

Ц 8406

35 15/11-1

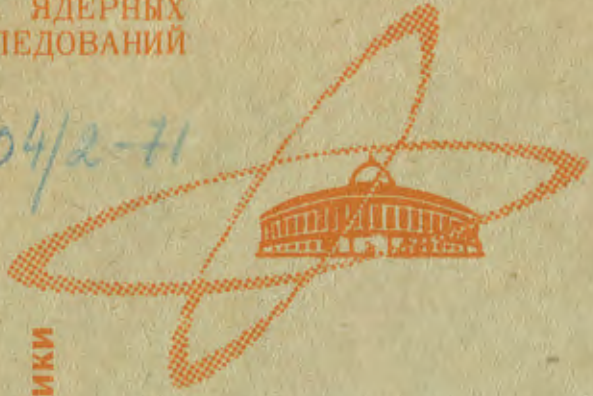
К-71

СООБЩЕНИЯ  
ОБЪЕДИНЕННОГО  
ИНСТИТУТА  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ

Дубна

10 - 5574

804/2-71



ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ  
И АВТОМАТИЗАЦИИ

З.М. Косарева, Л.А. Кулюкина, Г.А. Ососков ,  
Г.А. Эрлихман

МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ  
СКАНИРУЮЩЕГО АВТОМАТА  
"СПИРАЛЬНЫЙ ИЗМЕРИТЕЛЬ"

I. Программы фильтрации данных  
спирального сканирования

1971

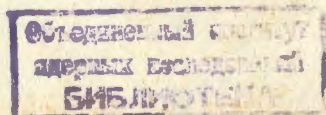
4200

10 - 5574

З.М. Косарева, Л.А. Кулюкина, Г.А. Ососков ,  
Г.А. Эрлихман

**МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ  
СКАНИРУЮЩЕГО АВТОМАТА  
"СПИРАЛЬНЫЙ ИЗМЕРИТЕЛЬ"**

**I. Программы фильтрации данных  
спирального сканирования**





Программное обеспечение автомата "Спиральный измеритель" (СИ)-  
полуавтоматического устройства для измерения снимков с пузырьковых  
камер со спиральной разверткой /1/ - состоит из двух основных  
частей:

- управляющих и тестовых программ для малой ЭВМ, работающей  
на линии с СИ;
- комплекса программы последующей обработки данных сканирова-  
ния на большой ЭВМ.

В настоящей работе, являющейся продолжением и развитием /3/,  
дано описание программ второй части матобеспечения СИ. Сюда входят  
программы, объединенные общим названием *FILTER* и осуществляющие  
отсев полезной информации от шумовой и формирование данных для  
использования их в геометрической программе *THRESH*.

Ниже приводится описание программы *FILTER* в порядке сле-  
дования основных подпрограмм, ее составляющих.

Программа *CALIB R*, также относящаяся к программам обработки  
данных спирального сканирования и осуществляющая определение калиб-  
ровочных констант СИ, будет описана отдельно

### Программа подготовки данных, *PREFIL*

Потребность в программе *PREFIL* (*PREPARATION FOR FILTER*)  
обусловлена различием в разрядности между малой управляющей ЭВМ,  
имеющей 12-разрядные слова, и большой 48-разрядной ЭВМ БЭСМ-6,  
на которой проводится обработка данных СИ, а также тем, что управ-  
ляющая ЭВМ в процессе фиксирования результатов измерений не имеет  
времени для какого-либо рода их препарации. Осуществление этих  
функций перекодировки и подготовки данных к счету естественно пе-  
редать большой ЭВМ, что и сделано в *PREFIL*.

Данные, поступающие с СИ, состоят из сведений об измеренном событии (декартовых координат вершины и четырех реперных крестов, номеров эксперимента, фильма, кадра и т.д.) и результатов спирального сканирования, содержащих следующую информацию о каждой точке пересечения сканирующей щели с изображением трека (или шумового образования):

- 1)  $R$  - радиус данной точки в полярной системе с началом координат, совпадающим с центром спирали;
- 2)  $\theta$  - угловую координату в той же системе координат;
- 3)  $H$  - амплитуду импульса, поступающего с фотумножителя в момент пересечения СИ элемента трека;
- 4)  $\Delta\theta$  - ширину этого импульса.

Последние две величины характеризуют "яркость" или, вернее, степень почернения трека на фотопленке и используются для определения ионизации трека.

В настоящее время конструкцией СИ/1/ фиксация и выдача четвертой характеристики -  $\Delta\theta$  не предусматривается.

Величины  $R$ ,  $\theta$ ,  $H$  выдаются в виде целочисленных показаний соответствующих отсчетных систем, занимая, соответственно,  $R$  - 15 двоичных разрядов,  $\theta$  - 17 и  $H$  - 3 двоичных разряда.

Кроме этого, конструкцией СИ предусмотрено генерирование единицы в разряде "признак опорной метки" в моменты пересечения сканирующей щелью опорной линии, проходящей радиально под углом  $\pi$ , с одновременным считыванием координат  $R$  и  $\theta$ . Все эти данные занимают в сумме 36 двоичных разрядов, что составляет 3 слова управляющей ЭВМ.

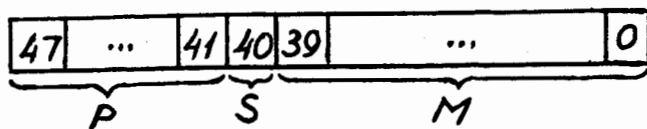
Результаты измерений декартовых координат вершины, а также измеряемых по необходимости вспомогательных точек: конечных ( $ept$ ), опорных ( $spt$ ), которые ставятся оператором на треках, пересекаемых под малым углом треком пучка, и точек излома треков ( $Spt$ ) - занимают 18 двоичных разрядов на координату  $X$ , 17 - на  $Y$ , что вместе с признаками, идентифицирующими точку, составляет по четыре 12-разрядных слова на каждую точку.

При описании общего массива данных, поступающих с СИ, следует также учесть размер буферов в памяти управляющей ЭВМ, необходимых для осуществления динамической буферизации при записи на магнитную ленту или другой внешний накопитель. Если принять размер буфера равным 900 слов (это достаточно для записи данных о 300 точках), то получаем следующую структуру массива данных: массив будет состоять из блоков по 920 12-разрядных слов, что равно 230 48-разрядным словам БЭСМ-6.

