n),6)                      ! sec. vertex

c      print *, 'n, id_recn, mass: ', n, id_recn(1,n), mass_recn(1,n)

c      print *, 'svx_tgt_recon_1:6: ', tgt_recn(1,n), q_recn(1,n),
c      &      recdf_recn(1,n), bit_recn(1,n), pmtrk_recn(1,n)

        q_recn(1,n) = q_recdf(1,irecdf)
        recdf_recn(1,n) = irecdf
c      Write(*,*) recdf_recn(1,n)
        bit_recn(1,n) = out_recdf(1,irecdf) ! mode bits

c      print *, 'q, recdf, bit, nr: ', q_recn(1,n),
c      &      recdf_recn(1,n), bit_recn(1,n), nr(irecdf)

        Do r = 1, nr(irecdf)
            link_recn(r,n) = recn_t_recn + iirecn(perm_array(r)) ! link to
                recon
            pmtrk_recn(r,n) = 0                                ! later put partnumber

c      print *, 'r, iirecn, link: ', r, link_recn(r,n), iirecn(perm_array(r)),
c      &      iirecn(perm_array(r))

        EndDo

        If (.not.do_refit_prim) Goto 140

        If (nr(irecdf).lt.lkretn) Then      ! otherwise no space for link
            If (ivtx.ne.0) Then              ! have tracks from primary vertex

                If (nvtx2+2.le.mvtx2) Then    ! space left in vtx2

                    nvtxp = nvtx2 + 1        ! get next free entry vtx2
                    nvtxs = nvtx2 + 2        ! get next free entry vtx2

```

```

&      Call ucopyii(nt_vtx2(1,ivtx),nt_vtx2(1,nvtxp), ! copy primary
&      nt_vtx2(1,ivtx)+1)

Do itrk = 1,nt_vtx2(1,nvtxp) ! negate secondary tracks
  Do i = 1,ntracks
    If (tracks(i).gt.0) Then
      If (list_vtx2(itrk,nvtxp).eq.tracks(i)) then
        list_vtx2(itrk,nvtxp) =
&          -list_vtx2(itrk,nvtxp)
      EndIf
    EndIf
  EndDo
EndDo

Call vertex2(nvtxp,npass,chi2_prim) ! primary fit
If (status_vtx2(1,nvtxp).ne.0 .and. ! good fit
&     chi2_vtx2(1,nvtxp).le.chi2_prim) Then
  status_vtx2(1,nvtxp) = ior(status_vtx2(1,nvtxp),
&     recon_s_vtx2+ primary_s_vtx2)

  tgt = 1
  Do itgt = 2,ntgt_pos ! primary tgt
    If (abs(z_vtx2(1,nvtxp)-z_tgt_pos(1,itgt)).lt.
&         abs(z_vtx2(1,nvtxp)-z_tgt_pos(1,tgt)))
&         tgt = itgt
  EndDo ! primary tgt
  nbeam_vtx2(1,nvtxp) = nbeam_vtx2(1,ivtx) ! save nbeam
  id_vtx2(1,nvtxp) = tgt ! save primary target
  tgt_recn(1,n) = tgt ! save prim target
  link_recn(nr(irecdf)+1,n) = pvtx2_t_recn + nvtxp
&     - beg_dir_vtx2(1,recon_vtx2)+1
  If (beg_dir_vtx2(1,recon_vtx2).eq.0)
&     beg_dir_vtx2(1,recon_vtx2) = nvtxp
  end_dir_vtx2(1,recon_vtx2) = nvtxp
  n_dir_vtx2(1,recon_vtx2) =
&     end_dir_vtx2(1,recon_vtx2) -
&     beg_dir_vtx2(1,recon_vtx2)+1

  If (.not.do_refit_sec .or.
&     pid_type(0,irecn).eq.1) Then
    nvtxs = 0
    Do i = 1,lkrecon
      if (
&         link_recn(i,mod(link_recn(1,n),
&         shift_link_recn))/svtx2_t_recn
&         .eq.1) ! we have a sec. vertex link
&         nvtxs = mod(link_recn(i,
&         mod(link_recn(1,n),shift_link_recn)),
&         shift_link_recn)
    EndDo
    If (nvtxs.eq.0) Goto 150

```

```

        nvtxs = nvtxs + beg_dir_vtx2(1,recon_vtx2) - 1
      EndIf
    Else
      ! prim fit failed
      Write(*,*) 'Primary_vertex_fit_failed'
      Goto 150
    EndIf
  Else
    ! space in vtx2
    Goto 900
  EndIf
  ! space in vtx2
  ! have tracks from primary
  Goto 900
Endif
! have tracks from primary
! space in link
Else
  Goto 900
EndIf
! space in link

If (do_refit_sec .and. pid_type(0,iirecn).eq.0) Then
  If (nr(iirecdf).lt.lkreco-1) Then ! otherwise no space for link
    if (ntrk+1.le.mtrk) Then ! space in trk
      if (ntseg+1.le.mtseg) Then !space in tseg

        jprim = mod(link_reco(1,iirecn(perm_array(2))),
          & shift_link_reco) !400
        lcnvtx_virt = link_reco(1,iirecn(perm_array(2))) +
        & beg_dir_vtx2(1,recon_vtx2)-1-pvtx2_t_reco+1
        i_pvtx = jprim +beg_dir_vtx2(1,recon_vtx2)-1
        jsec = mod(link_reco(2,iirecn(perm_array(2))),
        & shift_link_reco) !400
        i_svtx = jsec + beg_dir_vtx2(1,recon_vtx2)-1 !400
        xsec = x_vtx2(1,i_svtx)
        ysec = y_vtx2(1,i_svtx)
        zsec = z_vtx2(1,i_svtx) !z 400
        jprim2 = mod(link_reco(1,iirecn(perm_array(1))),
        & shift_link_reco) !400
        i_pvtx2 = jprim2 +beg_dir_vtx2(1,recon_vtx2)-1 ! 500
        jsec2 = mod(link_reco(2,iirecn(perm_array(1))),
        & shift_link_reco) !500
        i_svtx2 = jsec2 + beg_dir_vtx2(1,recon_vtx2)-1
        zsec2 = z_vtx2(1,i_svtx2) !z 500
        ztt_virt = zsec - zsec2
        pxsect = 0.
        pysect = 0.
        pzsect = 0.
        ttx = 0.
        tty = 0.
        sigx_virt = 0.
        sigy_virt = 0.

c    calculate x, y intercept and x y slope for virtual track
      Do i=1,ntrackssec2
        r_tmp = curv_trk(1,trackssec2(i))

```

```

isegsec = seg_trk(vx_spec, trackssec2(i))
pzsec = abs(1./sqrt(1+tx_tseg(1, isegsec)**2)
&      /r_tmp) !pz
pxsec = pzsec*tx_tseg(1, isegsec) !px
pysec = pzsec*ty_tseg(1, isegsec) !py
tx = pzsec * tx_tseg(1, isegsec)
ty = pzsec * ty_tseg(1, isegsec)
ttx = tx + ttx
tty = ty + tty
pxsect = pxsec + pxsect
pysect = pysec + pysect
pzsect = pzsec + pzsect
sigx_virt = (pzsec**2)*stx_tseg(1, isegsec) +
&      sigx_virt
sigy_virt = (pzsec**2)*sty_tseg(1, isegsec) +
&      sigy_virt
End do
sigx_virt = sigx_virt/pzsect**2 !err**2 in theta x for v trk
sigy_virt = sigy_virt/pzsect**2 !err**2 in theta y for v trk
param = -zsec/pzsect
xsec = xsec + (pxsect*param)
ysec = ysec + (pysect*param)
zsec = 0
bx = ttx/pzsect
by = tty/pzsect

nptracks = nt_vtx2(1, nvtxp) ! tracks in prim vertex

ntrackssec = ntrackssec + 1
trackssec(ntrackssec) = ntrk+1 ! add new track
seg_trk(vx_spec, ntrk+1) = ntseg + 1 !add vx segment for vtrk
status_trk(1, ntrk+1) = ior(virt_s_trk,
&      status_trk(1, ntrk+1))
nptracks = nptracks + 1
ntseg = ntseg + 1
list_vtx2(nptracks, nvtxp) = -(ntrk+1)

```

c adding ax, ay, bx and by **to common block**

```

x_tseg(1, seg_trk(vx_spec, ntrk+1)) = xsec
y_tseg(1, seg_trk(vx_spec, ntrk+1)) = ysec
tx_tseg(1, seg_trk(vx_spec, ntrk+1)) = bx
ty_tseg(1, seg_trk(vx_spec, ntrk+1)) = by

Do i = 1, ntrackssec
  list_vtx2(i, nvtxs) = trackssec(i)
EndDo
nt_vtx2(1, nvtxs) = ntrackssec
Call vertex2(nvtxs, 1, chi2_sec) ! virtual trk fit
if (status_vtx2(1, nvtxs).ne.0.and.
&      chi2_vtx2(1, nvtxs).le.chi2_sec) Then !good fit \w vtrk
  status_vtx2(1, nvtxs) = ior(status_vtx2(1, nvtxs),

```

