9-1964

Writing topological programs for I.S.U. guts

William J. Kernan
Iowa State University

William J. Higby
Iowa State University

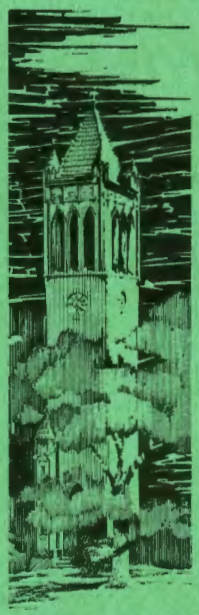
Iva H. Boessenroth
Iowa State University

Writing topological programs for I.S.U. guts

Abstract
This paper is devoted to the analysis of bubble chamber data through the use of a computer program, GUTS. This system is designed in a general manner. That is, a wide variety of interactions in bubble chambers may be studied by writing control programs describing the interaction. The techniques employed and the method of writing these programs will be presented.

Disciplines
Computer Sciences | Mathematics | Physical Sciences and Mathematics

IS-989



IOWA STATE UNIVERSITY

WRITING TOPOLOGICAL PROGRAMS FOR I. S. U. GUTS

by

William J. Kernan, William J. Higby
and Iva H. Boessenroth

AMES LABORATORY

RESEARCH AND DEVELOPMENT REPORT

U.S.A.E.C.

IS-989

ERRATA SHEET

WRITING TOPOLOGICAL PROGRAMS FOR ISU GUTS

Page 3, CONTENTS, Change page numbers:

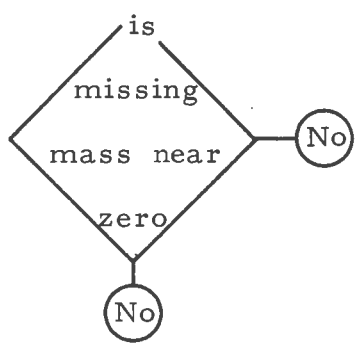
Reduction of Constraints from	28 to 29
Configuration Numbers from	30 to 31
Conclusion from	30 to 31
Appendix I from	31 to 33
Appendix III from	33 to 34

Page 13, 2nd line:

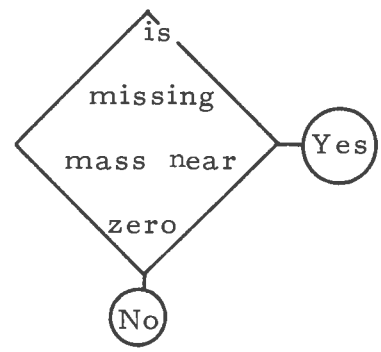
Change "CALL COPLN (I, J, K, V. DV)" to
"CALL COPLN (I, J, K, V, DV)"

Page 15, Flow Chart, Decision Block:

change



to



Page 16, line 19

Insert the FORTRAN Statement "NTYP(1B) = 8 between "NTYP(1A) = 5" and "NPART = 3"

IS-989

Mathematics and Computers(UC-32)
TID 4500, December 1, 1964

UNITED STATES ATOMIC ENERGY COMMISSION

Research and Development Report

WRITING TOPOLOGICAL PROGRAMS FOR I. S. U. GUTS

by

William J. Kernan, William J. Higby
and Iva H. Boessenroth

September, 1964

Ames Laboratory
at
Iowa State University of Science and Technology
F. H. Spedding, Director
Contract W-7405 eng-82

This report is distributed according to the category Mathematics and Computers(UC-32), as listed in TID-4500, June 1, 1964.

Printed in USA. Price $2.00 . Available from the Clearinghouse for Federal Scientific and Technical Information, National Bureau of Standards, U. S. Department of Commerce, Springfield, Virginia

CONTENTS

WRITING TOPOLOGICAL PROGRAMS FOR I. S. U. GUTS

William J. Kernan, William J. Higby and Iva H. Boessenroth

ABSTRACT

This paper is devoted to the analysis of bubble chamber data through the use of a computer program, GUTS. This system is designed in a general manner. That is, a wide variety of interactions in bubble chambers may be studied by writing control programs describing the interaction. The techniques employed and the method of writing these programs will be presented.

INTRODUCTION

Any attempt to analyze bubble chamber events necessarily involves both a reconstruction routine, to find the momentum and angles of each track involved and a kinematical fitting procedure to test various possible mass assignments or particle identifications. Many such programs have been written. One program for the kinematical analysis of events which originated at Berkeley is "GUTS". The ISU GUTS system is a rewriting of the technique used by Berkeley into a FORTRAN program.

The analysis is essentially a least squares fit of the input measured variables (azimuth, dip and momentum) for each track. This fit is done subject to constraint conditions (conservation of energy and momentum). The information that we do not have after the reconstruction is made is the mass assignment. That is, we have the tracks oriented in space by the two angles azimuth and dip. We also know the momentum for each track. Our problem is to make an hypothesis of the final state for an

event. Once an hypothesis is made, we can associate certain particles with each track and attempt a fit.

GUTS is written in a general manner, so as to be able to process data from any interaction in a bubble chamber within certain limitations. The actual definition of an event so far as the physical characteristics are concerned is done by a topological program. A top program must be written for each distinct class of events measured. For instance, a given top program may only be concerned with events that have an incoming track and two outgoing tracks. Another top program may only deal with four outgoing tracks.

This description is written to give the reader the knowledge necessary to write top programs. A complete knowledge of the GUTS system is not necessary. All the rules and procedures in this paper must be obeyed to the letter. If not, very wierd things could happen in the system.

COMMUNICATION TO GUTS

The top programs serve as a command and communication link for GUTS. They tell GUTS exactly what to do, and what material (data) to use. We assume here that the reader is fairly familiar with the physical situation involved. The only concept that may not be common knowledge is that of constraint conditions used by GUTS.

The number of constraints to be applied at any given vertex may be 0, 1, 2, 3 or 4. The number of constraints essentially informs GUTS of the number of missing variables. Consider the interaction shown in

Fig. 1. We have a simple elastic scatter, $K^- + P \rightarrow K^- + P$. The tracks are well measured, and all variables (azimuth, dip and momentum) associated with each track are known. The appropriate number of constraints for this hypothesis is 4.

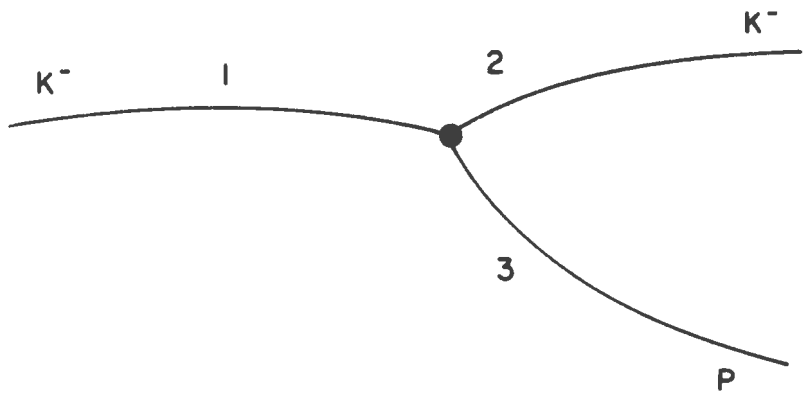


Fig. 1.

Let us further complicate this event by adding a neutral unseen particle (Fig. 2).

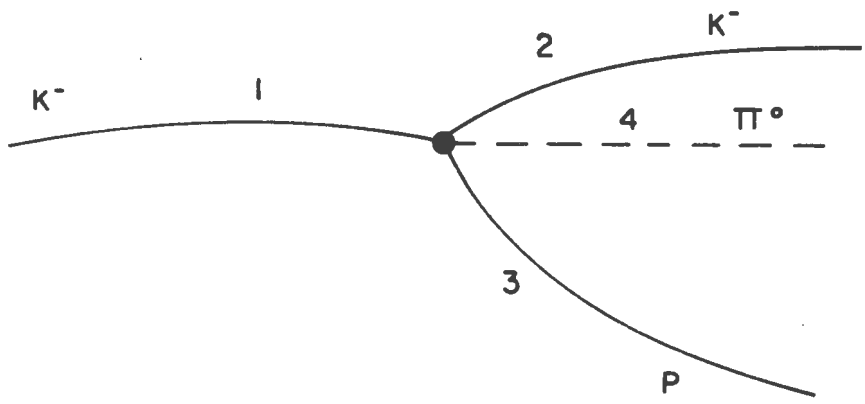


Fig. 2

The reaction is $K^- + P \rightarrow K^- + P + \pi^\circ$. The π° is unseen, therefore unmeasured. In other words, the azimuth, dip and momentum of this track are unknown. There are 3 missing variables, so our constraints will be reduced by 3, giving us a 1 constraint case.

Note that in our figures we have numbered the tracks. In this paper we will be using the term storage bank. The measured variables for track 1 will be contained in storage bank 1, track 2 is in storage bank 2, etc.

The determination of the constraint class (number of constraints) to be applied to a vertex for a given particle configuration hypothesis can be made by using the following rules.

4c. All variables for all tracks are known.

3c. The momentum of one track is unknown or poorly measured.

2c. The momenta of two tracks are unknown or poorly measured.

1c. All variables (ϕ, λ, p) for the last track are unknown.