Каждый блок начинается шапкой из 240 разрядов, содержащей служебную информацию (номер эксперимента, фильма, кадра и т.д.), а также специальный "код браковки", и состоит из 300 36-разрядных троек слов малой ЭВМ, описывающих точки при спиральном сканировании. Код браковки используется оператором, чтобы аннулировать результат неправильных измерений. В нормальном случае он равен нулю, но в случаях ошибки, вовремя замеченной оператором, по специальной команде с телетайпа может быть установлен в 1. Предпоследний из блоков в группе, относящейся к данному спиральному скану, может оказаться неполным (меньше 230 слов БЭСМ-6). Группа завершается коротким блоком, содержащим, помимо обычной шапки, признак того, что этот блок последний в группе, число просканированных точек, а также декартовы координаты вершины, четырех реперных крестов и вспомогательных точек ( $spt$ ,  $spt$ ,  $ept$ ) с их признаками.  $\longrightarrow$

Первая из подпрограмм программы  $PREFIL-INPUT 1$  поочередно считывает с магнитной ленты физические рекорды, содержащие упомянутые блоки, и анализирует первые 5 слов БЭСМ-6, вмещающие 240 разрядов шапки, для определения, во-первых, не равен ли 1 код браковки, и во-вторых, не является ли этот блок последним в группе. Если код браковки равен 0 и блок не является последним, то он запоминается в рабочем массиве  $WORK$ , и программа  $INPUT 1$  вводит следующий по порядку физический рекорд. Если код браковки равен 1, то независимо от признака последнего блока, вводится следующий.

Наконец, если код браковки равен 0, а блок последний в группе, то  $INPUT 1$  передает управление программе  $UNPACK$ , превращающей тройки слов БЭСМ-6, содержащие 12 слов малой ЭВМ, в четыре слова БЭСМ-6 с данными об одной точке в каждом, расположенными в разрядах с 4 по 40 этого слова, так что знак числа совпадает с



*P* - машинный порядок  
*S* - знак числа  
*M* - мантисса числа

Рис. 1. Формат вещественных и целых чисел в оперативной памяти БЭСМ-6

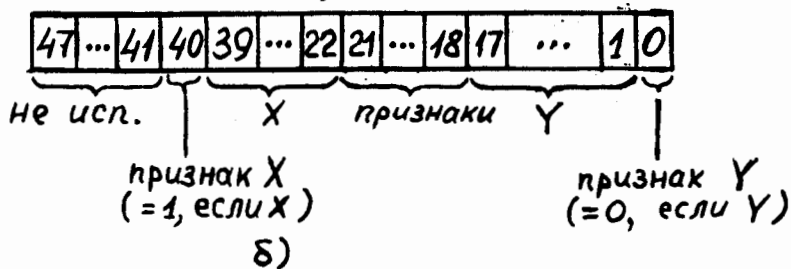
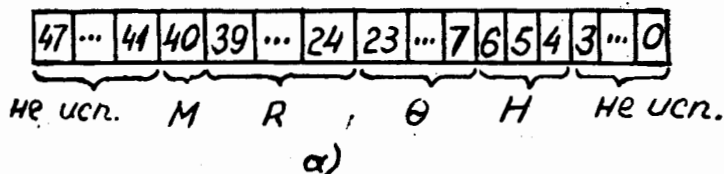


Рис. 2. Формат  $R-\theta$  координат в массиве  $IDIG(\alpha)$  и  $X-Y$  координат в массивах  $CPTXY, VXY, FIDXY(\delta)$

признаком опорной метки  $M$  (см. Формат записи чисел в ЭВМ БЭСМ-6 на рис. 1).

Эти слова заполняют общий помеченный блок  $IDIG$  размерностью 6000 слов. Формат слова  $IDIG$  дан на рис. 2.

Для случаев, когда требуется обработать не всю ленту подряд, а по запросу физиков выбрать только одно или несколько событий по их номерам, предусмотрена программа ввода  $INPUT 2$ .

Массив  $IDIG$  является единственным, где хранятся истинные координаты точек снимка в процессе фильтрации.

На всех промежуточных этапах для получения характеристик какой-либо точки используется адрес  $INDX$  этой точки внутри массива  $IDIG$ . Из числа, хранящегося по этому адресу, выделяется значение  $R$ ,  $\theta$  и  $N$  с помощью специальных подпрограмм-функций:

- $IRDIGF(INDX)$  - выдает целые значения  $R$  в единицах отсчета;
- $RDIGF(INDX)$  - значение  $R$  в мм. в форме с плавающей запятой;
- $ITDIGF(INDX)$  - целое значение  $\theta$  в единицах отсчета;
- $TDIGF(INDX)$  - значение  $\theta$  в радианах в форме с плавающей запятой;
- $INDIGF(INDX)$  - целое значение  $N$  в единицах отсчета;
- $NDIGF(INDX)$  - значение  $N$  в форме с плавающей запятой.

Программой  $UNPKXY$  аналогично преобразуются четверки 12-разрядных слов из последнего блока, содержащие  $X-Y$  координаты вспомогательных точек ( $spt, spt, ept$ ), вершины и центров четырех реперных крестов, в одно 48-разрядное слово БЭСМ-6, заполняя соответственно массивы (общие помеченные блоки)  $CPTXY$  размерности 10,  $VXY$  - размерности 1 и  $FIDXY$  размерности 4. Формат слова БЭСМ-6, содержащего  $X-Y$  координаты, приведен на рис. 2б).

Все промежуточные вычисления в процессе фильтрации производятся над точками в системе координат  $R-\theta$ , и только конечный результат - координаты отфильтрованных точек треков события - переводятся в координаты  $X-Y$ . Поэтому все точки, заданные в  $X-Y$  системе, перед началом фильтрации переводятся в полярную систему координат  $R-\theta$ . Преобразование выполняется с помощью подпрограммы  $GO TO RT(IVERT, IPT, R, TH)$ , где

*IPT* - метка слова, в котором хранятся координаты  $X_{pt}$  и  $Y_{pt}$  точки, подлежащей переводу;

*IVERT* - слово, в котором хранятся координаты вершины  $X_v$  и  $Y_v$ ;

*R, TH* - результаты перевода в полярную систему.

Формулы, по которым осуществляется перевод с одновременной коррекцией на различного рода дисторсии и искажения, приведены в описании калибровочной программы *CALBR* /2/.

В будущем программа *PREFIL*, возможно, будет изменена в связи с необходимостью увеличения разрядности  $N$  и добавления ширины импульса  $\Delta \theta$ , что потребует расширения информации о каждой точке трех слов малой ЭВМ до четырех.