```

&          recon_s_vtx2+ secondary_s_vtx2)
link_recn(nr(irecdf)+2,n) =
&          svtx2_t_recn + nvtxs
&          - beg_dir_vtx2(1,recon_vtx2) + 1 !update link to svtx
Call ucoppyrr(x_vtx2(1,nvtxs),x_recn(1,n),3) !copy vtx
          location
ELSE
c      Write(*,*) 'RECON_RECON_NO_GOOD_FIT_WITH_VTRK'
c      &          ,chi2_vtx2(1,nvtxs)
          ntseg = ntseg -1 ! clear tseg
          nt_vtx2(1,nvtxs) = nt_vtx2(1,nvtxs)-1 !clear number of
          trks
          Goto 150
Endif !good fit with vtrk
          ntseg = ntseg -1 ! clear tseg
          nt_vtx2(1,nvtxs) = nt_vtx2(1,nvtxs)-1 !clear number of trks
Else ! space in tseg
c      Write(*,*) 'RECON_RECON_NO_SPACE_IN_TSEG_FOR_VTRK'
          Goto 900
Endif ! space in tseg
ELSE ! space in track
c      Write(*,*) 'RECON_RECON_NO_SPACE_IN_TRK_FOR_VTRK'
          Goto 900
Endif ! space in trk
Else ! otherwise no space for link
          Goto 900
EndIf ! otherwise no space for link
EndIf ! do secondary vertex refit

c Recalculate l and sigma
z1 = z_vtx2(1,nvtxp)
z2 = z_vtx2(1,nvtxs)
w1 = 1./max(1.e-6,verr_vtx2(9,nvtxp))
wt = 3./z_tgt_sz(1,lksz_tgt_pos(1,tgt))*2
zt = z_tgt_pos(1,tgt)

zp = (w1*z1+wt*zt)/(w1+wt)
sig_zp= 1./sqrt(w1+wt)
ndof = 1
chi2 = wt*(zt-zp)**2 + w1*(z1-zp)**2
zd = z2 - zp ! redefine L
sigma =
&      sqrt(max(1.e-6,verr_vtx2(9,nvtxs)) +
&      sig_zp**2) ! L error
l_recn(1,n) = zd
sigma_recn(1,n) = sigma
chi2_recn(1,n) = chi2
ndof_recn(1,n) = ndof
pvtx_recn(1,n) = recon_pvtx(n) ! pointback rchi2
nvtx2 = nvtx2 + 2 !update pointer

```

```

140      Continue
      Do r = 1,nr(irecdf) ! fill all sub-reconstruction
        Call ucopyrr(recn(1,iirecn(perm_array(r))),
&          recn(1,n+r),lrecn)
        bit_recn(1,n+r) = ior(bit_recn(1,n+r),
&          iand(out_recdf(1,irecdf),X1F)) ! transfer output bits

        id_recn(1,n+r) = id_recdf(1,irecdf)*1000
&          + id_recn(1,n+r)
      EndDo

      nrecn = n + nr(irecdf) ! update size of recn
      EndIf ! mass cut

150      Continue
      Call Combi(perm_array,nirecn,nr(irecdf)) ! CERNLIB V201
      EndDo ! loop over all permutation

c      900 print *, 'exiting_recon_recon'
900      Continue
      return

1001 Format ('_Recon_Recon_Pid_',I3,':_Combining_',I1,'_recons:_',%)
1002 Format (I3,'_',%)
1101 Format ('_In_Recon_Recon:_'_Using_',F4.1,'_for_primary_and_',
&          F4.1,'_for_secondary_chi2_cut_',I2,'_for_pruning_pass')
1103 Format ('_In_Recon_Recon:_'_Will',A,'refit_primary_and_will',A,
&          'refit_secondary_vertex')
      end

c
c $Log: recon_recon.F,v $
c Revision 1.34 2010/08/04 14:32:36 jurgen
c declare recon_pvtx
c
c Revision 1.33 2010/08/03 00:17:18 jurgen
c calculate pointback
c
c Revision 1.32 2008/12/10 16:06:56 jurgen
c remove unused variables and includes
c
c Revision 1.31 2007/06/15 18:28:48 jurgen
c do not overwrite last entry in recn, but rather leave in case of overflow
c
c Revision 1.30 2007/06/12 16:56:57 jurgen
c move nvtxs by beg_dir for no sec. fit
c
c Revision 1.29 2007/06/11 00:21:32 jurgen
c fix again identical ids
c
c Revision 1.28 2007/06/07 21:57:42 jurgen
c fix handling of identical subrecons

```

```

c
c Revision 1.27  2007/06/06 22:50:52  jurgen
c fix loop over links , use typesafe vzero and ucopy
c
c Revision 1.26  2007/05/04 13:09:44  jurgen
c fix typo
c
c Revision 1.25  2007/05/03 18:12:15  jurgen
c fix bug when using same recons , improve printouts
c
c Revision 1.24  2006/08/28 22:05:32  jurgen
c fix indirect reference
c
c Revision 1.23  2006/08/25 16:25:10  jurgen
c make code cleaner
c
c Revision 1.22  2006/08/01 14:50:11  jurgen
c better printout , fix boundary check
c
c Revision 1.21  2006/02/03 03:15:16  jurgen
c fix endif order with printouts
c
c Revision 1.19  2006/01/28 18:44:43  jurgen
c add sec vertex chi2 , adopt for larger vtx2 block
c
c Revision 1.18  2006/01/28 03:06:34  jurgen
c add secondary vertex fit to recon-recon (code by I.Torres)
c
c Revision 1.17  2003/03/06 19:48:37  jurgen
c get prim. vertex in rr now from calculation
c
c Revision 1.16  2003/03/05 00:30:09  jurgen
c make primary vertex fit in recon-recon
c
c Revision 1.15  2002/12/18 01:35:42  jurgen
c add writeouts in compilation
c
c Revision 1.14  2000/06/26 23:08:48  braunger
c 1. Removed obsolete/unused pmtrk pointers for vees/kinks
c 2. Added cut on vee/kink hyperon mass to make baryons; using
c the 1/sigma area in recdf table.
c
c Revision 1.13  2000/03/11 05:16:06  braunger
c For rr of vees: store hyperon delta mass in dmass_recn
c & rel likelihood in pmtrk5/6 in order to put cuts on vee to improve charm
c
c Revision 1.12  2000/02/23 22:35:17  braunger
c corrected parent id for kinks 29-34 (is omega, not xi-)
c
c Revision 1.11  1999/11/05 16:00:50  syjun
c more fix hex operation for linux

```

c
c Revision 1.10 1999/11/05 15:54:51 syjun
c fix hex operation for linux
c
c Revision 1.9 1999/10/26 22:41:44 syjun
c merge linux-port
c
c Revision 1.8 1999/08/17 00:07:06 syjun
c store negative index **in** track list **in case** of gamma
c
c Revision 1.6.2.1 1999/08/02 21:47:57 procario
c Linux port
c
c Revision 1.7 1999/07/22 16:10:16 pgouffon
c fix **call to** vzero with a comma instead of a * (flagged as wrong **number** of
c arguments during Linux port
c
c Revision 1.6 1999/06/10 22:59:48 newsom
c Add protection for vee/kink nonusers **if** they **use** vee/kink id
c
c Revision 1.5 1999/06/09 20:41:23 newsom
c **Use** particle table hyperon masses **in** mass calculations.
c
c Revision 1.4 1998/01/29 18:30:50 jurgen
c fix constant **to double precision**
c
c Revision 1.3 1998/01/29 18:13:05 jurgen
c put correct id **in** id word of used recon
c
c Revision 1.2 1998/01/28 21:51:59 pcooper
c collected changes and bug fixes for pass11_7
c
c Revision 1.1 1998/01/22 18:26:24 jurgen
c Add recon_recon

Apéndice C

Programa *anal.F*

A continuación se presenta el programa de *anal.F*.

```

Program anal
Implicit none
Character*256 ftupfile
Integer entries ,PawSize ,HMEMOR

#include "recon_ntuple.inc"

Logical ftup_open
Integer iargc ,narg
Integer idi ,ilam ,nlam ,plam
Real mass_l_nom ,p ,z ,gamma ,ctau ,lmass (2)
Real ppa (4 ,2) ,pout (4) ,efms ,ams (2) ,c ,dc ,rki

Integer nbinmom ,nbinz ,ihist (6)
Real maxmom ,maxz

c      Esto es lo nuevo:
Real lpv (3) ,lpvxs (2) ,x ,y ,dx ,dy ,wx ,wy ,lsv (3) ,lsvxs (2) ,sx ,sy
Real lpvtx (2) ,lp (3) ,en (2) , llmass , ene (2) , mass2
Real pp2 (3) , hls , sv (2) , massl

c      Data mass_l_nom /1.1159/
Data mass_l_nom /1.1156/
Data ctau /7.89/

External ftup_open

Parameter (PawSize=400000)
COMMON/PAWC/HMEMOR(PawSize)
```

```

If (iargc().lt.1) Then
  Write(*,*) 'Usage: _anal_ftup_file(s)'
  Call Exit(1)
EndIf

Call HLIMIT(PawSize)

Call HBOOK1(2, 'Lambda', 100, 1.090, 1.140, 0.)
Call HBOOK1(3, 'antiLambda', 100, 1.090, 1.140, 0.)
Call HBOOK1(4, 'proton, pion-', 100, 1.06, 1.112, 0.)
Call HBOOK1(5, 'proton-, pion', 100, 1.06, 1.112, 0.)