0c. The ϕ, λ, and p of the last track are unknown, and the p of one other track is unknown or poorly measured.

All information transferred to GUTS is stored in common, so the names used in this description are identical with the names used in the top programs. Basically GUTS must have the following information in order to perform a fit of the configuration hypothesis:

1. Particles involved
2. Track numbers
3. Total number of particles
4. Number of constraints

Particles Involved

We must set up the appropriate mass assignments for the interaction we are testing for this event. The particle assignment table is given in Appendix I. The desired information will be transferred to GUTS by setting the singly subscripted FORTRAN variable NTYP(I) to the appropriate number. The proper FORTRAN statements to initialize NTYP corresponding to Fig. 1 would be:

```
NTYP(1) = 5
NTYP(2) = 5
NTYP(3) = 8.
```

For Fig. 2, the statements would be

```
NTYP(1) = 5
NTYP(2) = 5
NTYP(3) = 8
NTYP(4) = 3.
```

Track Numbers

In addition to establishing mass assignments for a given event, we must now identify the tracks that correspond to the mass hypothesis. One of the first tasks accomplished by the GUTS system is to read a tape that was previously prepared by a spacial reconstruction program. The information on this tape will consist of variables that are associated with each track of the event. These variables will then be stored in certain storage banks. That is, the variables for track 1 will be stored in storage bank 1, those for track 2 will be stored in storage bank 2, etc. Now, a decision must be made as to what variables are to be considered as

associated with a particular vertex. This is essentially asking the question, "Which storage banks contain the variables for the tracks associated with this vertex?" A singly subscripted variable, NOTRK(I), is used to communicate to GUTS which storage banks to consider. If we were to set, say NOTRK(4) = 8, this will be interpreted by GUTS as a special definition of which variables to use for the kinematical analysis. This statement means that the variables associated with the fourth track at this vertex are stored in storage bank 8.

This assignments for Fig. 1 would be

NOTRK(1) = -1

NOTRK(2) = 2

NOTRK(3) = 3.

Any time a track number is given a negative value, GUTS assumes that

this is the incoming track for that vertex. We would have the same statements for Fig. 2 with the addition of the statement

NOTRK(4) = 0 .

This statement may appear confusing at this stage, since Fig. 2 shows a π° being associated with a track numbered 4. We may ask, "Why not set NOTRK(4) = 4?" Remember that the π° is an unseen track with no measured variables. GUTS will fit this event assuming no measured variables for the π°. For this configuration, storage bank 4 will not have any stored variables.

Total Number of Particles

For each entry into GUTS, a variable NPART must be set to the number of particles for the assumed configuration. The total number is the incoming plus all outgoing. The target particle is not included in this number. For Fig. 1, the FORTRAN statement would be

NPART = 3,

and for Fig. 2

NPART = 4 .

Number of Constraints

The previous discussion of constraints should be sufficient at this time. The FORTRAN statements for Figs 1 and 2, respectively, would be

NCONST = 4

and

NCONST = 1 .

As we mentioned above, the number of constraints may also be 0, 2 and 3. NCONST will never be set to 0, 2 or 3 in a top program. The system will use a procedure to reduce constraints to these values if it is necessary. A full discussion of the constraint reduction will be given later.

Up to this point we have discussed the basic procedure necessary to initialize variables for a GUTS fit. Now, let us turn to the actual techniques which will be employed in writing a top program. We can save a considerable amount of computing time by a simple analysis of the physical situation of a given event. Depending on the circumstances involved, some final states may or may not be physically possible. It is a waste of computer time to attempt a fit for these impossible final states.

Using Fig. 1 again, two questions must be answered. Number one is, "Do these three tracks all lie in the same plane?" Only if this event is coplanar do we have the possibility of a two body final state. Subroutine COPLN will evaluate the coplanarity volume enclosed by the tracks and the associated error. The method used is to compute the scalar V as defined by

$$V = \vec{P}_1 \cdot (\vec{P}_2 \times \vec{P}_3)$$

where $\vec{P}_1$, $\vec{P}_2$ and $\vec{P}_3$ are unit vectors along the directions of tracks 1, 2 and 3, respectively. We see that if V = 0, this implies that all 3 vectors lie in the same plane. Since our calculation of V will have an associated uncertainty δV, we must take this into consideration in our test for V = 0.

The call to COPLN uses parameters in the call list as

CALL COPLN (I, J, K, V. DV) .

The first three parameters must be the track numbers (in fixed point) for which the coplanarity calculation is desired. The last parameters V and DV are the volume and error of the coplanarity calculation computed by COPLN.

Our criteria for coplanarity would then be $|V| \leq 3. \times DV$. The factor of 3. is completely arbitrary.

We would like to ask a second question concerning Fig. 1. Is a π° possible in the final state? This involves a treatment of the energy imbalance of the hypothesis of $K^- + P \rightarrow K^- + P$ for tracks 1, 2 and 3. We can use GUTS to evaluate the missing mass and uncertainty in the missing mass for a given hypothesis. The only restriction is that the hypothesis must be 4 constraint. The missing mass and its associated uncertainty are called UMASS and DMIM, respectively. To obtain these quantities the FORTRAN calling statements must be

```
MM = 1
CALL VRTFIT(MM).
```

The name of the argument of VRTFIT is arbitrary, but it must be a fixed point name.

Our next procedure is to determine if UMASS is relatively close to zero (elastic scatter), or the mass of a π°. The particle mass table is stored in a singly subscripted array SPMASS(I). The ordering of the mass assignments corresponds to Appendix I. The mass of a π° is stored in SPMASS(3). Make special note that for a fit rather than a

missing mass calculation, the procedure is

```
MM = 0
CALL VRTFIT(MM).
```

Figure 3 gives a basic flow chart of the logic involved in performing the operations to fit the hypothesis of Figs. 1 and 2. The FORTRAN statements following this figure are the program.

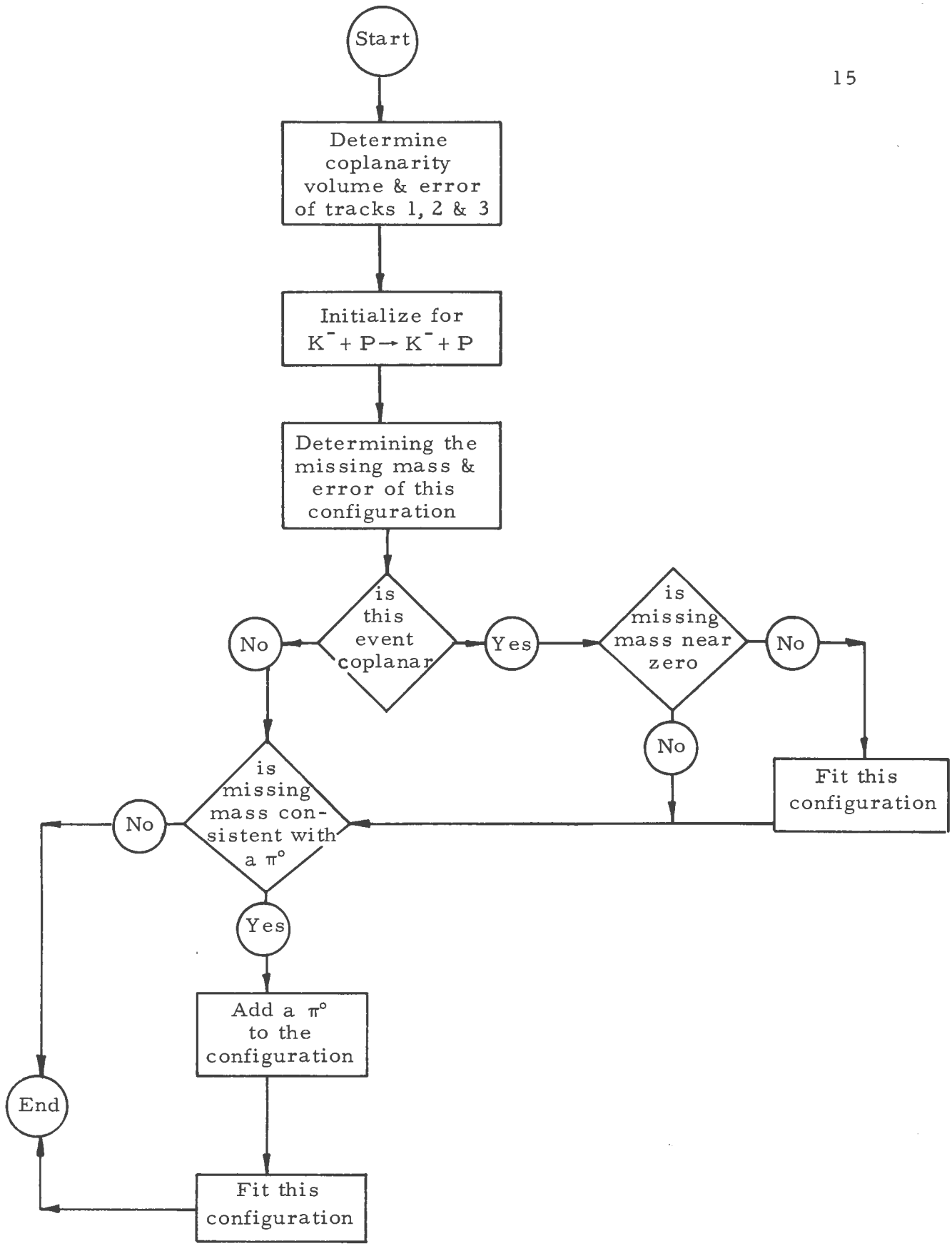

Start
Determine coplanarity volume & error of tracks 1, 2 & 3
Initialize for $K^- + P \rightarrow K^- + P$
Determining the missing mass & error of this configuration
is this event coplanar
No
Yes
is missing mass near zero
No
No
Fit this configuration
is missing mass consistent with a π°
No
Yes
Add a π° to the configuration
End
Fit this configuration