#### Опознавание треков методом гистограммирования

Функциональная схема метода гистограммирования достаточно хорошо известна /3,4/ и сводится к разбиению области, в которой необходимо обнаружить трек, на ряд коридоров, имеющих последовательные наклоны, расположенные веером между двумя экспериментальными значениями ( $\pm SLMAX$ ), соответствующими некоторому минимальному в данном эксперименте радиусу кривизны  $R_0$  треков. Если в какой-то интервал гистограммы попадет число точек, большее половины числа оборотов спирали, покрывающих область гистограммирования, то эти точки рассматриваются как возможно составляющие трек. За счет перекрытия интервалов неизбежно появляются "лишние" треки, которые можно исключить по такому правилу: если два интервала содержат не менее 60% одних и тех же точек, то эти интервалы считаются включающими тот же самый трек, и из двух кандидатов выбирается более гладкий, т.е. тот, у которого меньше среднеквадратичное отклонение  $\sigma$  точек от середины трека.

Метод гистограммирования применяется прежде всего в программе *RECOGN* для опознавания треков в области, прилегающей к вершине события на  $1/5$  части всего спирального скана (примерно первые 15 оборотов спирали, не считая самого первого, на котором измерения не производятся), где фильтрующий эффект щели наиболее значителен. Ширина интервала гистограммирования в существующем варианте програм-

мы принята равной  $\Delta \theta = \pi \cdot 2^{-6}$ , что соответствует /4/  $1,5$  ширины трека на радиусе  $R_{MIN}$ , относящемся к началу области гистограммирования, и 10-кратной ширине трека на радиусе  $R_{MAX}$  в конце этой области.

Для определения максимального наклона  $SLMAX$  можно воспользоваться приближенным уравнением трека в плоскости  $(R, \theta)$  /5/

$$\theta = \theta_0 + \alpha r + \beta/r + \gamma r^3, \quad (1)$$

где  $\theta_0$  - угол между полярной осью и касательной к треку в вершине,

$$\alpha = (2R)^{-1}$$

$R$  - радиус кривизны трека,

$\beta$  - мера близости трека к вершине.

Отсюда получаем, что для минимального радиуса кривизны  $R_0$

$$SLMAX = \alpha = \frac{1}{2R}. \quad (2)$$

Кроме этого, потребность в опознании трека может возникнуть при использовании вспомогательных точек, отмеченных оператором.

Тут возможны два случая:

1) Наличие короткого трека, останавливающегося в области начального гистограммирования и помеченного оператором точкой *Spt* или *ept*, которую мы будем называть "короткой" вспомогательной точкой. Для радиуса такой точки справедливо неравенство:

$$R_p \leq \frac{1}{5} RLIM, \quad (3)$$

где  $RLIM$  - наибольший возможный радиус спирального скана.

2) Наличие "длинной" вспомогательной точки ( $R_p > \frac{1}{5} RLIM$ ), которой не удалось сопоставить ни одного трека из числа треков, обнаруженных программой *RECOGN* и прослеженных программой *FOLLOW*. (Такое сопоставление осуществляется программой *HUNT*, описанной ниже).

В случае 1) вызывается программа *SHORT*, которая выполняет гистограммирование в районе "короткой" вспомогательной точки.

Во втором случае программа *HUNT* вызывает программу *LONG*, осуществляющую поиск трека в области от "длинной" вспомогательной точки назад, в направлении к вершине, на 15 оборотов спирали.

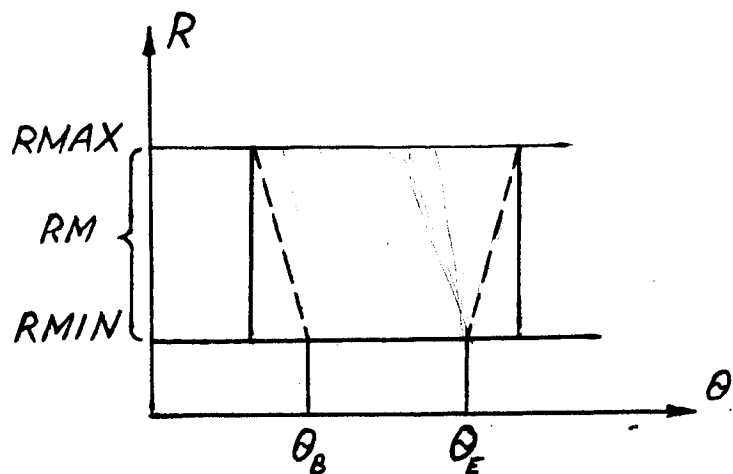


Рис. 3

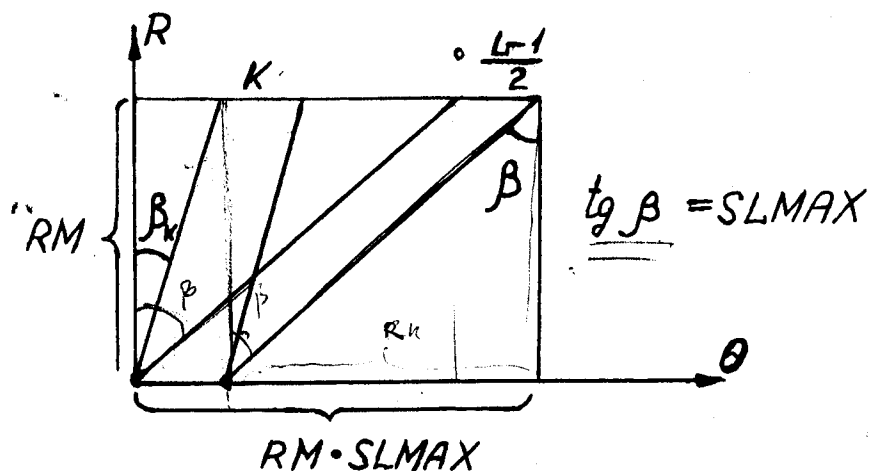


Рис. 4

Каждая из программ *RECOGN*, *SHORT* и *LONG* пользуется одним и тем же комплексом программ, осуществляющих вышеописанный метод гистограммирования:

```

SUB ROUTINE WFIELD (TMN, TMX)
SUB ROUTINE HIST (IZ, IB, IE)
SUB ROUTINE HITEST (IB, IE)
SUB ROUTINE SORT (NN, NCPT)
SUB ROUTINE ELIMTR (NN)
SUB ROUTINE SPIR (NREV)

```

Параметры, необходимые для работы этих подпрограмм, задаются перед их вызовом в программах *RECOGN*, *SHORT* и *LONG* в общих блоках (*COMMON*), описанных в приложении.