Call HBOOK1(801, 'Lambda_Lambda', 100, 2.15, 3., 0.)
Call HBOOK1(802, 'aLambda_aLambda', 100, 2.15, 3., 0.)
Call HBOOK1(803, 'Lambda_aLambda', 100, 2.15, 3., 0.)
Call HBOOK1(804, 'H', 100, 2., 2.4, 0.)
Call HBOOK1(805, 'H2', 100, 2., 2.4, 0.)

Call HBOOK1(1801, 'pointback_ll', 100, 0., 30., 0.)
Call HBOOK1(1802, 'pointback_aa', 100, 0., 30., 0.)
Call HBOOK1(1803, 'pointback_la', 100, 0., 30., 0.)

Call HBOOK1(3801, 'Lambda_Lambda', 500, 2.2, 3.5, 0.)
Call HBOOK1(3802, 'aLambda_aLambda', 500, 2.2, 3.5, 0.)
Call HBOOK1(3803, 'Lambda_aLambda', 500, 2.2, 3.5, 0.)
Call HBOOK1(4801, 'Lambda_Lambda', 200, 2.2, 3.5, 0.)
Call HBOOK1(4802, 'aLambda_aLambda', 200, 2.2, 3.5, 0.)
Call HBOOK1(4803, 'Lambda_aLambda', 200, 2.2, 3.5, 0.)
Call HBOOK1(5801, 'Lambda_Lambda', 100, 2.2, 3.5, 0.)
Call HBOOK1(5802, 'aLambda_aLambda', 100, 2.2, 3.5, 0.)
Call HBOOK1(5803, 'Lambda_aLambda', 100, 2.2, 3.5, 0.)
Call HBOOK1(6801, 'Lambda_Lambda', 50, 2.2, 3.5, 0.)
Call HBOOK1(6802, 'aLambda_aLambda', 50, 2.2, 3.5, 0.)
Call HBOOK1(6803, 'Lambda_aLambda', 50, 2.2, 3.5, 0.)

Call HBOOK1(7801, 'Lambda_aLambda', 66, 2.2, 3.5, 0.)

Call HBOOK2(2801, 'Lambda_vs_Lambda',
& 100, 1.090, 1.140, 100, 1.090, 1.140, 0.)
Call HBOOK2(2802, 'Antilambda_vs_Antilambda',
& 100, 1.090, 1.140, 100, 1.090, 1.140, 0.)
Call HBOOK2(2803, 'Lambda_vs_Antilambda',
& 100, 1.090, 1.140, 100, 1.090, 1.140, 0.)

Call HBOOK1(8804, '1/sigma_H', 100, -10., 10., 0.)
Call HBOOK1(8805, '1/sigma_H2', 100, -10., 10., 0.)

Call HBOOK1(3804, 'Lambda_ppi', 5, 2.1, 2.5, 0.)
Call HBOOK1(3805, 'aLambda_p-pi', 5, 2.1, 2.5, 0.)
Call HBOOK1(4804, 'Lambda_ppi', 10, 2., 2.4, 0.)

```

```

Call HBOOK1(4805, 'aLambda_p-pi', 10, 2., 2.4, 0.)
Call HBOOK1(5804, 'Lambda_ppi', 15, 2., 2.4, 0.)
Call HBOOK1(5805, 'aLambda_p-pi', 15, 2., 2.4, 0.)
Call HBOOK1(6804, 'Lambda_ppi', 20, 2.1, 2.5, 0.)
Call HBOOK1(6805, 'aLambda_p-pi', 20, 2.1, 2.5, 0.)
Call HBOOK1(7804, 'Lambda_ppi', 30, 2.1, 2.5, 0.)
Call HBOOK1(7805, 'aLambda_p-pi', 30, 2.1, 2.5, 0.)
Call HBOOK2(2804, 'HvsH',
&      100, 1.060, 1.112, 100, 1.060, 1.112, 0.)
Call HBOOK2(2805, 'H2_vs_H2',
&      100, 1.060, 1.112, 100, 1.060, 1.112, 0.)

Call HBOOK1(9804, 'Lambda_ppi', 25, 2.19, 2.24, 0.)
Call HBOOK1(9805, 'aLambda_p-pi', 25, 2.19, 2.24, 0.)

Call HBOOK1(10804, 'Lambda_ppi', 25, 2.19, 2.24, 0.)
Call HBOOK1(10805, 'aLambda_p-pi', 25, 2.19, 2.24, 0.)

Call HBOOK1(11804, 'Lambda_ppi', 25, 2.19, 2.24, 0.)
Call HBOOK1(11805, 'aLambda_p-pi', 25, 2.19, 2.24, 0.)

Call HBOOK1(12804, 'Lambda_ppi', 25, 2.19, 2.24, 0.)
Call HBOOK1(12805, 'aLambda_p-pi', 25, 2.19, 2.24, 0.)

Call HBOOK1(13804, 'Lambda_ppi', 25, 2.19, 2.24, 0.)
Call HBOOK1(13805, 'aLambda_p-pi', 25, 2.19, 2.24, 0.)

Call HBOOK1(14804, 'Lambda_ppi', 25, 2.19, 2.24, 0.)
Call HBOOK1(14805, 'aLambda_p-pi', 25, 2.19, 2.24, 0.)

Call HBOOK1(15804, 'Lambda_ppi', 25, 2.19, 2.24, 0.)
Call HBOOK1(15805, 'aLambda_p-pi', 25, 2.19, 2.24, 0.)

Call HBOOK1(16804, 'Lambda_ppi', 25, 2.19, 2.24, 0.)
Call HBOOK1(16805, 'aLambda_p-pi', 25, 2.19, 2.24, 0.)

Call HBOOK1(17804, 'Lambda_ppi', 25, 2.19, 2.24, 0.)
Call HBOOK1(17805, 'aLambda_p-pi', 25, 2.19, 2.24, 0.)

Call HBOOK1(18804, 'Lambda_ppi', 25, 2.19, 2.24, 0.)
Call HBOOK1(18805, 'aLambda_p-pi', 25, 2.19, 2.24, 0.)

Call HBOOK1(19804, 'Lambda_ppi', 25, 2.19, 2.24, 0.)
Call HBOOK1(19805, 'aLambda_p-pi', 25, 2.19, 2.24, 0.)

Call HBOOK1(20804, 'Lambda_ppi', 25, 2.19, 2.24, 0.)
Call HBOOK1(20805, 'aLambda_p-pi', 25, 2.19, 2.24, 0.)

Call HBOOK1(21804, 'Lambda_ppi', 25, 2.19, 2.24, 0.)
Call HBOOK1(21805, 'aLambda_p-pi', 25, 2.19, 2.24, 0.)

```

Call HBOOK1(22804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(22805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(23804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(23805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(24804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(24805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(25804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(25805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(26804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(26805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(27804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(27805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(28804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(28805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(29804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(29805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(30804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(30805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(31804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(31805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(32804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(32805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(33804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(33805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(34804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(34805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(35804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(35805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(36804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(36805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(37804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(37805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

Call HBOOK1(38804, 'Lambda_{ppi}', 25, 2.19, 2.24, 0.)
 Call HBOOK1(38805, 'aLambda_{p-pi}', 25, 2.19, 2.24, 0.)

```

Call HBOOK1(39804,'Lambda_p pi ',25,2.19,2.24,0.)
Call HBOOK1(39805,'aLambda_p pi ',25,2.19,2.24,0.)

Call HBOOK1(40804,'Lambda_p pi ',25,2.19,2.24,0.)
Call HBOOK1(40805,'aLambda_p pi ',25,2.19,2.24,0.)

Call HBOOK1(94804,'Lambda_p pi ',25,2.19,2.24,0.)
Call HBOOK1(94805,'aLambda_p pi ',25,2.19,2.24,0.)

entries = 0

Do narg = 1,iargc()

    Call getarg(narg,ftupfile)

    If (.not.Ftup_Open(10,ftupfile)) Goto 999

Do While (.True.)
    read(10,end=12,err=11) tuple
    If (run.le.0) Goto 12
    entries = entries + 1
    idi = id

    If (idi.ge.2.and.idi.le.5)Then
        Call HFILL(idi,mass,0.,1.)
    EndIf

    If (idi.ge.801 .and. idi.le.805)Then
c        vertice primario
            lpv(1) = pvx_x
            lpv(2) = pvx_y
            lpv(3) = pvx_z
c        errores del primer vertice
            lpvxs(1) = pvx_sx
            lpvxs(2) = pvx_sy

c            massl = mass
            nlam = 0
            ilam = 0
            plam = 0

            Call HFILL(idi,mass,0.,1.)