Fig. 3

```
C           EXAMPLE FORTRAN STATEMENTS FOR
C           FLOW CHART IN FIGURE 3

            IF(RCRV(2)) 5, 5, 10
C       RCRV IS THE TRACK CURVATURE, ITS SIGN
C       DETERMINES THE PARTICLE CHARGE THAT
C       IS ASSOCIATED WITH THIS TRACK

          5 IA = 2
            IB = 3
            GO TO 20
         10 IA = 3
            IB = 2

C       DETERMINE COP. VOLUME AND ERROR

         20 CALL COPLN(1, 2, 3, AN, ERAN)

C       K- PLUS PROTON INTO K - PLUS PROTON

            NOTRK(1) = -1
            NOTRK(2) = 2
            NOTRK(3) = 3
            NTYP(1) = 5
            NTYP(IA) = 5
            NPART = 3
            NCONST = 4
            MM = 1
C           MEANING OF FOLLOWING STATEMENT EXPLAINED ON pp. 20.
            IT = 8
C       DETERMINE MISSING MASS AND ERROR
            CALL VRTFIT(MM)
            SAVEM = UMASS
            SAVEE = DMIM

C       ARE THESE TRACKS COPLANAR
```

```
      IF(ABSF(AN) - 3. *ERAN) 25, 25, 30
C     IS MISSING MASS NEAR ZERO
   25 IF(ABSF(SAVEM) - 3.5 *SAVEE) 27, 27, 30
C     FIT THE ELASTIC SCATTER
   27 MM = 0
      CALL VRTFIT(MM)
C     IS MISSING MASS CLOSE TO THAT OF A PI-0
   30 IF(ABSF(SAVEM - SPMASS(3)) - 3.5 *SAVEE) 35, 35, 40
C     ADD THE PI-0
   35 NOTRK(4) = 0
      NTYP(4) = 3
      NPART = 4
      NCONST = 1
      MM = 0
C     FIT THIS CONFIGURATION
      CALL VRTFIT(MM)
   40 RETURN
      END
```

The program we have just shown is incomplete in the sense that it does not attempt to fit all the final states of this particle interaction type.

Multiple Vertex Fits

The procedures of our version of GUTS are limited to fitting one vertex at a time with a maximum of seven particles (incoming plus outgoing) associated with this vertex. It is possible to fit multiple vertex events by projecting the fit obtained from one vertex to the connecting vertex for a fit. There are two available methods for handling the mechanics of the connection of multiple vertices. One is VFIT which is discussed later. This is used to connect the decay and production vertices of a neutral track. The other method is discussed below.

Programmed Vertex Connection

Suppose we are interested in looking at two vertex events in a π^- exposure. The interest may be only to find the elastic events (i. e. $\pi^- + P \rightarrow \pi^- + P$) where the outgoing proton rescatters elastically on another proton (i. e. $P + P \rightarrow P + P$). GUTS is written in such a manner that a measured track connecting two vertices must be programmed as

the last measured track from the first vertex. The event would appear as shown in Fig. 4.

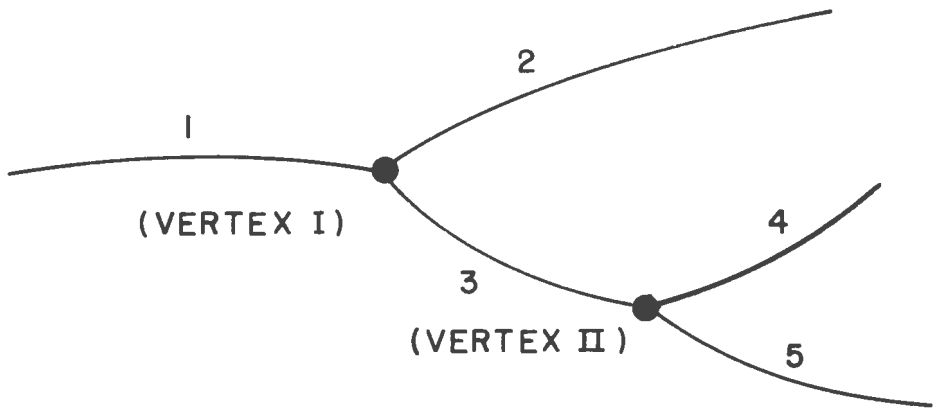


Fig. 4.

The technique is to fit vertex I, and then use the fitted variables for track 3 as input for the fit of vertex II. After a fit is made, VRTFIT will test a variable NOP. If NOP < 1, the fitted variables of the last track will be stored in storage bank 20. If NOP > 1, the fitted variables of the next to last track will be stored in storage bank 20. The latter is necessary if hypotheses are tried which have added a neutral. If NOP = 1, no change will be made in the contents of storage bank 20. To fit vertex I, we would set

NOP = 0

IT = 8

NOTRK(1) = -1

NOTRK(2) = 2

NOTRK(3) = 3

```
NTYP(1) = 2
NTYP(2) = 2
NTYP(3) = 8
NCONST = 4
NPART = 3
MM = 0
CALL VRTFIT(MM) .
```

Now we are ready to fit vertex II.

If we set NOTRK(1) = -20, the variables stored in storage bank 20 will be used for the second vertex fit. To fit vertex II, we would set

```
NOP = 1
IT = 8
NOTRK(1) = -20
NOTRK(2) = 4
NOTRK(3) = 5
NTYP(1) = 8
NTYP(2) = 8
NTYP(3) = 8
NPART = 3
NCONST = 4
MM = 0
CALL VRTFIT(MM) .
```

You will notice in the FORTRAN statements a new variable that we have not yet discussed. This variable informs GUTS of the mass of the target. In the example above, IT = 8, which is the number of the proton

mass in the mass table. For topological types that have a decay vertex, IT must be set equal to 25 for the fit of the decay. This will cause the target mass to be set equal to zero. In this example, we have shown only one final state for vertices I and II. Consider more than one possible final state at vertex II corresponding to a single final state at vertex I. For every fit attempted at vertex II, NOP must be equal to 1 before VRTFIT is entered. This will allow the variables calculated at vertex I for track 3 to be projected to vertex II for all final state fits.

GUTS will not fit all configurations that it is asked to. Some will be impossible because of energy and momentum imbalance. Non-physical situations may also occur in the fitting process. If any of these conditions arise, the fit is rejected. That is, a variable IRJCT is set to a non-zero value, a reject message is printed, and control is returned to the topological program. The reject messages are given in Appendix II. When a fit is not rejected, IRJCT will be zero. In some cases, a considerable amount of computer time may possibly be saved if IRJCT is tested after a fit is attempted. If a multiple vertex is being fitted and variables from one particular vertex fit are being projected to another vertex, all second (or higher) vertex fits will be meaningless if the initial vertex fit has been rejected.

USE OF SUBROUTINE VFIT FOR MULTIPLE VERTEX EVENTS

Another technique which is useful in fitting a multiple vertex event involves the use of the subroutine VFIT.

VFIT is a special purpose subroutine used to fit the decay of a neutral particle into two outgoing particles at a secondary vertex. It is used only for Λ°, K°, or $\overline{K}^\circ$ decays, i.e.,

(1) $\Lambda^\circ, \rightarrow P + \pi^-$

(2) $\overline{K}^\circ \rightarrow \pi^+ + \pi^-$

$K^\circ \rightarrow \pi^+ + \pi^-$

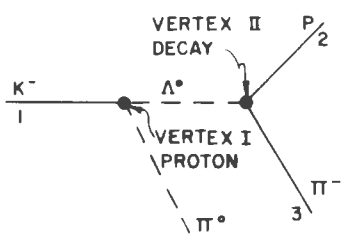
In VFIT the cases for $\overline{K}^\circ$ and K° are treated the same since they are neutral particles of the same mass with the same decay mode. The output of the present version of VFIT will always show a $\overline{K}^\circ$ decay. However, in writing the topological program for the vertex at which the $\overline{K}^\circ$ or K° particle originates, a distinction is made (i. e., $\overline{K}^\circ$ and K° have different type numbers).

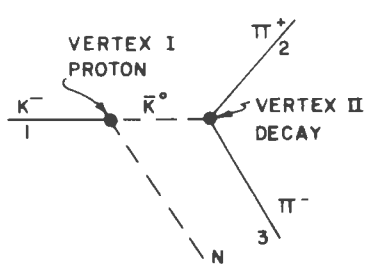
If a decay other than the ones described above occurs, then this must be taken care of in the topological program. For example VFIT could not be used for the decay $\overline{\Xi}^- \rightarrow \pi^- + \Lambda^\circ$.

The procedure is usually to use VFIT first to fit the decay vertex. After a successful fit is made, the fitted variables for the neutral connecting track can be projected to the connecting vertex for a fit.