Ниже даются описания подпрограмм, осуществляющих реализацию метода гистограммирования.

*SUB ROUTINE WFIELD* осуществляет выбор из массива *IDIG* всех точек  $(R_p, \theta_p)$ , удовлетворяющих условию:

$$R_{MIN} \leq R_p \leq R_{MAX}$$

$$\theta_B - RM \cdot SLMAX \leq \theta_p \leq \theta_E + RM \cdot SLMAX \quad (4)$$

т.е. лежащих в прямоугольнике плоскости  $(R, \theta)$ , имеющем высоту  $RM = R_{MAX} - R_{MIN}$  и в основании отрезок  $(\theta_B - RM \cdot SLMAX, \theta_E + RM \cdot SLMAX)$  (см. рис.3).

Выбранные точки помещаются в рабочий массив *WORK* размером 2000 слов.

*SUB ROUTINE HIST* осуществляет гистограммирование точек из массива *WORK*, проектируя их на отрезок  $(\theta_B, \theta_E)$  оси  $R = R_0$ , разбитый на *NBIN* секторов длины

$$\Delta \theta = \frac{\theta_E - \theta_B}{NBIN}$$

Для каждого *I*-го сектора ( $I = 1, NBIN$ ) проектирование производится по множеству наклонов *L* ( $L = 1, NSL$ , где *NSL* - нечетное число), меняющихся от  $-SLMAX$  до  $+SLMAX$  с шагом  $DSL = \frac{\Delta \theta}{RM}$ , так что любой наклон может быть вычислен по своему номеру *K* ( $K = -\frac{NSL-1}{2}, +\frac{NSL-1}{2}$ ):

$$SL_K = \text{tg } \beta_K = (K \cdot DSL) \quad (\text{см. рис.4}).$$

Программа *HIST* работает по следующему алгоритму. Перебираются все *NSL* наклонов. Для каждого *K*-го наклона все точки *WORK* проектируются с этим наклоном на ось  $R = R_B$ . Координата  $\theta_{np}$  может быть вычислена по формуле:

$$\theta_{np} = TPR = \theta_p - \operatorname{tg} \beta_k \cdot (R_p - R_B). \quad (5)$$

Если результат проектирования удовлетворяет условию

$$\theta_B \leq TPR \leq \theta_E, \quad (6)$$

то вычисляется величина

$$IK = \left[ \frac{TPR}{\Delta \theta} + 1 \right] + 64,$$

( $[ ]$  - символ целой части) и по адресу *IK* в рабочую гистограмму *KS* добавляется единица.

Если условие (6) не выполняется, то в случае, когда *HIST* вызвана из *RECOGN*, к *TPR* добавляется  $2\tau$ , если  $TPR < \theta_B$ , или  $-2\tau$ , если  $TPR > \theta_E$ .

В случаях, когда *HIST* вызвана из *SHORT* или *LONG*, точку из *WORK*, для которой (6) не выполняется, пропускают.

После этого гистограмма *KS* просматривается в пределах номеров ячеек  $IB + IE$  (соответствующих значениям  $\theta_B$  и  $\theta_E$ ), и, если содержимое ячейки с номером *IK* равно или превышает порог *IP*, оно передается в массив гистограмм *IH* по адресу

$$NH = (IK - 1) \cdot NSL + L, \quad \text{где } L = \overline{1, NSL}.$$

После перебора всех *NSL* наклонов получается массив *IH*, состоящий из *NBIN* гистограмм по *NSL* ячеек, уже проверенный по порогу *IP*, а также подсчитывается число первоначально опознанных треков *KT*.

*SUBROUTINE HITEST* - проверяет результаты гистограммирования по следующей системе тестов:

- 1) нахождение возможных направлений треков путем отыскания максимума в каждой из *NBIN* гистограмм;
- 2) устранение "лишних" треков, появляющихся за счет перекрытия коридоров для соседних направлений.

Алгоритм второго теста следующий. Пусть максимум *I*-ой гистограммы находится в ячейке  $IK$  ( $IK = \overline{1, NSL}$ ). В (*I*+1)-ой гистограмме выбирается содержимое (*IK*-1)-ой ячейки, и, если она не пуста, сравнивается с максимумом *I*-ой гистограммы. Обозначим через  $\langle x \rangle$  содержимое ячейки с номером *X*. Тогда при

$$\langle IK \rangle_I > \langle IK - 1 \rangle_{I+1} \quad (8)$$

Трек считается опознанным в *I*-м секторе в направлении *K*, а трек с направлением (*K*-1) сектора (*I*+1) считается "лишним". После этого исключаются из рассмотрения, т.е. "обнуляются" ячейки с номерами (*IK*-1), *IK*, (*IK*+1) в гистограмме *I* и ячейки (*IK*-2), (*IK*-1), *IK* в гистограмме (*I*+1), ибо в них может быть отмечен тот же трек, но с меньшим числом точек.

Пара (*I*, *K*), дающая направление опознанного трека, заносится в массив *NT* и добавляется единица в счетчик отфильтрованных треков *KFT*.

Если же условие (8) не выполнено, то трек опознается в направлении (*I*+1, *K*-1), а трек с направлением (*I*, *K*) считается "лишним" и затирается вместе с боковыми лишними треками.

Затем продолжается обработка гистограммы *I*, т.е. снова ищется максимум из оставшихся значений и сравнивается со значениями в соседних гистограммах до тех пор, пока гистограмма не будет пуста.

После завершения обработки *I*-ой гистограммы обрабатывается (*I*+1)-я и т.д.

Таким образом, после работы *HITEST* появляются данные о числе *KFT* треков, прошедших тесты 1) и 2) и массив возможных направлений треков *NT*. Эта информация выдается на печать.

Для завершения процедуры опознавания треков необходимо выделить точки, принадлежащие отобраным направлениям, и собрать эти точки в отдельный массив. Это делается с помощью подпрограммы *SORT*.

*SUBROUTINE SORT* - создает массив *TM*, состоящий из *KFT* треков. С этой целью для каждой пары (*I*, *K*) из массива *NT* осуществляется проектирование всех точек *WORK* на ось  $R = R_B$  с наклоном  $SL = K \cdot D \cdot SL$ , после чего отбираются только те точки, которые попадают в сектор *I*, т.е. пары  $(R_p, \theta_p)$ , для которых



после проектирования по формуле (5) выполняется соотношение:

$$\Delta \theta (I-1) \leq TPR \leq \Delta \theta \cdot I. \quad (9)$$

Отобранные таким образом точки, принадлежащие к "трекам-кандидатам", заносятся в массив  $TM$ , в котором каждый трек занимает два подмассива:

$M1$  - паспорт трека, состоящий из 10 ячеек,

$M2$  - массив координат точек трека.