    EndIf

    If (idi.eq.801002 .or. idi.eq.802003 .or.
&        idi.eq.803002 .or. idi.eq.803003.or.
&        idi.eq.804002 .or. idi.eq.805003.or.
&        idi.eq.804004 .or. idi.eq.805005)Then
        nlam = nlam + 1

```

```

      lmass(nlam) = mass
c     vertice secundario
      lsv(1) = svx_x
      lsv(2) = svx_y
      lsv(3) = svx_z
c     errores del segundo vertice
      lsvxs(1) = svx_sx
      lsvxs(2) = svx_sy
c     momento
      lp(1) = px
      lp(2) = py
      lp(3) = pz
c     Calculo de point back
      sx = lp(1)/lp(3)
      sy = lp(2)/lp(3)
      x = lsv(1) - lpv(1)
      y = lsv(2) - lpv(2)
      z = lsv(3) - lpv(3)
      dx = lsvxs(1)**2 + lpvxs(1)**2
      dy = lsvxs(2)**2 + lpvxs(2)**2
      wx = 1./max(dx, 1.e-8)
      wy = 1./max(dy, 1.e-8)
      lpvtx(nlam) = (wx*(x-sx*z)**2 + wy*(y-sy*z)**2)/2

If(idi.eq.804002.or.idi.eq.805003)Then
      sv(1) = svx_z
EndIf

If(idi.eq.804004.or.idi.eq.805005)Then
      sv(2) = svx_z
EndIf

If(idi.eq.804004 .or. idi.eq.805005)Then
      ilam = ilam + 1
      ams(ilam) = mass
      ppa(1,ilam) = px
      ppa(2,ilam) = py
      ppa(3,ilam) = pz
      hls = L/sigma
Else
      If (abs(mass-mass_l_nom).lt.0.005) Then
        ilam = ilam + 1
        ams(ilam) = mass_l_nom
        ppa(1,ilam) = px
        ppa(2,ilam) = py
        ppa(3,ilam) = pz
      EndIf
EndIf

If(idi.eq.801002 .or. idi.eq.802003.or.

```

```

&      idi.eq.803002 .or. idi.eq.803003)Then
If(abs(mass-mass_l_nom).lt.0.005) Then
      plam = plam + 1
      ams(plam) = mass_l_nom
      ppa(1,plam) = px
      ppa(2,plam) = py
      ppa(3,plam) = pz
EndIf
EndIf

If (nlam.eq.2) then
  Call HFILL(2000+idi/1000,lmass(1),lmass(2),1.)
  Call HFILL(1000+idi/1000,lpvtx,0.,1.)
EndIf

If (plam.eq.2) then
  If (lpvtx(1).gt.5.or.lpvtx(2).gt.5) Goto 10
  Call efmass(2,ams,ppa,efms,pout)

  Call HFILL(3000+idi/1000,efms,0.,1.)
  Call HFILL(4000+idi/1000,efms,0.,1.)
  Call HFILL(5000+idi/1000,efms,0.,1.)
  Call HFILL(6000+idi/1000,efms,0.,1.)
  Call HFILL(7000+idi/1000,efms,0.,1.)

EndIf

If (ilam.eq.2) Then

  If(sv(2).gt.sv(1)) Goto 10
  Call efmass(2,ams,ppa,efms,pout)

  If (hls.ge.1.) Then
    Call HFILL(9000+idi/1000,efms,0.,1.)
  EndIf

  If(hls.ge.1.1) Then
    Call HFILL(10000+idi/1000,efms,0.,1.)
  EndIf

  If(hls.ge.1.15) Then
    Call HFILL(11000+idi/1000,efms,0.,1.)
  EndIf

  If(hls.ge.1.2) Then
    Call HFILL(12000+idi/1000,efms,0.,1.)
  EndIf

  If(hls.ge.1.3) Then
    Call HFILL(13000+idi/1000,efms,0.,1.)

```

EndIf

If(hls.ge.1.4) **Then**

Call HFILL(14000+idi/1000,efms,0.,1.)

EndIf

If(hls.ge.1.5) **Then**

Call HFILL(15000+idi/1000,efms,0.,1.)

EndIf

If(hls.ge.1.6) **Then**

Call HFILL(16000+idi/1000,efms,0.,1.)

EndIf

If(hls.ge.1.7) **Then**

Call HFILL(17000+idi/1000,efms,0.,1.)

EndIf

If(hls.ge.1.8) **Then**

Call HFILL(18000+idi/1000,efms,0.,1.)

EndIf

If(hls.ge.1.9) **Then**

Call HFILL(19000+idi/1000,efms,0.,1.)

If (efms.le.2.24 .and. idi/1000.eq.804)

Write(*,*) run,event,massl,efms

EndIf

If(hls.ge.1.94) **Then**

Call HFILL(94000+idi/1000,efms,0.,1.)

EndIf

If(hls.ge.2.) **Then**

Call HFILL(20000+idi/1000,efms,0.,1.)

EndIf

If(hls.ge.2.1) **Then**

Call HFILL(21000+idi/1000,efms,0.,1.)

EndIf

If(hls.ge.2.2) **Then**

Call HFILL(22000+idi/1000,efms,0.,1.)

EndIf

If(hls.ge.2.3) **Then**

Call HFILL(23000+idi/1000,efms,0.,1.)

EndIf

If(hls.ge.2.4) **Then**

Call HFILL(24000+idi/1000,efms,0.,1.)

EndIf

c

c

&

```
If(hls.ge.2.5) Then  
    Call HFILL(25000+idi/1000,efms,0.,1.)  
EndIf
```

```
If(hls.ge.2.6) Then  
    Call HFILL(26000+idi/1000,efms,0.,1.)  
EndIf
```

```
If(hls.ge.2.7) Then  
    Call HFILL(27000+idi/1000,efms,0.,1.)  
EndIf
```

```
If(hls.ge.2.8) Then  
    Call HFILL(28000+idi/1000,efms,0.,1.)  
EndIf
```

```
If(hls.ge.2.9) Then  
    Call HFILL(29000+idi/1000,efms,0.,1.)  
EndIf
```

```
If(hls.ge.3.) Then  
    Call HFILL(30000+idi/1000,efms,0.,1.)  
EndIf
```

```
If(hls.ge.3.1) Then  
    Call HFILL(31000+idi/1000,efms,0.,1.)  
EndIf
```

```
If(hls.ge.3.2) Then  
    Call HFILL(32000+idi/1000,efms,0.,1.)  
EndIf
```

```
If(hls.ge.3.3) Then  
    Call HFILL(33000+idi/1000,efms,0.,1.)  
EndIf
```

```
If(hls.ge.3.4) Then  
    Call HFILL(34000+idi/1000,efms,0.,1.)  
EndIf
```

```
If(hls.ge.3.5) Then  
    Call HFILL(35000+idi/1000,efms,0.,1.)  
EndIf
```

```
If(hls.ge.3.6) Then  
    Call HFILL(36000+idi/1000,efms,0.,1.)  
EndIf
```

```
If(hls.ge.3.7) Then  
    Call HFILL(37000+idi/1000,efms,0.,1.)
```

```

        EndIF

        If(hls.ge.3.8) Then
            Call HFILL(38000+idi/1000,efms,0.,1.)
        EndIF

        If(hls.ge.3.9) Then
            Call HFILL(39000+idi/1000,efms,0.,1.)
        EndIF

        If(hls.ge.4.) Then
            Call HFILL(40000+idi/1000,efms,0.,1.)
        EndIF

    EndIf

EndIf

10      Continue
      EndDo

11      Continue
      Write(*,*) 'Error reading file , continuing '
12      Continue
      Close(10)

EndDo

999    Continue
      Close(10)
      Write(*,*) 'Read',entries,'_Entries'

      Call HRPUT(0,'anal.hbk','NT')    ! write all histograms to file

End

```

Apéndice D

Programa *sc.kumac*

A continuación se presenta el programa *sc.kumac*.