Diagrams for some typical events requiring the use of VFIT are shown below. They will be used as illustrations in the following discussion of VFIT.

Case I - 3 Constraint Case





Case II - 1 Constraint Case

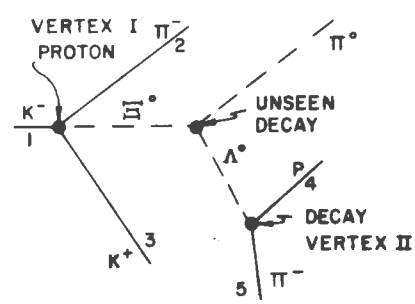


Fig. 5.

If VFIT is to be used to fit both a Λ° decay and a $\overline{K}^\circ$ (or K°) decay, the variables LFIT and KFIT are set equal to zero in the topological program before calling VFIT. If only a Λ° fit is desired, set KFIT $\neq 0$ and LFIT = 0. If only a $\overline{K}^\circ$ fit is desired, set KFIT = 0 and LFIT $\neq 0$. The variables KFIT and LFIT are in common. It is possible to make both a Λ° and $\overline{K}^\circ$ fit by calling VFIT only once. This is due to a special property of the subroutine, VRTFIT (which VFIT calls). For a Λ°, VRTFIT stores the fitted variables for this track in storage bank 16. Likewise the variables for $\overline{K}^\circ$ (or K°) will be stored in storage bank 17.

It might also be noted at this point that VRTFIT also stores the fitted variables for Ξ° in storage bank 18. Thus Λ°, $\overline{K}^\circ$ (or K°) and Ξ° are the only variables that have special storage schemes in VRTFIT.

The subroutine, VFIT, is called by the FORTRAN statement CALL VFIT (IA, IB, IC, MUMC). When VFIT is called by this statement, four arguments, IA, IB, IC and MUMC, are transferred from the topological program to the subroutine, VFIT. The argument, MUMC, informs VFIT if (I) the case is a three constraint case and the coplanarity test is to be used or (II) if it is a one constraint case and the coplanarity test is not to be used. For the former, Case I (also Case I in Fig. 5) MUMC = 0. For the latter, Case II (also Case II in Fig. 5), MUMC > 0.

IC is the incoming decay track number, and IA and IB are the track numbers of the outgoing particles at the decay vertex. Referring to Fig. 5 again, for Case I the track numbers IA and IB would be 2 and 3 respectively and 4 and 5 for Case II. The track number, IC, is positive (VFIT sets it negative to indicate an incoming particle).

The choice of the track number, IC, is less obvious, and the following discussion is intended to clarify it. If a 3 constraint case is involved (Case I), the variables, azimuth and dip for the neutral decay track connecting the two vertices, have been previously calculated and stored by the subroutine, FILL. The track number for IC would be chosen from the chart below:

Storage track no.	Line of flight between vertices
11	vertex II to vertex I
12	vertex III to vertex I
13	vertex IV to vertex I
14	vertex III to vertex II
15	vertex IV to vertex II
Not in present program	vertex IV to vertex III

Since in Case I, the line of flight is between vertex II and vertex I, we would choose storage bank 11 and set IC = 11.

If a 1 constraint case is involved, the angles have not been calculated previously, and it really does not matter what track number is used. As long as it is greater than zero.

After calling VFIT and before going to the primary vertex for a fit (in our cases vertex I), a test can be made to see if a Λ° or $\overline{K}^\circ$ (or K°) fit (or both) was successful in VFIT. If a Λ° fit was made, LFIT is set equal to one in VFIT. For a successful $\overline{K}^\circ$ fit, KFIT = 1. Therefore LFIT and KFIT can be tested for this information.

To summarize, the following FORTRAN steps might occur in a topological program calling VFIT for the cases being discussed.

Case I

NOP = 1 (see previous discussion on NOP)

KFIT = 0

LFIT = 0

MUMC = 0

IA = 2

IB = 3

IC = 11

CALL VFIT (IA, IB, IC, MUMC)

IF (LFIT) 2, 2, 1

1. Procedure to fit vertex I with $K^- + P \rightarrow \pi^\circ + \Lambda^\circ$ remembering that Λ° information is stored in bank 16.

2. IF (KFIT) 4, 4, 3.

3. Procedure to fit vertex I with $K^- + P \rightarrow N + K^\circ$ remembering that $\overline{K}^\circ$ information is stored in bank 17.

4. Procedure to continue or analyze for reject No. 80 or 90 - see last part of discussion.

Case II

IRJCT = 0

NOP = 1

KFIT = 1

LFIT = 0

MUMC = 1

IA = 4

IB = 5

IC = 11

CALL VFIT (IA, IB, IC, MUMC)

IF (LFIT) 2, 2, 1

1. Procedure to fit vertex I with $K^- + P \rightarrow \pi^- + K^+ + \Xi^\circ$ (Ξ° information is stored in bank 18); then to fit unseen decay $\Xi^\circ \rightarrow \pi^\circ + \Lambda^\circ$ remembering that Λ° information is stored in bank 16.

2. Procedure to continue or analyze for reject 80 or 90.

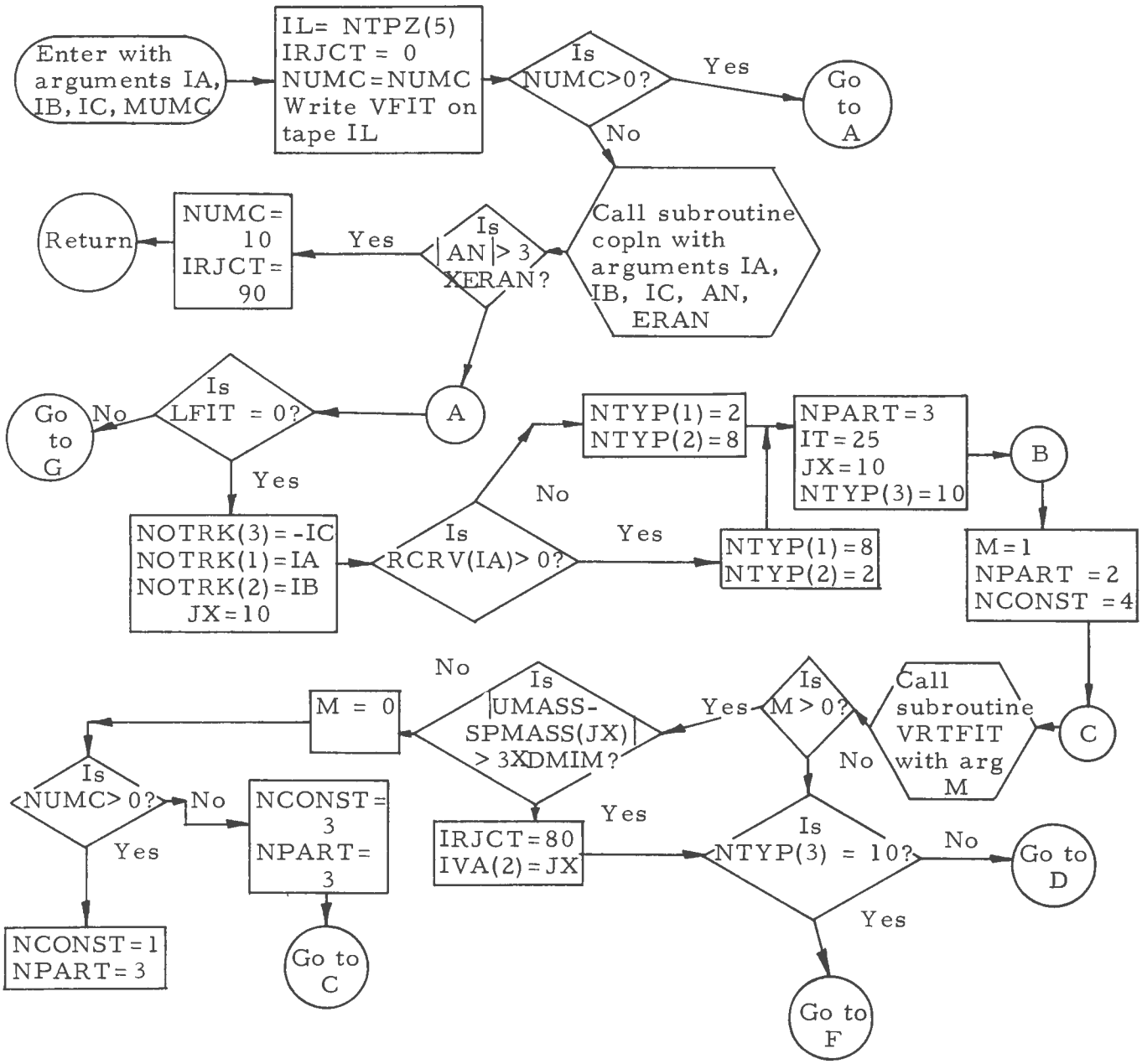
A message will be printed at the beginning of VFIT indicating that VFIT is being used.

If a 3 constraint case is being used and the coplanarity test fails, a reject message (90) will be printed as notification that a three body final state is involved and VFIT cannot be used. VFIT then returns to the topological program.