Паспорт  $M1$  имеет следующую структуру:

- $M1(1)$  -  $NPT$ - число точек трека.
  - $M1(2)$  -  $\theta_0$
  - $M1(3)$  -  $\alpha$
  - $M1(4)$  -  $\beta$
  - $M1(5)$  -  $\gamma$
  - $M1(6)$  -  $I$
  - $M1(7)$  -  $K$
  - $M1(8)$  -  $H$
  - $M1(9)$  -  $NCP$
  - $M1(10)$  -  $\delta$
- параметры уравнения трека вида (I), которые вычисляются в программе FOLLOW
- номер сектора, в котором лежит трек.
- номер направления.
- средняя амплитуда импульса.
- номер вспомогательной точки, соответствующей треку.
- среднеквадратичное отклонение точек трека от кривой вида (I).

Знак числа, находящегося в ячейке  $M1(I)$ , используется для "вычеркивания" трека; если трек будет забракован по тем или иным критериям (слишком мало точек на треке, велик параметр  $\beta$  и т.п.), полагается  $M1(I) < 0$ .

Массив  $M2$  имеет размерность  $2 NPT$  и следует непосредственно за  $M1$ .

Предусматривается вариант программы SORT, который в случае слишком большого числа треков  $KFT$  хранит в массиве  $M2$  точки в закованном виде. В этом варианте в  $M2$  хранятся не сами координаты, а номера соответствующих слов массива  $IDIG$ , причем, поскольку для записи такого номера достаточно 13 двоичных разрядов, то в соответствии с форматом слова БЭСМ-6, приведенном на рис. I, в каждом слове массива  $M2$  можно уместить по 3 номера. Это приведет к сокращению размерности  $TM$  почти в 6 раз.

**SUBROUTINE ELIMTR** выполняет отсев треков из  $TM$ , на 60% состоящих из одних и тех же точек. Для этого просматриваются пары  $(I, K)$  из массива  $NT$ , и, если значения  $I$  у соседних пар отличаются на величину, не превосходящую 2, сравниваются точки соответствующих треков (по их номерам в  $IDIG$ ).

При совпадении 60% и более точек у этих треков вызывается программа **FOL 1** (описана ниже) для вычисления среднеквадратичного отклонения  $SIG$  точек от трека, подгоняемого по кривой вида (IO). Трек, для которого величина  $SIG$  окажется больше, вычеркивается (приписывается минус числу  $M1(I)$ ). При этом программа **ELIMTR** заносит значения  $SIG, \theta, \alpha, \beta$  для оставшегося трека в массив  $M1$  этого трека.

Число треков, оставшихся после работы программы **ELIMTR**, фиксируется как  $NF$ . Собираются все оставшиеся  $NF$  треков в массив  $TMF$ , из которого они переносятся на поле  $TM$ , начиная с его первой свободной ячейки ( $NE + 1$ ).

**SUBROUTINE SPIR(NREV)** вычисляет число витков спирали между  $RMIN$  и  $RMAX$ .

Для этого массив **WORK** просматривается с целью подсчета числа опорных меток между значениями радиуса от  $RMIN$  до  $RMAX$ . (Опорная метка задается  $I$ -й сорокового разряда в записи числа).

Число витков спирали фиксируется как  $NREV$ .

После введения вышеперечисленных программ, осуществляющих процедуру гистограммирования, функции программ **RECOGN**, **SHORT** и **LONG** сводятся практически только к вычислению параметров, используемых в вызываемых подпрограммах, и вызову самих подпрограмм.

**SUBROUTINE RECOGN** - готовит параметры для общей процедуры гистограммирования.

Проектирование точек массива **WORK** осуществляется на ось  $R_\theta = 0.4$  на отрезок оси  $\theta (-T, T)$  в пределах первых 15 витков спирали, т.е.  $RMIN = 0.4$   
 $RMAX = 6.4$

Задаются такие величины, как число секторов  $NBIN$ , число наклонов  $L$ , порог  $IP$  для числа попаданий точек в коридор; вычисля-

ются *DSL* и *SLMAX*. Затем последовательно вызываются все подпрограммы комплекса гистограммирования.

*SUBROUTINE SHORT* - работает только в том случае, если есть "короткие" вспомогательные точки (*spc* и *epc*).

Программа задает набор параметров, необходимых для работы комплекса гистограммирующих программ.

Проектирование массива *WORK* производится на ось  $R_b = RCP$  - радиусу вспомогательной точки на отрезок оси  $\theta (\theta_b, \theta_e)$ .

В первом варианте программы было принято  $NBIN = 3$ . Отсюда

$$\begin{cases} \theta_b = \theta_{cpt} - \frac{3r}{128} \\ \theta_e = \theta_{cpt} + \frac{3r}{128} \end{cases}$$

Границы по радиусу для массива *WORK* были  $RMIN = 0.4$   
 $RMAX = RCP$ .

По программе *SPIR* вычисляется число оборотов спирали *NREV* между *RMIN* и *RMAX*; выбирается порог  $IP = [\frac{1}{2}(NREV-1)+1]$ . Задаются такие параметры, как *DSL* и *SLMAX*. При накоплении точек трека программой *SHORT*, если она вызвана из *SHORT*, в *M1(9)* сразу заносится номер вспомогательной точки, соответствующей данному треку.

*SUBROUTINE LONG* - программа предназначена для поиска треков вблизи "длинной" вспомогательной точки, если программа *HUNT* не смогла поставить в соответствие данной вспомогательной точке ни одного трека из массива *TM*, где хранятся треки, опознанные *RECOGN* и продолженные *FOLLOW*.

Гистограммирование по программе *LONG* происходит путем проектирования массива *WORK* на ось  $R_b = RCP$  - радиусу вспомогательной точки на отрезок оси  $\theta (\theta_b, \theta_e)$ , где

$$\theta_b = \theta_{cpt} - \frac{5r}{128}$$

$$\theta_e = \theta_{cpt} + \frac{5r}{128}$$

при  $NBIN = 5$ , как было принято в первоначальном варианте программы. Для массива *WORK* границы по радиусу выбирались равными

$$RMIN = R_{cpt} - 6.0$$

$$RMAX = R_{cpt}$$

Вычислялось значение порога  $IP = [\frac{1}{2}(NREV-1)+1]$ , где *NREV* - число оборотов спирали между *RMIN* и *RMAX*, выдаваемое программой *SPIR*; задаются *DSL* и *SLMAX*.