```
h/file 1 anal.hbk.mix
hrin 0
close 1

*entradas de proton/pion
anone = $HINFO(55804004,'ENTRIES')
acero = $HINFO(50804004,'ENTRIES')
auno = $HINFO(51804004,'ENTRIES')
ados = $HINFO(52804004,'ENTRIES')
atres = $HINFO(53804004,'ENTRIES')
acuatro = $HINFO(54804004,'ENTRIES')

*entradas y allchan de H con masa de 2.19 a 2.24
cnone = $HINFO(45804,'ENTRIES')
ccero = $HINFO(41804,'ENTRIES')
cuno = $HINFO(9804,'ENTRIES')
cdos = $HINFO(20804,'ENTRIES')
ctres = $HINFO(30804,'ENTRIES')
ccuatro = $HINFO(40804,'ENTRIES')

dnone = $HINFO(45804,'SUM')
dcero = $HINFO(41804,'SUM')
duno = $HINFO(9804,'SUM')
ddos = $HINFO(20804,'SUM')
dtres = $HINFO(30804,'SUM')
dcuatro = $HINFO(40804,'SUM')

*entradas y allchan de H con masa de 2.19 a 2.29
```

```

gnone   = $HINFO(65804,'ENTRIES')
gcero   = $HINFO(60804,'ENTRIES')
guno    = $HINFO(61804,'ENTRIES')
gdos    = $HINFO(62804,'ENTRIES')
gtres   = $HINFO(63804,'ENTRIES')
gcuatro = $HINFO(64804,'ENTRIES')

```

```

hnone   = $HINFO(65804,'SUM')
hcero   = $HINFO(60804,'SUM')
huno    = $HINFO(61804,'SUM')
hdos    = $HINFO(62804,'SUM')
htres   = $HINFO(63804,'SUM')
hcuatro = $HINFO(64804,'SUM')

```

*entradas y allchan de H con masa de 2.19 a 4.

```

knone   = $HINFO(85804,'ENTRIES')
kcero   = $HINFO(80804,'ENTRIES')
kuno    = $HINFO(81804,'ENTRIES')
kdos    = $HINFO(82804,'ENTRIES')
ktres   = $HINFO(83804,'ENTRIES')
kcuatro = $HINFO(84804,'ENTRIES')

```

```

lnone   = $HINFO(85804,'SUM')
lcero   = $HINFO(80804,'SUM')
luno    = $HINFO(81804,'SUM')
ldos    = $HINFO(82804,'SUM')
ltres   = $HINFO(83804,'SUM')
lcuatro = $HINFO(84804,'SUM')

```

h/del 0

```

h/file 1 anal.hbk.strip
hrin 0
close 1

```

*entradas de proton/pion

```

bnone   = $HINFO(55804004,'ENTRIES')
bcero   = $HINFO(50804004,'ENTRIES')
buno    = $HINFO(51804004,'ENTRIES')
bdos    = $HINFO(52804004,'ENTRIES')
btres   = $HINFO(53804004,'ENTRIES')
bcuatro = $HINFO(54804004,'ENTRIES')

```

*entradas y allchan de H con masa de 2.19 a 2.24

```

enone   = $HINFO(45804,'ENTRIES')
ecero   = $HINFO(41804,'ENTRIES')
euno    = $HINFO(9804,'ENTRIES')
edos    = $HINFO(20804,'ENTRIES')
etres   = $HINFO(30804,'ENTRIES')
ecuatro = $HINFO(40804,'ENTRIES')

```

```

fnone   = $HINFO(45804, 'SUM')
fcero   = $HINFO(41804, 'SUM')
funo    = $HINFO(9804, 'SUM')
fdos    = $HINFO(20804, 'SUM')
ftres   = $HINFO(30804, 'SUM')
fcuatro = $HINFO(40804, 'SUM')

```

*entradas y allchan de H con masa de 2.19 a 2.29

```

inone   = $HINFO(65804, 'ENTRIES')
icero   = $HINFO(60804, 'ENTRIES')
iuno    = $HINFO(61804, 'ENTRIES')
idos    = $HINFO(62804, 'ENTRIES')
itres   = $HINFO(63804, 'ENTRIES')
icuatro = $HINFO(64804, 'ENTRIES')

```

```

jnone   = $HINFO(65804, 'SUM')
jcero   = $HINFO(60804, 'SUM')
juno    = $HINFO(61804, 'SUM')
jdos    = $HINFO(62804, 'SUM')
jtres   = $HINFO(63804, 'SUM')
jcuatro = $HINFO(64804, 'SUM')

```

*entradas y allchan de H con masa de 2.19 a 4.

```

mnone   = $HINFO(85804, 'ENTRIES')
mcero   = $HINFO(80804, 'ENTRIES')
muno    = $HINFO(81804, 'ENTRIES')
mdos    = $HINFO(82804, 'ENTRIES')
mtres   = $HINFO(83804, 'ENTRIES')
mcuatro = $HINFO(84804, 'ENTRIES')

```

```

nnone   = $HINFO(85804, 'SUM')
ncero   = $HINFO(80804, 'SUM')
nuno    = $HINFO(81804, 'SUM')
ndos    = $HINFO(82804, 'SUM')
ntres   = $HINFO(83804, 'SUM')
ncuatro = $HINFO(84804, 'SUM')

```

*proton pion

```

cero    = ([acero] -25.*[bcero]) / [bcero]
sigma da = sqrt(1./[acero] + 1./[bcero])*[cero]
a = da
mess [a] [acero] [bcero]

```

```

uno     = ([auno] -25.*[buno]) / [buno]
sigma db = sqrt(1./[auno] + 1./[buno])*[uno]
b = db
mess [b] [auno] [buno]

```

```

dos     = ([ados] -25.*[bdos]) / [bdos]
sigma dc = sqrt(1./[ados] + 1./[bdos])*[dos]
c = dc

```

```

mess [c] [ados] [bdos]

tres = ([atres] -25.*[btres]) / [btres]
sigma dd = sqrt(1./[atres] + 1./[btres])*[tres]
d = dd
mess [d] [atres] [btres]

cuatro = ([acuatro] -25.*[bcuatro]) / [bcuatro]
sigma de = sqrt(1./[acuatro] + 1./[bcuatro])*[cuatro]
e = de
mess [e] [acuatro] [bcuatro]

none = ([anone] -25.*[bnone]) / [bnone]
sigma df = sqrt(1./[anone] + 1./[bnone])*[none]
f = df
mess [f] [anone] [bnone]

mess [none] [cero] [uno] [dos] [tres] [cuatro]

* 1/sigma entradas 2.19 a 2.24
ocero = ([ccero] -25.*[ecero]) / [ecero]
sigma dg = sqrt(1./[ccero] + 1./[ecero])*[ocero]
g = dg
mess [g] [ccero] [ecero]

ouno = ([cuno] -25.*[euno]) / [euno]
sigma dh = sqrt(1./[cuno] + 1./[euno])*[ouno]
h = dh
mess [h] [cuno] [euno]

odos = ([cdos] -25.*[edos]) / [edos]
sigma di = sqrt(1./[cdos] + 1./[edos])*[odos]
i = di
mess [i] [cdos] [edos]

otres = ([ctres] -25.*[etres]) / [etres]
sigma dj = sqrt(1./[ctres] + 1./[etres])*[otres]
j = dj
mess [j] [ctres] [etres]

ocuatro = ([ccuatro] -25.*[ecuatro]) / [ecuatro]
sigma dk = sqrt(1./[ccuatro] + 1./[ecuatro])*[ocuatro]
k = dk
mess [k] [ccuatro] [ecuatro]

onone = ([cnone] -25.*[enone]) / [enone]
sigma dl = sqrt(1./[cnone] + 1./[enone])*[onone]
l = dl
mess [l] [cnone] [enone]

mess [onone] [ocero] [ouno] [odos] [otres] [ocuatro]

```

```

*1/sigma allchan 2.19 a 2.24
pcero = ([dcero] -25.*[fcero]) / [fcero]
sigma dm = sqrt(1./[dcero] + 1./[fcero])*[pcero]
m = dm
mess [m] [dcero] [fcero]

puno = ([duno] -25.*[funo]) / [funo]
sigma dn = sqrt(1./[duno] + 1./[funo])*[puno]
n = dn
mess [n] [duno] [funo]

pdos = ([ddos] -25.*[fdos]) / [fdos]
sigma do = sqrt(1./[ddos] + 1./[fdos])*[pdos]
o = do
mess [o] [ddos] [fdos]

ptres = ([dtres] -25.*[ftres]) / [ftres]
sigma dp = sqrt(1./[dtres] + 1./[ftres])*[ptres]
p = dp
mess [p] [dtres] [ftres]

pcuatro = ([dcuatro] -25.*[fcuatro]) / [fcuatro]
sigma dq = sqrt(1./[dcuatro] + 1./[fcuatro])*[pcuatro]
q = dq
mess [q] [dcuatro] [fcuatro]

pnone = ([dnone] -25.*[fnone]) / [fnone]
sigma dr = sqrt(1./[dnone] + 1./[fnone])*[pnone]
r = dr
mess [r] [dnone] [fnone]

mess [pnone] [pcero] [puno] [pdos] [ptres] [pcuatro]