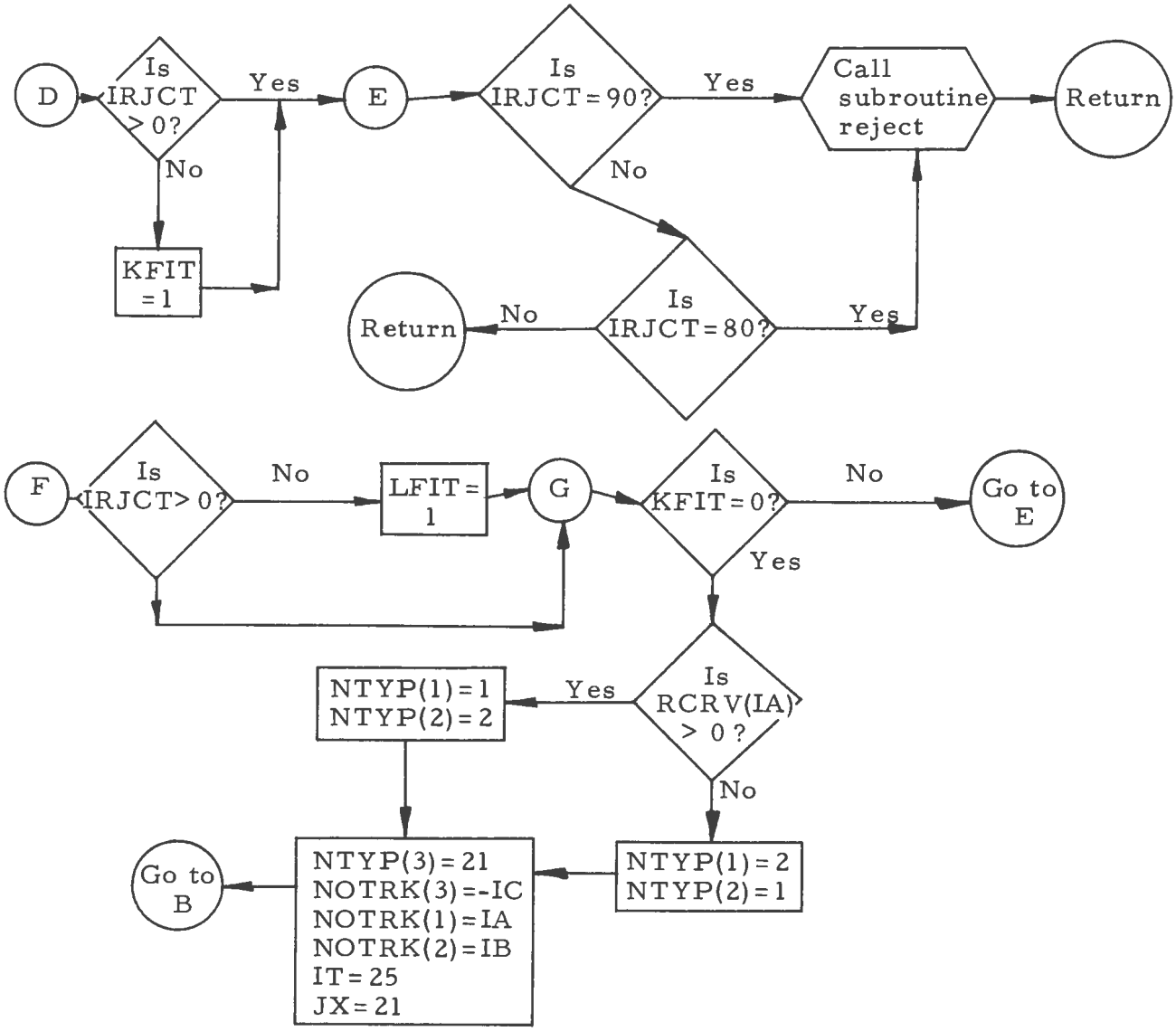
If the missing mass test fails in VFIT for a Λ° or $\overline{K}^\circ$, a reject message (80), specifying which particle is involved, will be printed as notification. Afterwards in the case of Λ°, VFIT skips the Λ° fit and tries the $\overline{K}^\circ$ fit if requested. If the missing mass test for the $\overline{K}^\circ$ fails, VFIT returns to the topological program.

Therefore, if LFIT or KFIT (or both) = 0 after calling VFIT, it would be wise to check for rejects 80 or 90 in the topological program. This is not done in many of the present K^- topological programs. The following flowchart shows the basic logic of subroutine VFIT.

FLOW CHART FOR SUBROUTINE VFIT



FLOW CHART FOR SUBROUTINE VFIT (Cont.)



Reduction of Constraints

The constraint reduction feature of GUTS is a feature that the reader should be aware of. If an input momentum is less than the associated error*, the constraints will be reduced by VRTFIT. This will cause trouble if the topological program does not take this into account when fitting a 1 constraint vertex. When VRTFIT is called with NCONST = 1 it will return with NCONST = 1 even though the number of constraints may have been reduced to zero for the fit. In the zero constraint fit, the mathematics in the computation of the curvature of the next to last particle involves a quadratic solution. To be exhaustive in the treatment of this case, we should fit the particular hypothesis for this case with both signs attached to the radical. When VRTFIT is called with MM = 0 for the 0 constraint fit, a fit will be made using a positive value of the radical. Upon exit from VRTFIT, MM will be set to -3. VRTFIT will also print out DAMN = + 1, to denote that the positive solution of the quadratic was used for this fit. If VRTFIT is entered with MM = -3, the negative solution of the quadratic will be used for the fit. Before control is returned to the calling program, MM will be set to zero. DAMN = -1. will also be printed. The logic to be applied in the top program to get both fits could be the following:

```
      MM = 0
10    CALL VRTFIT (MM)
      IF (MM + 3) 11, 10, 11
11    CONTINUE
```

If this example is confusing, Fig. 6 is a flow diagram of the logic of these statements and the critical steps of VRTFIT which handle this situation.

*Actually the value of $\delta P/P$ for which the constraint reduction takes place can be set by the experimenter in the subroutine VRTFIT.

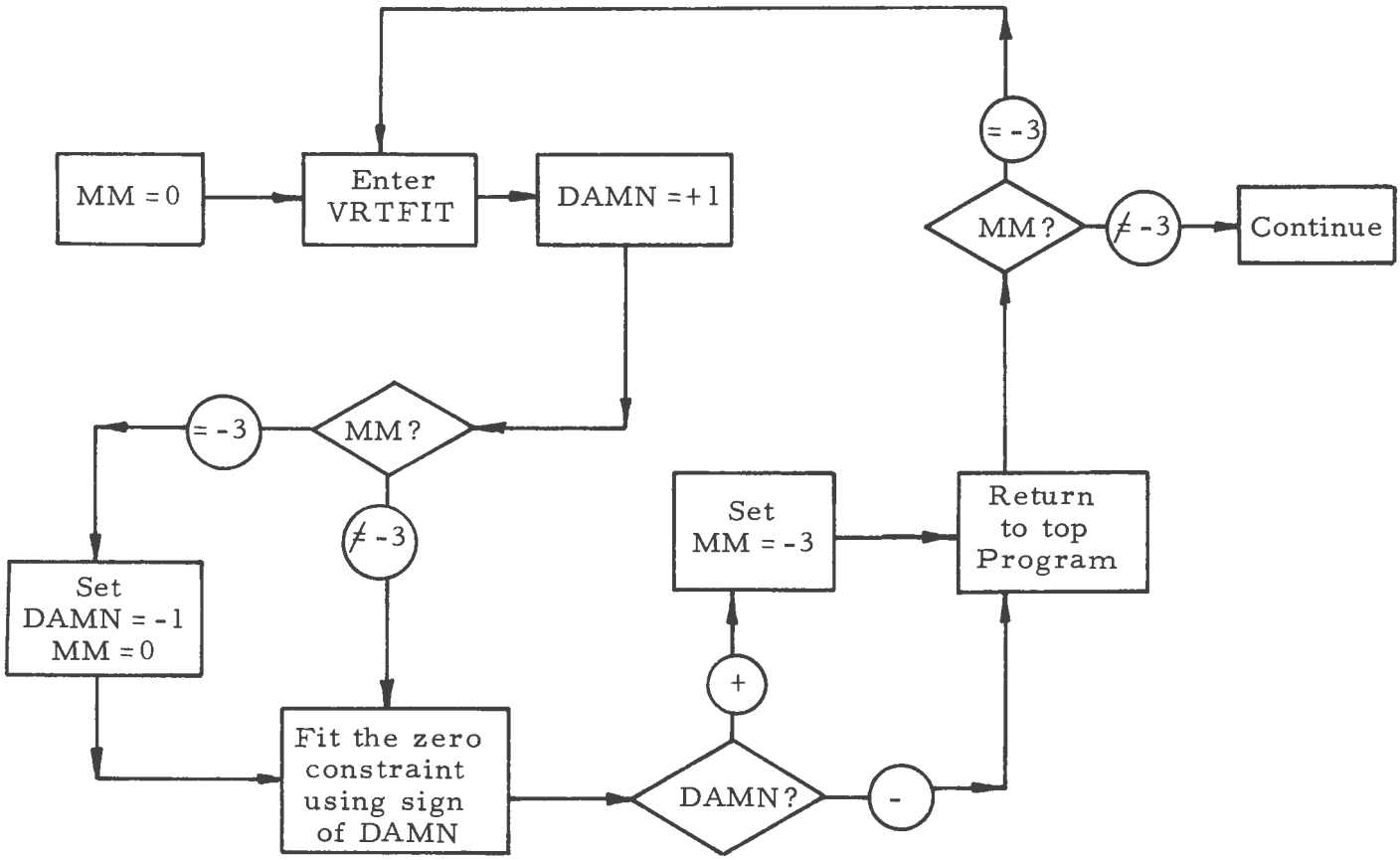


Fig. 6.