При создании трека программой *SHORT*, если она вызывается из *LONG*, в *M1(9)* сразу заносится номер вспомогательной точки, в окрестности которой нашелся данный трек.

### Прослеживание опознанных треков

Следующий этап фильтрации - прослеживание опознанных треков - осуществляется подгонкой по найденным точкам трека кривой вида (I) методом наименьших квадратов (М.Н.К.) с последующим прогнозированием положения следующей точки трека и, после нахождения её в прогнозируемом районе, - повторной подгонкой, новым прогнозированием и т.д. На каждом этапе осуществляется выброс точек трека, отстоящих более, чем на утроенную величину среднеквадратичного отклонения. Прослеживание продолжается до границ области сканирования или до появления нескольких пропусков точек в предсказанных местах, когда специальная "функция пропуска" превышает критический уровень.

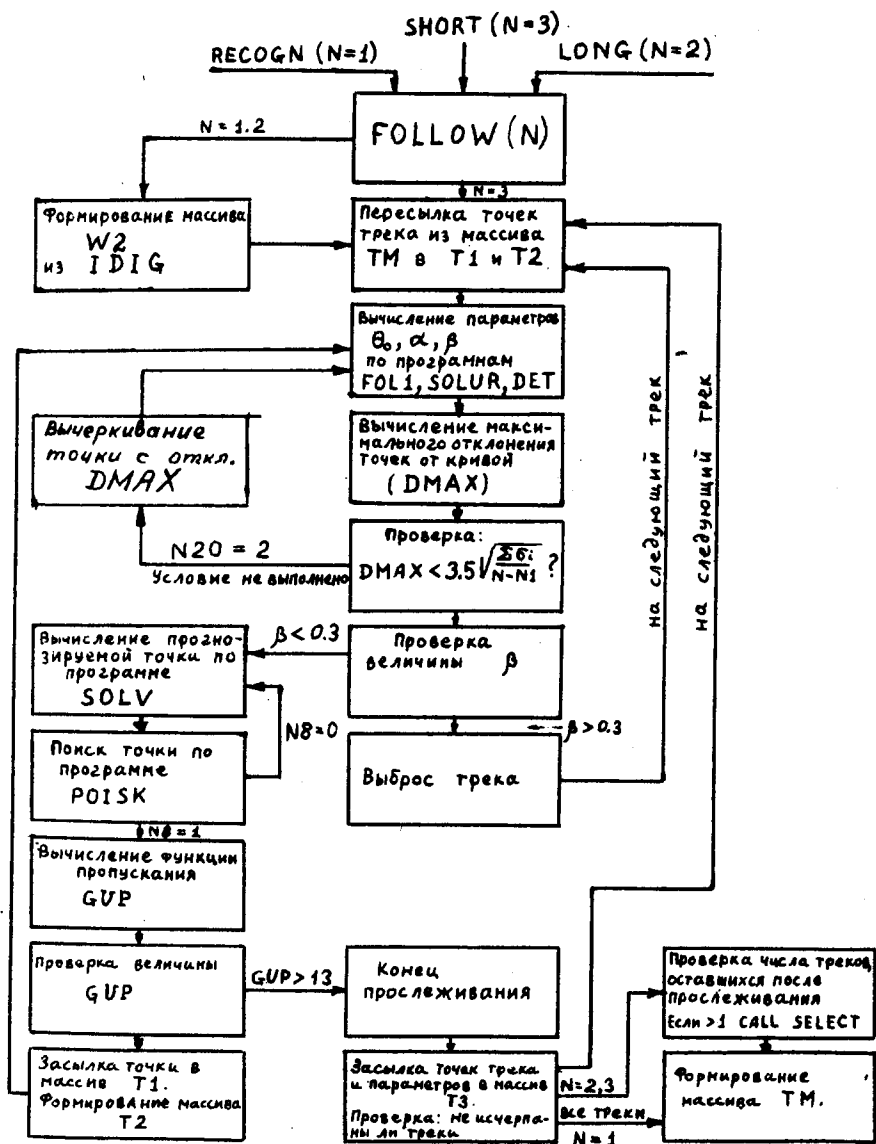
Как было предложено в <sup>13/</sup>, для ускорения процесса прослеживания, в уравнении (I) для подгонки трека число параметров было сведено к трем, за счет исключения параметра  $\gamma$ , а число точек трека в массиве, по которому осуществляется подгонка, сохраняется постоянным (при включении в этот массив новой точки исключается точка с противоположного конца трека). Оба эти предложения позволяют существенно сократить время на многократно повторяемые

в М.Н.К. процедуры вычисления детерминантов для определения параметров и функционала для определения среднеквадратичной погрешности.

Окончательная подгонка и выброс с учетом всех четырех параметров делается по общему массиву точек, отнесенных к треку при прослеживании, после выхода на конец трека.

Управляющей программой этапа прослеживания треков является программа *FOLLOW(N)*, параметр *N* которой принимает соответственно значения 1,2,3, в зависимости от того, после какой программы

# БЛОК-СХЕМА ПРОГРАММЫ FOLLOW(N)



RECOGN, LONG или SHORT вызывается FOLLOW. В соответствии со значением  $N$  программа FOLLOW вычисляет входные параметры для поочередного вызова подпрограмм, реализующих отдельные части алгоритма прослеживания в соответствии с блок-схемой, приведенной на стр. 18.

Подпрограмма FOLLOW ( $N$ ) - управляющая программа этапа прослеживания треков на основе материала, полученного после первого этапа фильтрации.

Предыдущий этап опознавания треков дает данные о полосах, идущих от  $z$  нач. до  $z_{max}$  и содержащих кандидаты в треки.

По точкам, лежащим в этих коридорах, с помощью М.Н.К. (по подпрограмме FOL 1) проводится начальная кривая

$$\theta = \theta_0 + \alpha z + \beta / z \quad (10)$$

и определяются параметры  $\theta_0, \alpha, \beta$ .

Дальше вычисляется величина  $G_i$  - отклонение от кривой в каждой точке, максимальное отклонение  $DMAX = \max G_i$  и проверяются условия

$$(a) \quad DMAX \leq 3,5 \sqrt{\frac{\sum G_i}{N - N_1}},$$

где  $N$  - число точек трека,  
 $N_1$  - число параметров

$$(b) \quad DMAX \leq Const.$$

Если DMAX не удовлетворяет этим условиям, то соответствующая точка выбрасывается из массива точек трека и проводится новая кривая с последующей проверкой DMAX.