*1/sigma entries 2.19 a 2.29
qcero = ([gcero] -25.*[icero]) / [icero]
sigma ds = sqrt(1./[gcero] + 1./[icero])*[qcero]
s = ds
mess [s] [gcero] [icero]

quno = ([guno] -25.*[iuno]) / [iuno]
sigma dt = sqrt(1./[guno] + 1./[iuno])*[quno]
t = dt
mess [t] [guno] [iuno]

qdos = ([gdos] -25.*[idos]) / [idos]
sigma du = sqrt(1./[gdos] + 1./[idos])*[qdos]
u = du
mess [u] [gdos] [idos]

qtres = ([gtres] -25.*[itres]) / [itres]

```

```

sigma dv = sqrt(1./[gtres] + 1./[itres])*[qtres]
v = dv
mess [v] [gtres] [itres]

qcuatro = ([gcuatro] -25.*[icuatro]) / [icuatro]
sigma dw = sqrt(1./[gcuatro] + 1./[icuatro])*[qcuatro]
w = dw
mess [w] [gcuatro] [icuatro]

qnone = ([gnone] -25.*[inone]) / [inone]
sigma dx = sqrt(1./[gnone] + 1./[inone])*[qnone]
x = dx
mess [x] [gnone] [inone]

mess [qnone] [qcero] [quno] [qdos] [qtres] [qcuatro]

*1/sigma sum 2.19 a 2.29
rcero = ([hcero] -25.*[jcero]) / [jcero]
sigma dy = sqrt(1./[hcero] + 1./[jcero])*[rcero]
y = dy
mess [y] [hcero] [jcero]

runo = ([huno] -25.*[juno]) / [juno]
sigma dz = sqrt(1./[huno] + 1./[juno])*[runo]
z = dz
mess [z] [huno] [juno]

rdos = ([hdos] -25.*[jdos]) / [jdos]
sigma daa = sqrt(1./[hdos] + 1./[jdos])*[rdos]
aa = daa
mess [aa] [hdos] [jdos]

rtres = ([htres] -25.*[jtres]) / [jtres]
sigma dab = sqrt(1./[htres] + 1./[jtres])*[rtres]
ab = dab
mess [ab] [htres] [jtres]

rcuatro = ([hcuatro] -25.*[jcuatro]) / [jcuatro]
sigma dac = sqrt(1./[hcuatro] + 1./[jcuatro])*[rcuatro]
ac = dac
mess [ac] [hcuatro] [jcuatro]

rnone = ([hnone] -25.*[jnone]) / [jnone]
sigma dad = sqrt(1./[hnone] + 1./[jnone])*[rnone]
ad = dad
mess [ad] [hnone] [jnone]

mess [rnone] [rcero] [runo] [rdos] [rtres] [rcuatro]

*1/sigma entries 2.19 a 4.
scero = ([kcero] -25.*[mcero]) / [mcero]

```

```

sigma dae = sqrt(1./[kcero] + 1./[mcero])*[scero]
ae = dae
mess [ae] [kcero] [mcero]

suno = ([kuno] -25.*[muno]) / [muno]
sigma daf = sqrt(1./[kuno] + 1./[muno])*[suno]
af = daf
mess [af] [kuno] [muno]

sdos = ([kdos] -25.*[mdos]) / [mdos]
sigma dag = sqrt(1./[kdos] + 1./[mdos])*[sdos]
ag = dag
mess [ag] [kdos] [mdos]

stres = ([ktres] -25.*[mtres]) / [mtres]
sigma dah = sqrt(1./[ktres] + 1./[mtres])*[stres]
ah = dah
mess [ah] [ktres] [mtres]

scuatro = ([kcuatro] -25.*[mcuatro]) / [mcuatro]
sigma dai = sqrt(1./[kcuatro] + 1./[mcuatro])*[scuatro]
ai = dai
mess [ai] [kcuatro] [mcuatro]

snone = ([knone] -25.*[mnone]) / [mnone]
sigma daj = sqrt(1./[knone] + 1./[mnone])*[snone]
aj = daj
mess [aj] [knone] [mnone]

mess [snone] [scero] [suno] [sdos] [stres] [scuatro]

*1/sigma sum 2.19 a 4.
tcero = ([lcero] -25.*[ncero]) / [ncero]
sigma dak = sqrt(1./[lcero] + 1./[ncero])*[tcero]
ak = dak
mess [ak] [kcero] [mcero]

tuno = ([luno] -25.*[nuno]) / [nuno]
sigma dal = sqrt(1./[luno] + 1./[nuno])*[tuno]
al = dal
mess [al] [luno] [nuno]

tdos = ([ldos] -25.*[ndos]) / [ndos]
sigma dam = sqrt(1./[ldos] + 1./[ndos])*[tdos]
am = dam
mess [am] [ldos] [ndos]

ttres = ([ltres] -25.*[ntres]) / [ntres]
sigma dan = sqrt(1./[ltres] + 1./[ntres])*[ttres]
an = dan
mess [an] [ltres] [ntres]

```

```

tcuatro = ([lcuatro] -25.*[ncuatro]) / [ncuatro]
sigma_dao = sqrt(1./[lcuatro] + 1./[ncuatro])*[tcuatro]
ao = dao
mess [ao] [lcuatro] [ncuatro]

tnone = ([lnone] -25.*[nnone]) / [nnone]
sigma_dap = sqrt(1./[lnone] + 1./[nnone])*[tnone]
ap = dap
mess [ap] [lnone] [nnone]

mess [tnone] [tcero] [tuno] [tdos] [ttres] [tcuatro]

```

Apéndice E

Programa *mix.kumac*

A continuación se presenta el programa *mix.kumac*.

```
*H de 2.19 a 2.24 entries
```

```
bscn = 1./503.94  
bsc0 = 1./452.059  
bsc1 = 1./633.406  
bsc2 = 1./865.355  
bsc3 = 1./1097.75  
bsc4 = 1./1135.68
```

```
*H de 2.19 a 2.24 sum
```

```
cscn = 1./483.43  
csc0 = 1./452.051  
csc1 = 1./557.94  
csc2 = 1./715.457  
csc3 = 1./780.019  
csc4 = 1./830.08
```

```
*H de 2.19 a 2.29 entries
```

```
dscn = 1./503.94  
dsc0 = 1./492.87  
dsc1 = 1./633.406  
dsc2 = 1./865.355  
dsc3 = 1./1097.75  
dsc4 = 1./1135.68
```

```
*H de 2.19 a 2.29 sum
```

```
escn = 1./480.289  
esc0 = 1./482.57  
esc1 = 1./605.622  
esc2 = 1./824.044
```

```

esc3 = 1./1010.38
esc4 = 1./897.767

*H de 2.19 a 4.entries
fscn = 1./503.94
fsc0 = 1./492.987
fsc1 = 1./633.406
fsc2 = 1./865.355
fsc3 = 1./1097.75
fsc4 = 1./1135.68

*H de 2.19 a 4 sum
gscn = 1./503.934
gsc0 = 1./492.981
gsc1 = 1./633.398
gsc2 = 1./865.343
gsc3 = 1./1097.73
gsc4 = 1./1135.65

*Proton pion
easc = 1./500.

*H de 2.19 a 2.24 entries
ebsc = 1./500

*H de 2.19 a 2.24 sum
ecsc = 1./500

*H de 2.19 a 2.29 entries
edsc = 1./500

*H de 2.19 a 2.29 sum
eesc = 1./500

*H de 2.19 a 4. entries
efsc = 1./500

*H de 2.19 a 4. sum
egsc = 1./500

close 1
h/del 0
vec/del *

h/file 1 anal.hbk.mix
hrin 0
close 1

* H de 2.19 a 2.24
bbin=\$HINFO(41804,'XBINS')
vec/crea bnone([bbin]) r

```

```
vec/crea bcero([bbin]) r
vec/crea buno([bbin]) r
vec/crea bdos([bbin]) r
vec/crea btres([bbin]) r
vec/crea bcuatro([bbin]) r
```

```
vec/crea benone([bbin]) r
vec/crea becero([bbin]) r
vec/crea beuno([bbin]) r
vec/crea bedos([bbin]) r
vec/crea betres([bbin]) r
vec/crea becuatro([bbin]) r
```

```
get_vec/cont 45804 bnone
get_vec/err 45804 benone
get_vec/cont 41804 bcero
get_vec/err 41804 becero
get_vec/cont 9804 buno
get_vec/err 9804 beuno
get_vec/cont 20804 bdos
get_vec/err 20804 bedos
get_vec/cont 30804 btres
get_vec/err 30804 betres
get_vec/cont 40804 bcuatro
get_vec/err 40804 becuatro
```

*ENTRIES

```
vec/op/vscale bnone [bscn] noneb
vec/op/vscale bcero [bsc0] cerob
vec/op/vscale buno [bsc1] unob
vec/op/vscale bdos [bsc2] dosb
vec/op/vscale btres [bsc3] tresb
vec/op/vscale bcuatro [bsc4] cuatrob
```

```
vec/op/vscale benone [ebsc] enoneb
vec/op/vscale becero [ebsc] ecerob
vec/op/vscale beuno [ebsc] eunob
vec/op/vscale bedos [ebsc] edosb
vec/op/vscale betres [ebsc] etresb
vec/op/vscale becuatro [ebsc] ecuatrob
```

*SUM