(Note: This flow chart is only an abbreviated version of the logic of VRTFIT. The logic that we have shown is true only if the constraints are reduced to zero.)

Configuration Numbers

A bookkeeping option is available that will allow an identification number to be printed and also written on the GUTS file. This is called the particle configuration number. It is used to uniquely identify the final state of a fit. To use this option, the top program must set a variable IFIG1 equal to the identification number desired.

Conclusion

If this report manages to accomplish its purpose, it should be possible for a physicist to write down the interactions of interest to him; hand that plus this report to a programmer who is not an expert on high energy physics and receive back from the programmer a correct and workable topological program. Hopefully the report is clear enough that this will actually work.

The authors are grateful to many people who have worked on this problem before. Particular thanks are due to Prof. R. K. Adair, of Yale University and Dr. L. Leipuner of Brookhaven who did the original versions of the top programs used in Appendix III and to Dr. D. Barge of Brookhaven who worked on later versions of these top programs.

APPENDIX II - Reject Numbers and Messages

Reject No.	Message
1	Energy or momentum negative
2	Too many cutdowns
3	CHI^2 too large on first step
4	Too many iterations
5	HLM singular
6	MPRIME not equal to GMFCN
7	GYFCN imaginary
8	FNORM imaginary
9	Missing mass of V imaginary
10	CHI^2 large after 15 steps
11	GYFCN singular
12	Divide check
14	GINA singular
15	Imaginary momentum in zero constraint
16	Error in missing mass imaginary
17	Negative momentum or energy in zero constraint
18	A diagonal element of GINA1 is less than or equal to zero
19	Error in momentum is imarinary
20	CHI^2 too large, CHI^2 = _____
21	Negative curvature on initial step, two constraint case
30	Poor momentum neasurement
80	Missing mass for type _____ (no.) particle rejected in VFIT
90	VFIT, three body decay
111	Zero particles in VRTFIT

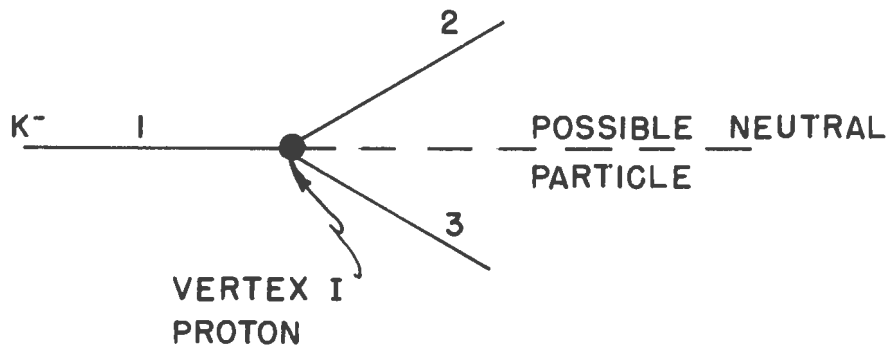
APPENDIX I

Assignment No.	Particle
1	π^+
2	π^-
3	π°
4	K^+
5	K^-
6	K°
7	γ
8	P
9	N°
10	Λ°
11	Σ^+
12	Σ^-
13	Σ°
14	$\bar{\Xi}^-$
15	$\bar{\Xi}^\circ$
16	e^+
17	e^-
18	μ^+
19	μ^-
20	ν
21	$\overline{K^\circ}$
26	Ω^-

APPENDIX III

K-Experiment

TOP 1



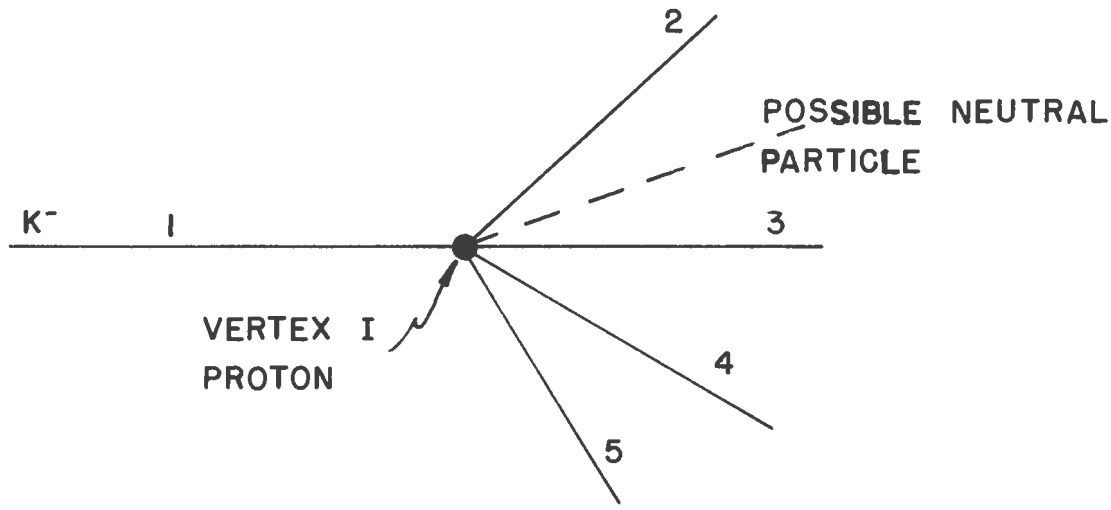
Vertex I - Interaction

Particle Conf. No.	Track		Target		Track 3		Track 2		Neutral Track
201(302)	K^-	+	P	→	P	+	K^-	+	(π°)
303				→	π^+	+	K^-	+	N
301				→	P	+	π^-	+	$\overline{K}^\circ$
305				→	π^+	+	π^-	+	Λ°
319				→	K^+	+	π^-	+	Ξ°
313				→	π^+	+	π^-	+	Σ°
315				→	K^+	+	K^-	+	Σ°
307				→	K^+	+	K^-	+	Λ°

Notation: Parenthesis around last particle indicates that it may or may not occur. If it does occur, its particle configuration number is also in parenthesis.

K^- Experiment

TOP 2



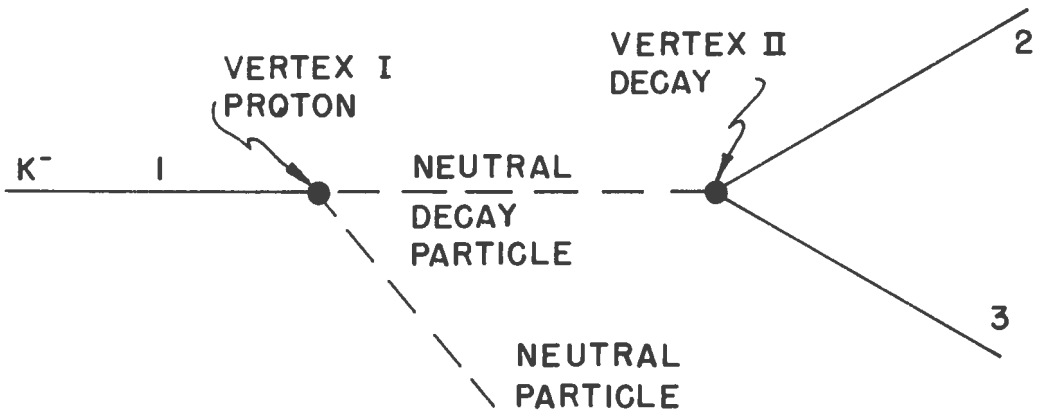
Vertex I - Interaction

Particle Conf. No.	Track 1		Target		Track 2		Track 3		Track 4		Track 5		Neutral Track
401(502)	K^-	+	P	→	P	+	π^+	+	K^-	+	π^-	+	(π°)
522				→	K^+	+	π^+	+	π^-	+	π^-	+	Ξ°
516				→	π^+	+	π^+	+	π^-	+	π^-	+	Σ°
509				→	π^+	+	π^+	+	π^-	+	π^-	+	Λ°
501				→	P	+	π^+	+	π^-	+	π^-	+	$\overline{K}^\circ$
505				→	π^+	+	π^+	+	K^-	+	π^-	+	N

Note: For any tracks of same charge, non-identical masses - the mass assignments are permuted and tried again.

