Fermilab Library
0 1160 0009500 4

SCIPP-92-15

JUL 1992

SCIPP 92/15
SDC Note 92-209
June 11, 1992

Description of the TS Track Reconstruction Package

BRADLEY HUBBARD, WILLIAM LOCKMAN, AND KATHY O'SHAUGHNESSY

Santa Cruz Institute for Particle Physics
University of California, Santa Cruz, CA 95064

ALAN J. WEINSTEIN

California Institute of Technology

ROUTE TO
NAME
LOCATION

⋆ Work supported in part by the U.S. Department of Energy.

1. INTRODUCTION

In order to evaluate designs for the SDC tracking system, substantial effort has been devoted to detector simulation and track reconstruction. In the past two years, a number of reconstruction packages have been developed for use within the SDC shell program. One such package is TS (for Tracking with Silicon). This note describes the TS package version (v04) used for studies for the Technical Design Report.

The TS package includes routines for finding space coordinates and segments in the silicon strip subsystem and combining them to make tracks. The original track reconstruction code was written by Alan Weinstein, and uses a conceptually simple and fast algorithm for clustering hits in pairs of adjacent layers. The package also includes a track fitting program written originally by Jim Hylen, which fits tracks to a helix including the effects of multiple scattering, and includes data from the various tracking subsystems in the fit. See Reference 1 for parameter definitions† and for more formalized mathematical descriptions of the track finding and fitting.

In this note, we adopt the standard SDC coordinate system, in which the x axis points toward the center of the SSC ring, y points upward, and z points in one beam direction such that the system is right handed.

2. DESCRIPTION

The TS reconstruction is divided into four parts, which will be discussed in detail in the following sections: coordinate finding, segment finding, track finding, and track fitting. The first two are for reconstructing data within the silicon system, and as such are used also by other track reconstruction algorithms. Each part can be independently used from within the TF combined track reconstruction package, which allows one to easily choose between different algorithms. Parts of the TS package can be used without TF, but this is not recommended in general. Many of the reconstruction parameters and cuts used in the TS package can be adjusted using KUIP commands, which are also listed in Appendix A. The quantities calculated in each part are stored in ZEBRA track banks in the SDC framework. Details are given in Appendix B.

In order to conserve memory and keep the program size manageable, instead of using fixed arrays to hold temporary information, the TS package uses dynamically allocated 'temporary arrays'. These are allocated and deallocated with the shell utility routines SHTSCR and SHTSDL, described in Reference [2]. In most cases,

† Note that the SDC convention for curvature is different by a factor of 2 and the sign of the impact parameter is multiplied by $-q$.

when raw or reconstructed quantities are accessed frequently, they are unpacked from the ZEBRA data structures into temporary arrays and accessed via statement functions, saving a large amount of overhead compared with frequent ZEBRA access.

The detailed silicon system geometry are stored in an internal data structure which is accessed via statement functions. This geometry is set up by the ST package, described in Reference 3, and is either initialized at run time (if simulating and reconstructing in the same job) or copied from the begin run record. The ST package contains utility routines STCOOR, STBPOS and STFPOS for translation from strip numbers to coordinates r, ϕ, and z, and vice versa, which are used in the TS reconstruction.

2.1 Coordinates

The reconstruction of coordinates from the raw data is handled by the routine TSFLCO. The raw digitization data bank /RAWD/HTRK/STRA/STSS/STDI, described in Reference 3, consists of clusters of one or more hit strips on the axial and stereo sides of the silicon wafers. The centroid of a hit cluster is reconstructed as the center of the middle strip (for odd numbers of strips in a cluster) or between two strips. Due to the small stereo angle (10 mrad in the default geometry), a stereo strip only crosses 24 axial strips along the length of a readout unit; combined with the low occupancy of the system, this means the association of stereo and axial hits is usually straightforward. There are times, however, where either there is no stereo digitization to match with the axial (*e.g.*, if the particle went through the wedge of dead stereo area near the edge of the wafer) or more than one stereo strip matches the axial digitization. Both cases are flagged in word 7 of the coordinate information bank /RECO/HTRK/STRE/STRS/STCO (see Appendix B).

The errors associated with the coordinate are also stored in the coordinate bank. In the usual case, the errors are taken to be

$$d\phi = \frac{\text{pitch}}{R\sqrt{12}}$$

$$d(long) = \frac{\text{pitch}}{\tan\mu\sqrt{6}}$$

where R is the radius at which the pitch is defined (*i.e.*, the location of the readout chips for the forward) and *long* is the longitudinal coordinate: z for the barrel and r for the forward modules. The stereo angle is μ. These errors are conservative, assuming only single strips are hit. The Monte-Carlo resolution and the predicted system resolution is somewhat better [3].