Далее идет проверка на число оставшихся точек. Если  $N < 8$ , то трек выбрасывается с соответствующим выводом информации о причине выброса.

При соблюдении вышенаписанных условий проверяется величина параметра  $\beta / \beta_1$ . Если  $\beta > 0,3$ , то трек выбрасывается с выдачей диагностики на печать.

При  $\beta \leq 0,3$  начинается дальнейший процесс прослеживания трека.

С помощью подпрограммы *SOLV* находится прогнозируемая точка  $(R^*, \varphi^*)$ . Подпрограмма *POISK* находит среди массива точек *IDIG* на данном витке спирали ближайшую точку к прогнозируемой.

Найденная точка подпрограммой *ZASILK* включается в массив точек трека, при этом точка с противоположного конца трека отбрасывается, т.е. при подгонке по М.Н.К сохраняется все время постоянное число точек.

По новому массиву точек вновь осуществляется вычисление параметров  $\theta_0, \alpha, \beta$  по программе *FOL1* с последующим выбросом плохих точек.

Прослеживание заканчивается в двух случаях:

- 1) величина радиуса выше найденной точки будет превосходить предельный радиус *RLIM* ;
- 2) для того, чтобы найденная выше точка принадлежала треку, необходимо, чтобы величина функции пропуска *GAP* не превышала 13. Функция пропуска *GAP* определяется так:

$$GAP = 4G_{l-2} + 2G_{l-1} + G_l,$$

где  $G_l$ , названная функцией пропуска, есть число оборотов спирали между  $l$  и  $(l-1)$  точками.

Следующий этап - проведение кривой (I) методом М.Н.К по всем найденным точкам трека и вычисление параметров  $\theta, \alpha, \beta, j$  с соответствующим уточнением и проверкой параметра  $\beta$  и окончательная засылка параметров  $\theta_0, \alpha, \beta, j$  и точек трека  $(R, \theta)$  в массив *TM*.

С помощью подпрограммы *OUTFIL* полученный массив *TM* преобразуется в формат, нужный для ввода в геометрическую программу *THRESH*.

Подпрограмма *FOLLOW(N)* включает в себя следующий комплекс подпрограмм:

### I. SUBROUTINE FOL1 (T1, L1, SIG)

Подпрограмма вычисляет коэффициенты для системы нормальных уравнений (6), из которой определяются параметры

$$\theta_0, \alpha, \beta$$

$$\theta_0 N + \alpha \sum z_i + \beta \sum 1/z_i = \sum \theta_i$$

$$\theta_0 \sum z_i + \alpha \sum z_i^2 + \beta N = \sum \theta_i z_i$$

$$\theta_0 \sum 1/z_i + \alpha \sum 1/z_i^2 + \beta \sum 1/z_i^3 = \sum \theta_i / z_i^2$$

Суммирование ведется по числу точек трека. В этой же подпрограмме вычисляется величина среднеквадратичного отклонения

$$SIG = 3.5 \sqrt{\frac{\sum G_i}{N - N1}},$$

где  $G_i = |\theta_i - \theta_0 - \alpha R_i - \beta / R_i|$ ,

$N$  - число точек трека,

$N1$  - число исконых параметров в уравнении (I).

Значения формальных параметров:

$T1$  - массив координат трека размерностью 90,

$L1$  - число точек трека,

$SIG$  - вычисляемое среднеквадратичное отклонение.

Определение параметров  $\theta_0, \alpha, \beta$  осуществляется обычным путем вычисления отношения определителей системы уравнений с помощью подпрограмм *SOLUR* и *DET*.

II. SUBROUTINE SOLUR (A, TH, ALFA, BETA) - подпрограмма решения системы 3-х линейных уравнений методом Крамера.

Значения формальных параметров:

$A$  - матрица коэффициентов систем размерностью I2,

$\theta_0$   
ALFA } - исконое решение системы  
BETA }

III. SUBROUTINE DET (A, D) - подпрограмма вычисления определителя 3-го порядка.

Значения формальных параметров:

$A$  - массив коэффициентов определителя размерностью 3,

$D$  - значение вычисляемого определителя.



IV. SUBROUTINE SOLV(T2, L1, TPN, RPN, NSOL) -

подпрограмма предсказания следующей точки трека по следующему алгоритму.

Предположим, что предсказываемое значение радиуса

$$RPN = R_n + \Delta R \times NSOL, \text{ где } \Delta R = R_n - R_{n-1},$$

NSOL - число оборотов спирали между предсказываемой точкой и последней точкой трека.

Здесь  $n$  - число точек трека.

Тогда значение угла TPN предсказываемой точки вычисляется по уравнению (I)

$$TPN = \theta_0 + \alpha \cdot RPN + \beta / RPN.$$

Значения формальных параметров:

T2 - массив координат трека размерностью 90,

L1 - число точек трека,

$\left. \begin{matrix} TPN \\ RPN \end{matrix} \right\}$  координаты предсказываемой точки,

NSOL - определено выше.

V. SUBROUTINE POISK(SIG, RPN, TPN, \theta\theta, RR, N1, N12, N7) -

подпрограмма поиска вероятных точек, ближайших к прогнозируемой в коридоре  $\pm SIG$ . Из всех полученных вероятных точек выбирается та, у которой отклонение SIG наименьшее.

Значения формальных параметров:

SIG - среднеквадратичная ошибка отклонения,

$\left. \begin{matrix} RPN \\ TPN \end{matrix} \right\}$  - координата прогнозируемой точки,

N1 - число найденных точек,

N12 - переключатель режима работы, равный 1 при первом входе в подпрограмму POISK и равен 2 при последующих входах,

N7 - номер ячейки массива W2, с которой нужно начинать поиск вероятных точек,

$\left. \begin{matrix} \theta\theta \\ RR \end{matrix} \right\}$  - массивы размерностью 10 для найденных вероятных точек.

VI. SUBROUTINE ZASILK(RR, \theta\theta, N1, NS) -

подпрограмма включения в массив точек трека найденной вероятной точки.

Значение формальных параметров:

$\left. \begin{matrix} RR \\ \theta\theta \end{matrix} \right\}$  - массивы с координатами найденных вероятных точек,

N1 - число найденных вероятных точек,

NS - общее число точек трека после работы подпрограммы.

VII. SUBROUTINE GAVEL(ANUM, NUMBER, KSIGN) -

подпрограмма, располагающая точки трека либо в возрастающем по радиусу порядке (при KSIGN = +1), либо в убывающем порядке (при KSIGN = -1).