K^- Experiment

TOP 3



Vertex I - Interaction

Particle Conf. No.	Track 1		Target		Neutral Track		Neutral Decay track
203	K^-	+	P	→	π°	+	Λ°
202				→	N	+	$\overline{K}^\circ$
208				→	Ξ°	+	K°

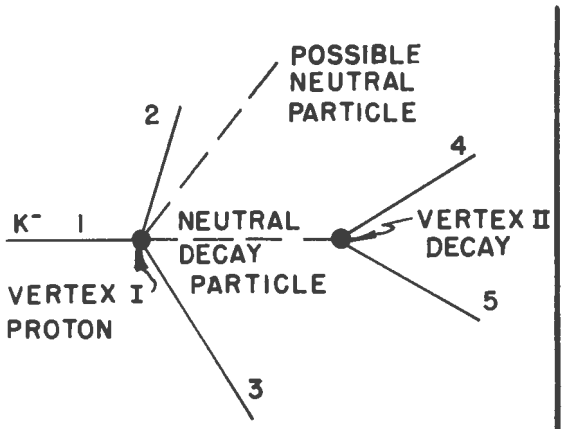
Vertex II - Decay

Vertex I Particle Conf. No.	Neutral Decay Track		Track 2		Track 3
203	Λ°	→	P	+	π^-
202	$\overline{K}^\circ$	→	π^+	+	π^-
208	K°	→	π^+	+	π^-

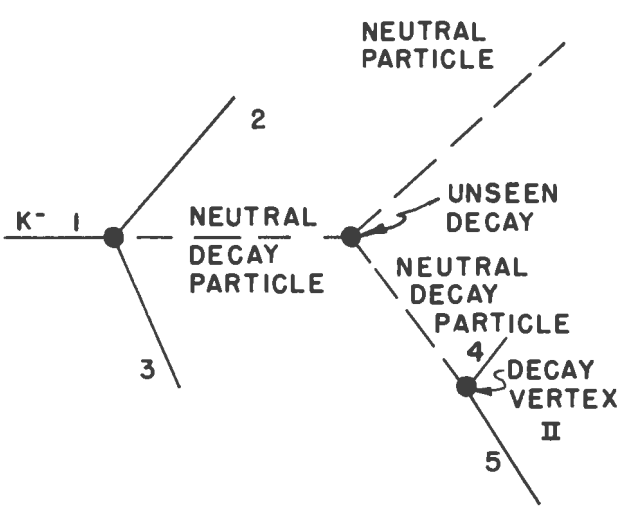
K^- Experiment

TOP 4

Case I - Regular Decay



Case II - Unseen Decay



Vertex I - Interaction (Case I and II)

Particle Conf. No.	Track 1		Target		Track 2		Track 3		Neutral Decay Track		Possible Neutral Track	Comments
305(407)	K^-	+	P	→	π^-	+	π^+	+	Λ°	+	(π°)	Case I
307				→	K^+	+	K^-	+	Λ°			Case I
301(402)				→	P	+	π^-	+	$\overline{K}^\circ$	+	(π°)	Case I
404				→	π^+	+	π^-	+	$\overline{K}^\circ$	+	N	Case I
313				→	π^-	+	π^+	+	Σ°			Case II
315				→	K^+	+	K^-	+	Σ°			Case II
319				→	π^-	+	K^+	+	Ξ°			Case II

K^- Experiment

TOP 4 (Cont.)

Unseen Decay - Case II Only

Vertex I Part. Conf. No.	Decay Particle		Neutral Track		Neutral Decay Track	Comments
313 or 315	Σ°	→	Photon	+	Λ°	Σ° Decay length is so short, Λ° points to 1st vertex.
319	Ξ°	→	π°	+	Λ°	

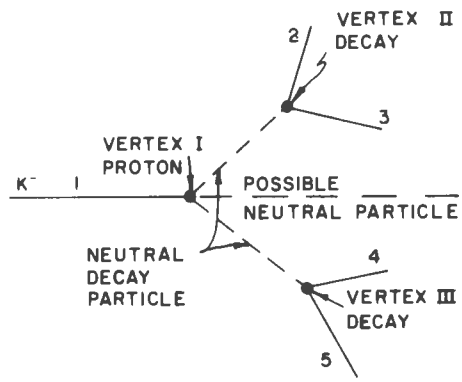
Vertex II - Decay

Part. Conf. No.					
305(407), 313, 307, 315, 319	Λ°	→	P	+	π^-
301(402), 404	$\overline{K}^\circ$	→	π^+	+	π^-

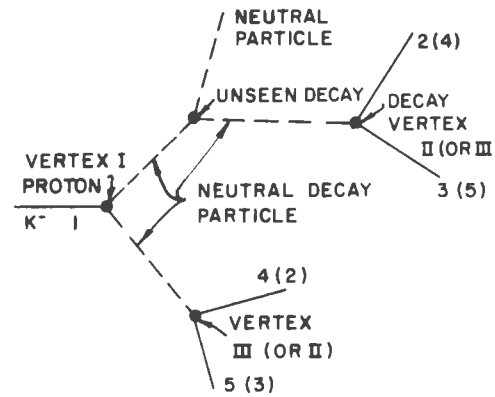
K^- Experiment

TOP 5

Case I



Case II



Vertex I - Interaction

Particle Conf. No.	Track 1		Target		Neutral Decay Track		2nd Neutral Decay Track		Possible Neutral Track	Comments
208	K^-	+	P	→	K°	+	Ξ°			Case II
308	K^-	+	P	→	Λ°	+	K°	+	$\overline{K}^\circ$	Case I These are not in present TOP 5
				or	Λ°	+	$\overline{K}^\circ$	+	K°	
				or	K°	+	$\overline{K}^\circ$	+	Λ°	

Unseen Decay - Case II

Vertex I Part. Conf. No.	Decay Particle		Neutral Track		Neutral Decay Track	Comments
208	$\overline{\Xi}^\circ$	→	π°	+	Λ°	Case II for TOP 5

K^- Experiment

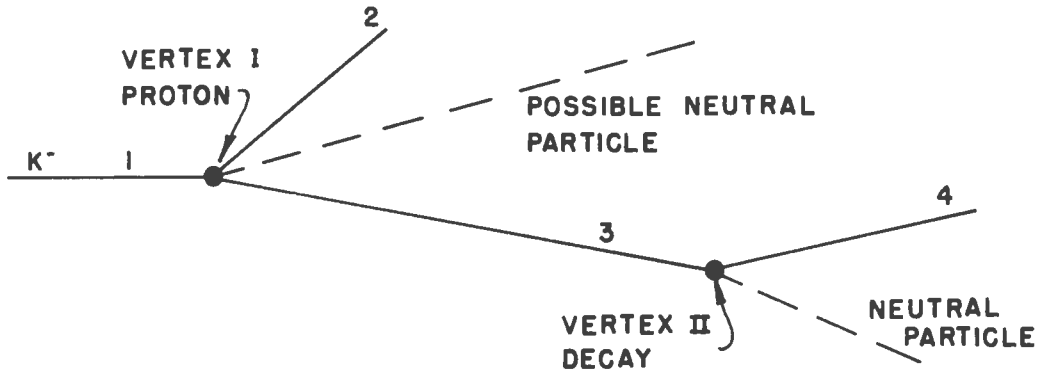
TOP 5 (Cont.)

Vertex II - Decay - Vertex III

Vertex I Part. Conf. No.	Decay Particle		Track 2 or 4		Track 3 or 5	Comments
208	Λ°	$\rightarrow$	P	+	π^-	Case II - If vertex 2 is Λ° decay then use tracks 2 & 3. If at vertex 3 use tracks 4 or 5.
	K°	$\rightarrow$	π^+	+	π^-	
308	Λ°	$\rightarrow$	P	+	π^-	Case I - Not in present program. Any 2 of the 3 cases give vertices 2 and 3.
	K°	$\rightarrow$	π^+	+	π^-	
	$\overline{K}^\circ$	$\rightarrow$	π^+	+	π^-	

K^- Experiment

TOP 6



Vertex I - Interaction

Conf. No.	Track 1		Target		(Decay) Track		Track 2		Possible Neutral Track	Comments
205(311)	K^-	+	P	→	Σ^-	+	π^+	+	(π°)	
204(309)				→	Σ^+	+	π^-	+	(π°)	
207(318)				→	Ξ^-	+	K^+	+	(π°)	
317				→	Ξ^-	+	π^+	+	K°	
310				→	Σ^+	+	K^-	+	K°	
312				→	Σ^-	+	K^+	+	$\overline{K}^\circ$	
201(302)				→	K^-	+	P	+	(π°)	- not in program

K^- Experiment

TOP 6 (Cont.)

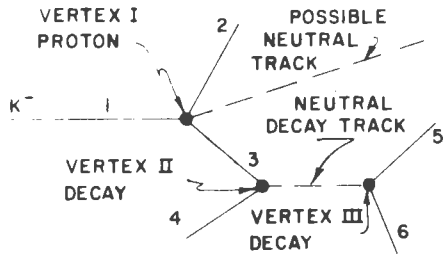
Vertex II - Decay

Vertex I Part. Conf. No.	Decay Track 3		Track 4		Neutral Track	Comments
204(309) or 310	Σ^+	$\rightarrow$	P	+	π°	
	Σ^+	$\rightarrow$	π^+	+	N	
205(311) or 312	Σ^-	$\rightarrow$	π^-	+	N	
	Ξ^-	$\rightarrow$	π^-	+	Λ°	
201(302)	K^-	$\rightarrow$	μ^-	+	ν	- not in present TOP 6
	K^-	$\rightarrow$	π^-	+	π°	- not in present TOP 6

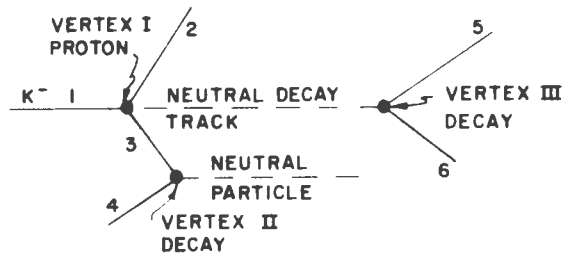
K^- Experiment

TOP 7

Case I



Case II



Vertex I - Interaction

Conf. No.	Track 1		Target		Track 2		Decay Track 3		Neutral Track	Comments
207(318)	K^-	+	P	→	K^+	+	Ξ^-	+	(π^o)	Case I
317				→	π^+	+	Ξ^-	+	K^{o*}	Case I or II
310				→	K^-	+	Σ^+	+	K^{o*}	Case II (Not in present TOP 7)
312				→	K^+	+	Σ^-	+	$\overline{K}^{o*}$	

*Can be decay track in case 2.