```
vec/op/vscale bnone [cscn] nonec
vec/op/vscale bcero [csc0] ceroc
vec/op/vscale buno [csc1] unoc
vec/op/vscale bdos [csc2] dosc
vec/op/vscale btres [csc3] tresc
vec/op/vscale bcuatro [csc4] cuatroc
```

```
vec/op/vscale benone [ecsc] enonec
vec/op/vscale becero [ecsc] eceroc
```

vec/op/vscale	beuno	[ecsc]	eunoc
vec/op/vscale	bedos	[ecsc]	edosc
vec/op/vscale	betres	[ecsc]	etresc
vec/op/vscale	becuatro	[ecsc]	ecuatroc

*ENTRIES

h/copy	45804	-45804	
h/put_vec/cont	-45804	noneb	
h/put_vec/err	-45804	enoneb	

h/copy	41804	-41804	
h/put_vec/cont	-41804	cerob	
h/put_vec/err	-41804	ecerob	

h/copy	9804	-9804	
h/put_vec/cont	-9804	unob	
h/put_vec/err	-9804	eunob	

h/copy	20804	-20804	
h/put_vec/cont	-20804	dosb	
h/put_vec/err	-20804	edosb	

h/copy	30804	-30804	
h/put_vec/cont	-30804	tresb	
h/put_vec/err	-30804	etresb	

h/copy	40804	-40804	
h/put_vec/cont	-40804	cuatrob	
h/put_vec/err	-40804	ecuatrob	

*SUM

h/copy	45804	-245804	
h/put_vec/cont	-245804	nonec	
h/put_vec/err	-245804	enonec	

h/copy	41804	-241804	
h/put_vec/cont	-241804	ceroc	
h/put_vec/err	-241804	eceroc	

h/copy	9804	-29804	
h/put_vec/cont	-29804	unoc	
h/put_vec/err	-29804	eunoc	

h/copy	20804	-220804	
h/put_vec/cont	-220804	dosc	
h/put_vec/err	-220804	edosc	

h/copy	30804	-230804	
h/put_vec/cont	-230804	tresc	
h/put_vec/err	-230804	etresc	

```
h/copy 40804 -240804
h/put_vec/cont -240804 cuatroc
h/put_vec/err -240804 ecuatroc
```

```
h/del 45804
h/del 41804
h/del 9804
h/del 20804
h/del 30804
h/del 40804
```

*H de 2.19 a 2.29

```
cbin=$HINFO(60804,'XBINS')
vec/crea cnone([cbin]) r
vec/crea ccero([cbin]) r
vec/crea cuno([cbin]) r
vec/crea cdos([cbin]) r
vec/crea ctres([cbin]) r
vec/crea ccuatro([cbin]) r
```

```
vec/crea cenone([cbin]) r
vec/crea cecero([cbin]) r
vec/crea ceuno([cbin]) r
vec/crea cedos([cbin]) r
vec/crea cetres([cbin]) r
vec/crea cecuatro([cbin]) r
```

```
get_vec/cont 65804 cnone
get_vec/err 65804 cenone
get_vec/cont 60804 ccero
get_vec/err 60804 cecero
get_vec/cont 61804 cuno
get_vec/err 61804 ceuno
get_vec/cont 62804 cdos
get_vec/err 62804 cedos
get_vec/cont 63804 ctres
get_vec/err 63804 cetres
get_vec/cont 64804 ccuatro
get_vec/err 64804 cecuatro
```

*ENTRIES

```
vec/op/vscale cnone [dscn] noned
vec/op/vscale ccero [dsc0] cerod
vec/op/vscale cuno [dsc1] unod
vec/op/vscale cdos [dsc2] dosd
vec/op/vscale ctres [dsc3] tresd
vec/op/vscale ccuatro [dsc4] cuatrod
```

```
vec/op/vscale cenone [edsc] enoned
vec/op/vscale cecero [edsc] ecerod
vec/op/vscale ceuno [edsc] eunod
```

vec/op/vscale	cedos	[edsc]	edosd
vec/op/vscale	cetres	[edsc]	etresd
vec/op/vscale	cecuatro	[edsc]	ecuatrod

*SUM

vec/op/vscale	cnone	[escn]	nonee
vec/op/vscale	ccero	[esc0]	ceroe
vec/op/vscale	cuno	[esc1]	unoe
vec/op/vscale	cdos	[esc2]	dose
vec/op/vscale	ctres	[esc3]	trese
vec/op/vscale	ccuatro	[esc4]	cuatroe

vec/op/vscale	cenone	[eesc]	enonee
vec/op/vscale	cecero	[eesc]	eceroe
vec/op/vscale	ceuno	[eesc]	eunoe
vec/op/vscale	cedos	[eesc]	edose
vec/op/vscale	cetres	[eesc]	etrese
vec/op/vscale	cecuatro	[eesc]	ecuatroe

*ENTRIES

h/copy	65804	-65804	
h/put_vec/cont	-65804	noned	
h/put_vec/err	-65804	enoned	

h/copy	60804	-60804	
h/put_vec/cont	-60804	cerod	
h/put_vec/err	-60804	ecerod	

h/copy	61804	-61804	
h/put_vec/cont	-61804	unod	
h/put_vec/err	-61804	eunod	

h/copy	62804	-62804	
h/put_vec/cont	-62804	dosd	
h/put_vec/err	-62804	edosd	

h/copy	63804	-63804	
h/put_vec/cont	-63804	tresd	
h/put_vec/err	-63804	etresd	

h/copy	64804	-64804	
h/put_vec/cont	-64804	cuatrod	
h/put_vec/err	-64804	ecuatrod	

*SUM

h/copy	65804	-265804	
h/put_vec/cont	-265804	nonee	
h/put_vec/err	-265804	enonee	

h/copy	60804	-260804	
h/put_vec/cont	-260804	ceroe	

h/put_vec/**err** -260804 eceroe

h/copy 61804 -261804

h/put_vec/cont -261804 unoe

h/put_vec/**err** -261804 eunoe

h/copy 62804 -262804

h/put_vec/cont -262804 dose

h/put_vec/**err** -262804 edose

h/copy 63804 -263804

h/put_vec/cont -263804 trese

h/put_vec/**err** -263804 etrese

h/copy 64804 -264804

h/put_vec/cont -264804 cuatroe

h/put_vec/**err** -264804 ecuatroe

h/del 65804

h/del 60804

h/del 61804

h/del 62804

h/del 63804

h/del 64804

*H de 2.19 a 4.

dbin=\$HINFO(80804,'XBINS')

vec/crea dnone([dbin]) r

vec/crea dcero([dbin]) r

vec/crea duno([dbin]) r

vec/crea ddos([dbin]) r

vec/crea dtres([dbin]) r

vec/crea dcuatro([dbin]) r

vec/crea denone([dbin]) r

vec/crea decero([dbin]) r

vec/crea deuno([dbin]) r

vec/crea dedos([dbin]) r

vec/crea detres([dbin]) r

vec/crea decuatro([dbin]) r

get_vec/cont 85804 dnone

get_vec/**err** 85804 denone

get_vec/cont 80804 dcero

get_vec/**err** 80804 decero

get_vec/cont 81804 duno

get_vec/**err** 81804 deuno

get_vec/cont 82804 ddos

get_vec/**err** 82804 dedos

get_vec/cont 83804 dtres

```

get_vec/err 83804 detres
get_vec/cont 84804 dcuatro
get_vec/err 84804 decuatro

```

*ENTRIES

```

vec/op/vscale dnone [fscn] nonef
vec/op/vscale dcero [fsc0] cerof
vec/op/vscale duno [fsc1] unof
vec/op/vscale ddos [fsc2] dosf
vec/op/vscale dtres [fsc3] tresf
vec/op/vscale dcuatro [fsc4] cuatrof

vec/op/vscale denone [efsc] enonef
vec/op/vscale decero [efsc] ecerof
vec/op/vscale deuno [efsc] eunof
vec/op/vscale dedos [efsc] edosf
vec/op/vscale detres [efsc] etresf
vec/op/vscale decuatro [efsc] ecuatrof

```

*SUM

```

vec/op/vscale dnone [gscn] noneg
vec/op/vscale dcero [gsc0] cerog
vec/op/vscale duno [gsc1] unog
vec/op/vscale ddos [gsc2] dosg
vec/op/vscale dtres [gsc3] tresg
vec/op/vscale dcuatro [gsc4] cuatrog

vec/op/vscale denone [egsc] enoneg
vec/op/vscale decero [egsc] ecerog
vec/op/vscale deuno [egsc] eunog
vec/op/vscale dedos [egsc] edosg
vec/op/vscale detres [egsc] etresg
vec/op/vscale decuatro [egsc] ecuatrog

```

*ENTRIES