Значения формальных параметров:

ANUM - массив точек трека размерностью 90,

NUMBER - число точек трека,

KSIGN =  $\pm 1$ ;

VIII. SUBROUTINE WORKRT(N, NWI) подпрограмма выбора точек из массива IDIG по заданным RMAX и RMIN для дальнейшего прослеживания точек.

Значения формальных параметров:

N - признак, равный 1, 2, 3, указывающий, из какой подпрограммы было обращение в FOLLOW(N),

NWI - номер ячейки массива W2, с которой нужно начинать выбор точек в подпрограмме POISK, в случае прослеживания к вершине.

IX. SUBROUTINE SELECT(NUMBER, JTM) -

подпрограмма выбора ближайшего к данной опорной точке трека.

Значения формальных параметров:

NUMBER - число треков,

JTM - номер опорной точки.

X. FUNCTION DEVIAT(R, T) - подпрограмма вычисления отклонения  $\delta(r, t)$  от кривой в данной точке (R, T) по формуле

$$DEVIAT = ABS(\theta_0 - \alpha R - \beta/R - T)$$

XI. FUNCTION GUP(G) - подпрограмма вычисления функции пропуска

$$GUP = 4G(I) - 2G(I-1) + G(I-2)$$

Значение формального параметра:

G - массив размерности 3, где хранятся значения функций пропуска в трех подряд идущих точках трека.

XII. SURROUTINE HUNT - подпрограмма сопоставления "длинной" опорной точке трека по следующему алгоритму.

Первоначально программа проверяет, есть ли вообще "длинные" *cpt*, т.е. из массива опорных точек выбираются все те, для которых

$$R_p > \frac{1}{5} RLIM$$

Далее строится матрица расстояний от каждой *cpt* до всех треков. Если обозначить через *i* - число выбранных *cpt*, через *j* - число треков, через  $(R_p, \theta_p)$  - координаты данной *cpt*, то элемент матрицы  $\delta_{ij}$  будет равен

$$\delta_{ij} = |\theta_p - \theta_n|$$

$\theta_n = \theta_0 + \alpha R_p + \beta/R_p$  - вычисленное значение угла  $\theta_n$  при данном радиусе  $R_p$ .

В построенной таким образом матрице находится минимальное значение  $\delta_i = \min |\delta_{ij}|$ . Если найденное минимальное значение будет меньше некоторой *const*, равной приблизительно трем ширинам трека, то данная *cpt* считается принадлежащей *j*-ому треку. Если же *cpt* не приписывается ни к одному треку, то происходит вызов подпрограммы *LONG*, осуществляющей вновь гистограммирование вокруг опорной точки.

## ЛИТЕРАТУРА

1. Астахов и др. Препринт ОИЯИ РЮ-4943, Дубна, 1970.
2. E.H. Eichmann. The Spiral Reader Calibration Program. CERN/D Ph II/PROG 69-2 25.4. 1969 г.
3. Материалы совещания по программированию и вычислительным методам решения физических задач. II-4655, ОИЯИ, ДВТА, май 1969 г.
4. J. Burkhard, L. Lloyd, G. Lynch, Programms for the SR, 1965 г.
5. Г.А. Ососков. Сообщение ОИЯИ РИИ-4463; Дубна, 1969.
6. А.Н. Крылов. Лекции о приближенных вычислениях. ГОСТЕХИЗДАТ. М-Л-1950 г.

1457  $\frac{117}{K-35}$

## ПРИЛОЖЕНИЕ

### I. Общие блоки (COMMON)

```
COMMON/BLOK/B
COMMON/B1/IDIG
COMMON/B2/NT, TM
COMMON/B3/RP, TP
COMMON/H1/RB, RM, TB, TE, RMIN, RMAX
COMMON/H2/NBIN, NSL, SLMAX, DSL, DQ, IP
COMMON/L1/LW, LTM, NE, KT, KFT, NUMCPT, RLIM
COMMON/A/A
```

IDIG (6000) - массив целых чисел, содержащих в разрядах с 4 по 40 результаты спирального сканирования.

B (4000) - рабочий массив, используемый в ряде подпрограмм под массивы WORK (2000), IH (2000), T3 (1000), W2 (3000).

$NT(100)$  - массив целых чисел, содержащий пары ( $I$ ,  $K$ ) направленных опознанных треков.  
 $TM(2000)$  - массив, содержащий опознанные треки в виде  $M1$  (паспорт трека), плюс  $M2$  - массив точек трека.

$COMMON /B3/ RP, TP$

$RP(10)$  - массив радиусов вспомогательных точек.

$TP(10)$  - массив углов  $\theta$  вспомогательных точек.

$COMMON /H1/$  содержит следующие параметры, необходимые для гистограммирования:

$R_B$  - расстояние от вершины события до прямой, на которую проектируется массив  $WORK$  при гистограммировании.  
 $RM = R_{MAX} - R_{MIN}$ .

$TB$  } начало и конец отрезка ( $\theta_B, \theta_E$ ) на оси  $R_B$ , на кото-  
 $TE$  } рый проектируется массив  $WORK$  при гистограммировании.

$R_{MIN}$  } минимальный и максимальный радиусы точек массива  $WORK$ .  
 $R_{MAX}$  }

$COMMON /H2/$  - также задает параметры для гистограммирования:

$NBIN$  - число секторов, на которое разбивается отрезок ( $\theta_B, \theta_E$ ).

$NSL$  - число наклонов при проектировании массива  $WORK$ .

$SLMAX$  - максимальный наклон при проектировании.

$DSL$  - шаг при переходе от одного наклона к другому.

$DQ$  - длина каждого из  $NBIN$  секторов.

$IP$  - порог для опознания треков по числу точек, попавших в коридор при проектировании.

$COMMON /L1/$

$LW$  - длина массива  $WORK$ .

$LTM$  - длина массива  $TM$ .

$NE$  - последняя занятая ячейка массива  $TM$ .

$KT$  - количество опознанных треков.

$KFT$  - количество треков (из числа  $KT$ ), оставшихся после тестов 1) и 2).

$NF$  - число треков, оставшихся после работы  $ELIMTR$ .

$NUMCPT$  - число вспомогательных точек.

$RLIM$  - максимальный радиус при спиральном сканировании.

$COMMON /A/$

$A(4)$  - массив параметров  $\theta_0, \alpha, \beta, \gamma$  в уравнении (1).

Рукопись поступила в издательский отдел

19 января 1971 года.