K^- Experiment

TOP 7 (Cont.)

Vertex II - Decay

Particle Part. Conf. No.	Decay Track 3		Track 4		Neutral Track	Comments
207(318) or 317	Ξ^-	→	π^-	+	Λ°	In case 1, Λ° is a decay track
310	Σ^+	→	P	+	π°	Case 2 (Not in present TOP 7)
	Σ^+	→	N	+	π^+	
312	Σ^-	→	N	+	π^-	

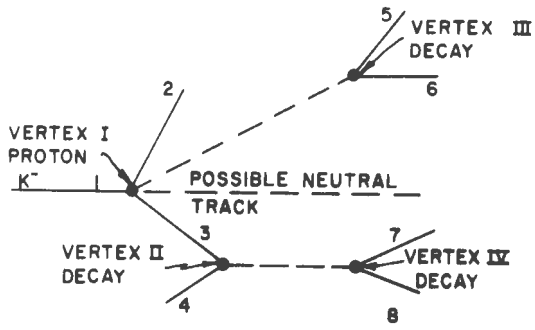
Vertex III - Decay

Particle Part. Conf. No.	Decay Track		Track 5		Track 6	Comments
207(318)	Λ°	→	P	+	π^-	Case I
317 or 310	K°	→	π^+	+	π^-	Case II(310 & 312 not
312	$\overline{K}^\circ$	→	π^+	+	π^-	Case II in present TOP 7)

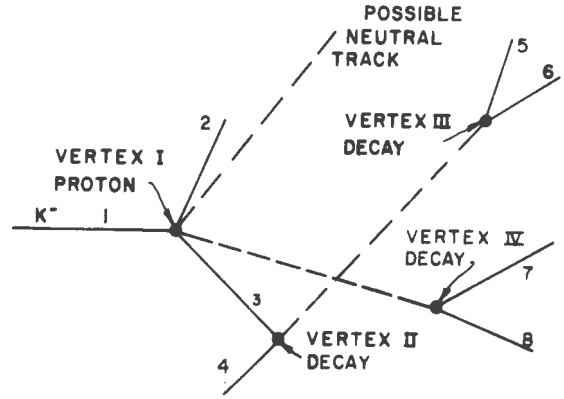
K^- Experiment

TOP 8

Case I



Case II



Vertex I - Interaction

Particle Conf. No.	Track 1	Target		Track 2		Track 3	Neutral Decay Track	Possible Neutral Track	Comments
317(416)	K^-	+ P	→	π^+	+	Ξ^-	+ K°	+ (π°)	π° not added in present TOP 8

Vertex II - Decay

Vertex I Part. Conf. No.	Decay Track		Track 4		Neutral Decay Track
317(416)	Ξ^-	→	π^-	+	Λ°

K^- Experiment

TOP 8 (Cont.)

Vertex III - Decay

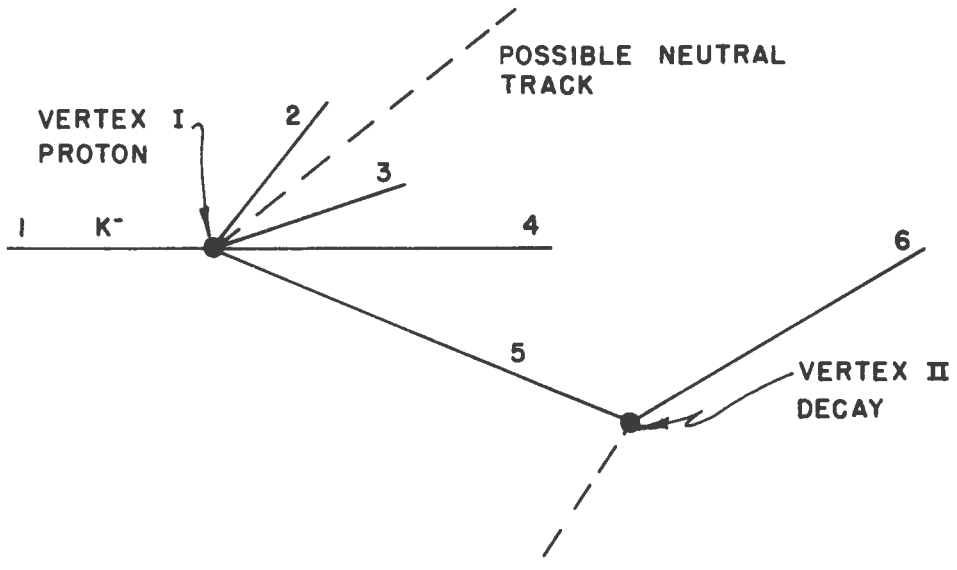
Vertex I Part. Conf. No.	Decay Track		Track 5		Track 6	Comments
317(416)	K°	$\rightarrow$	π^+	+	π^-	Case I, line of flight 1 to 3
	Λ°	$\rightarrow$	P	+	π^-	Case II, line of flight 2 to 3

Vertex IV - Decay

Part. Conf. No.						
317(416)	Λ°	$\rightarrow$	P	+	π^-	Case I, line of flight 2 to 4
	K°	$\rightarrow$	π^+	+	π^-	Case II, line of flight 1 to 4

K^- Experiment

TOP 9



Vertex I - Interaction

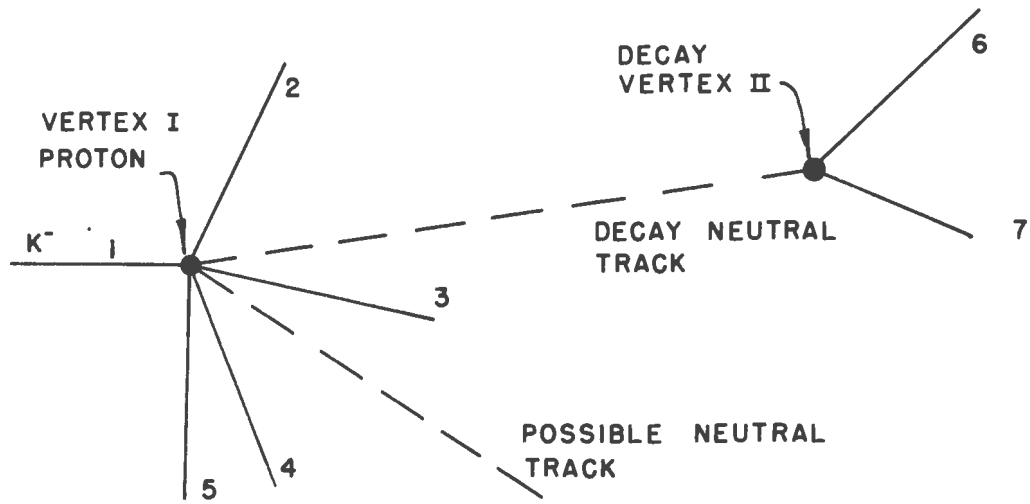
Particle Conf. No.	Track 1		Target		Track 2		Track 3		Track 4		Track 5		Possible Neutral Track	Comments
411(514)	K^-	+	P	→	π^+	+	π^-	+	π^+	+	Σ^-	+	(π°)	π° not added in present TOP 9
409(512)				→	π^+	+	π^-	+	π^-	+	Σ^+	+	(π°)	

Vertex II - Decay

Vertex II Part. Conf. No.	Decay Track		Track 6		Neutral Track
411(514)	Σ^-	→	π^-	+	N
409(512)	Σ^+	→	π^+	+	N
	Σ^+	→	P	+	π°

K^- Experiment

TOP 10



Vertex I - Interaction

Particle Conf. No.	Track 1	Target	Track 2	Track 3	Track 4	Track 5	Decay Track	Possible Neutral Track	Comments
501(602)	K^- +	P →	P +	π^+ +	π^- +	π^- +	$\overline{K}^\circ$ +	(π°)	π° not added in present
509(611)		→	π^+ +	π^+ +	π^- +	π^- +	Λ° +	(π°)	TOP 10
516		→	π^+ +	π^+ +	π^- +	π^- +	Σ°		This case not in present TOP 10

Vertex II - Decay

Vertex I Part. Conf. No.	Decay Particle		Track 6		Track 7
501(602)	$\overline{K}^\circ$	→	π^+	+	π^-
509(611) or 516(620)	Λ°	→	P	+	π^-

Note: *In the case of 516(620) there is an unseen decay $\Sigma^\circ \rightarrow \Lambda^\circ$ + Photon between Vertex I and II. This is not in present program.