```

h/copy 85804 -85804
h/put_vec/cont -85804 nonef
h/put_vec/err -85804 enonef

```

```

h/copy 80804 -80804
h/put_vec/cont -80804 cerof
h/put_vec/err -80804 ecerof

```

```

h/copy 81804 -81804
h/put_vec/cont -81804 unof
h/put_vec/err -81804 eunof

```

```

h/copy 82804 -82804
h/put_vec/cont -82804 dosf
h/put_vec/err -82804 edosf

```

```
h/copy 83804 -83804
h/put_vec/cont -83804 tresf
h/put_vec/err -83804 etresf

h/copy 84804 -84804
h/put_vec/cont -84804 cuatrof
h/put_vec/err -84804 ecuatrof
```

*SUM

```
h/copy 85804 -285804
h/put_vec/cont -285804 noneg
h/put_vec/err -285804 enoneg
```

```
h/copy 80804 -280804
h/put_vec/cont -280804 cerog
h/put_vec/err -280804 ecerog
```

```
h/copy 81804 -281804
h/put_vec/cont -281804 unog
h/put_vec/err -281804 eunog
```

```
h/copy 82804 -282804
h/put_vec/cont -282804 dosg
h/put_vec/err -282804 edosg
```

```
h/copy 83804 -283804
h/put_vec/cont -283804 tresg
h/put_vec/err -283804 etresg
```

```
h/copy 84804 -284804
h/put_vec/cont -284804 cuatrog
h/put_vec/err -284804 ecuatrog
```

```
h/del 85804
h/del 80804
h/del 81804
h/del 82804
h/del 83804
h/del 84804
```

```
h/file 1 anal.hbk.strip
hrin 0
close 1
```

```
zone 2 2
```

*Hist H de 2.19 a 2.24

*ENTRIES

```
h/plot 45804
h/plot -45804 s
```

h/plot 41804
h/plot -41804 s

h/plot 9804
h/plot -9804 s

h/plot 20804
h/plot -20804 s

h/plot 30804
h/plot -30804 s

h/plot 40804
h/plot -40804 s

*SUM
h/plot 45804
h/plot -245804 s

h/plot 41804
h/plot -241804 s

h/plot 9804
h/plot -29804 s

h/plot 20804
h/plot -220804 s

h/plot 30804
h/plot -230804 s

h/plot 40804
h/plot -240804 s

*Hist H de 2.19 a 2.29

*ENTRIES
h/plot 65804
h/plot -65804 s

h/plot 60804
h/plot -60804 s

h/plot 61804
h/plot -61804 s

h/plot 62804
h/plot -62804 s

h/plot 63804
h/plot -63804 s

h/plot 64804
h/plot -64804 s

*SUM

h/plot 65804
h/plot -265804 s

h/plot 60804
h/plot -260804 s

h/plot 61804
h/plot -261804 s

h/plot 62804
h/plot -262804 s

h/plot 63804
h/plot -263804 s

h/plot 64804
h/plot -264804 s

*Hist H de 2.19 a 4.

*ENTRIES

h/plot 85804
h/plot -85804 s

h/plot 80804
h/plot -80804 s

h/plot 81804
h/plot -81804 s

h/plot 82804
h/plot -82804 s

h/plot 83804
h/plot -83804 s

h/plot 84804
h/plot -84804 s

*SUM

h/plot 85804
h/plot -285804 s

h/plot 80804
h/plot -280804 s

h/plot 81804
h/plot -281804 s

h/plot 82804
h/plot -282804 s

h/plot 83804
h/plot -283804 s

h/plot 84804
h/plot -284804 s

Bibliografía

- [1] M.Mattson, *et al.*, “*First Observation of the Doubly Charmed Baryon Ξ_{cc}^+* ”
Phys. Rev. Lett. **89**, (2002).
- [2] A.V.Evdokimov, *et al.*, “*Confirmation of the Double Charm Baryon Ξ_{cc}^+
via its Decay to $p D^+ K^-$* ”, Phys. Lett. B **628**, 18-24 (2005)
- [3] A.Ocherashvili, M.A.Moinester, J.Russ, J.Engelfried, I.Torres, *et al.*,
“*First observation of a narrow charm-strange meson $D_{sJ^+}(2632) \rightarrow D_s \eta$ and $D^0 K^+$* ” Phys. Rev. Lett. **93**, SELEX Collaboration, 242001 (2004)
- [4] D.Diakonov, V.Petrov and M.Polyakov, “*Exotic Anti-Decuplet of Baryons: Prediction
from Chiral Solitons*” Z. Phys. A **359**, 305 (1997).
- [5] R.L.Jaffe, “*Perhaps a Stable Dihyperon*”, Phys. Rev. Lett. **38**, 195 (1977).
- [6] C.J.Yoon, *et al.*, “*Search for the H -dibaryon resonance in $^1_2C(H^-, K^+ \Lambda \Lambda X)$* ”,
Phys. Rev. C **75**, KEK-PS E522 Collaboration, 022201(R)(2007),
- [7] R.W.Stotzer, *et al.*, “*Search for the H -Dibaryon in $^3He(K^-, K^+ Hn)$* ,
Phys. Rev. Lett. **78**, BNL-E836 Collaboration, 3646 (1997).
- [8] A.Alavi-Harati, *et al.*, “*Search for the Weak Decay of a Lightly Bound H^0 Dibaryon*
Phys. Rev. Lett. **84**, KTeV Collaboration, 2593 (2003).
- [9] I.Chemakin, *et al.*, “*Strange particle production and an H -dibaryon
search in p -A collisions at the AGS*”, Nucl. Phys. A **639**, E910 Collaboration, 407c
(1998).
- [10] S.Lars, “*Properties of non= $q\bar{q}$ XYZ mesons and results of a search
for the H -dibaryon*” Proceedings of Science.
- [11] E.Vázquez Jáuregui, “*Measurement of Branching Ratios for Non Leptonic
Cabibbo-suppressed Decays of the Charmed-Strange Baryon χ_c* ”

- [12] N.Estrada Tristán, "*Estudio de las multiplicidades en interacciones hadrónicas*"
- [13] U.Dersch. "*Messung totaler Wirkungsquerschnitte mit Σ^- , p , π^- und π^+ bei 600 GeV/c Laborimpuls*"
- [14] U.Dersch, *et al.*, "*Total cross section measurements with π^- , Σ^- and protons on nuclei and nucleons around 600 GeV/c*". Nucl. Phys. **B579** (2000) 277.
- [15] N.Bondar, *et al.*, "*E781 beam transition radiation detector*", **H-Note 746**, SELEX internal report, 1995.
- [16] M.Mattson A. Kushnirenko and J. Russ. "*Geometry of vertex SSD*", **H/Note 756**, SELEX Collaboration, 1995.
- [17] J.Engelfried, "*Instrumentation*"
- [18] A.Blanco-Covarrubias, "*Medición de la sección eficaz de hadrones encantados y la dependencia nuclear α* "
- [19] N.Akchurin y A. Penzo, "*What, why, how, and when of E781 Scintillating Fiber Beam Hodoscope*", **H-Note 705**, SELEX Collaboration, 1995.
- [20] J.Amaro-Reyes, J.Engelfried, "*Efficiency of the eTRD in SELEX*", **H-Note 878**, SELEX internal report, 2007.
- [21] V.Maleev, *et al.*, "*Description and test results for DPWC and TRD in E781*", **H-note 747**, SELEX internal note, 1995.
- [22] J.Engelfried, *et al.*, "*The E781 (SELEX) RICH detector*", Nucl. Instrum. Meth., **A409**, 1998.
- [23] J.Engelfried, *et al.*, "*The SELEX Phototube RICH Detector*", Nucl. Instrum. Meth., **A431**, 1999.
- [24] I.E.T.Lungov, "*Vector Drift Chambers Database*", **H-Note 779**, SELEX internal report, 1995.
- [25] A.Kulyavtsev *et al.*, "*E781 hardware trigger and preliminary design.*" **H-Note 676**, SELEX internal report, 1999.
- [26] J.Engelfried, "*Recon*", Research note, SELEX Collaboration, 1998.
http://www.selex.fnal.gov/computing/soap/soap_frame.html.

- [27] J.You, “*Event Embedding in SELEX*”,
H-Note 815, SELEX internal report, 1998.
- [28] M.Olivo, “*Producción inclusiva de Λ , $\bar{\Lambda}$ y K_S mediante las colisiones Σ^- , $\pi^\pm$ y p - Nucleón*”.
- [29] *PAW - Physics Analysis Workstation*, CERN-IT Division, Geneva, Switzerland, January 1999.
- [30] J.Sánchez, “*Polarización de Λ^0 y $\bar{\Lambda}^0$ en colisiones Σ^- , $\pi^\pm$ y p -Nucleón*”.
- [31] M.Procaro and S.Kushnirenko, “*EDG - Embedded Data Generator for SELEX*”, Research note, SELEX Collaboration,1998,
<http://home.fnal.gov/syjun/edg-doc.html>.