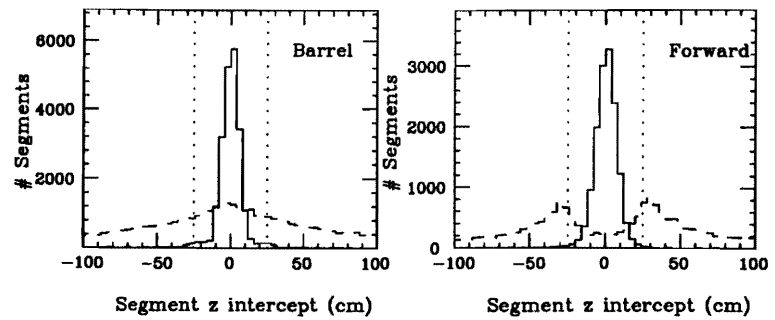

Figure 1 a) z intercept calculated for correct segments (solid), where all hits come from the same MC track, compared to incorrect segments (dashes) in the barrel silicon. The cut at ± 25 cm is indicated with dotted lines. b) z intercept calculated for correct segments (solid), where all hits come from the same MC track, compared to incorrect segments (dashes) in the forward silicon.

2.2 Segments

Local track vectors, or segments, are constructed from coordinates in two adjacent silicon layers. A simple algorithm, driven by the routine TSSEGF, matches a coordinate in an inner layer with the coordinates nearest in ϕ in the next layer, using a minimum p_t cut (default 0.8 GeV). The procedure favors high momentum tracks, but if there is more than one possible match within the ϕ road, then up to 3 choices are kept.

In general, the two coordinates have both axial and stereo information. Some segments will have missing stereo information in one or both layers, in which case the z or r value for the coordinate assumes the middle of the readout unit, with correspondingly large errors. If both coordinates have stereo information, the reconstructed z intercept $z_0 = (r_2 z_1 - r_1 z_2)/(r_2 - r_1)$ is required to be within 25 cm of $z = 0$. This distribution is shown in Figure 1a and b, for the barrel and forward detectors, respectively.

The segments are characterized by parameters measured with respect to the distance of closest approach to the origin (DCAO). The parameters are calculated in TSVPAR. As the barrel and forward systems measure different quantities, the parameter set is different for segments in the two systems. In the barrel, one measures the ϕ coordinate using the axial strips at a given r_i. If we assume an impact parameter of zero, we have the linear relationship

$$\phi_i = \phi_0 + \frac{1}{2}\rho r_i$$

where $\rho = -0.03Bq/p_t$, B in Tesla and p_t in GeV/c.

Therefore two axial measurements in the segment allow a determination of (ϕ_0, ρ) and an associated error matrix (the assumed error on ϕ_i is just the strip resolution divided by r_i). A correction is applied to improve the value of the curvature:

$$\rho \rightarrow \frac{\rho}{\sqrt{1 + \frac{1}{4}\rho^2 r_{avg}^2}} .$$

The association of the small angle stereo strips with the axial strips to measure z_i in the barrel was described in the previous section. The linear relationship

$$z_i = z_0 + s r_i ,$$

where $s = \tan\lambda$ and λ is the dip angle, can be solved for the variables z_0 and s using the information from the two layers. Again, there is an associated error matrix. If any of the coordinates used didn't have a stereo strip match, then the z_i used for the calculation of z_0 and s will be an approximation with a correspondingly large error. This is reflected in the error matrix.

In the forward region, the measured quantities are ϕ_i and r_i at a fixed z_i. The equivalent linear relationships are, in this case,

$$\phi_i = \left(\phi_0 - \frac{\rho z_0}{2s}\right) + \frac{\rho}{2s} z_i \equiv \phi_f + \frac{1}{2}\rho_f z_i$$

and

$$r_i = \left(\frac{-z_0}{s}\right) + \frac{1}{s} z_i \equiv z_{0f} + s_f z_i .$$

The values of (ϕ_f, ρ_f),(z_{0f}, s_f) and their errors can be determined in the same way as for the barrel, using the information from the 2 matched coordinates.

This segment finding procedure can be used on any two adjacent layers. In the current implementation of the algorithm, middle layers are used to match hits with layers on either side (*e.g.*, segments are formed using layers 1 and 2 as well as layers 2 and 3). Double use of hits is resolved at the track finding stage. The segments are stored in the banks /RECO/HTRK/STRE/STRS/STSG (barrel) and /STIE/STRS/STSG (forward). The daughter banks /STSP contain a list of the hits used to determine the segment.

2.3 Track finding

The track finding scheme in TSFIND uses the segments determined above and (potentially) those from other systems to globally associate hits to a track. For speed of access, the segment banks are unpacked into a temporary array by the routine TSUPSG. Figure 2 shows a plot of all the barrel silicon segments from a Higgs event in the (ϕ_0, ρ) plane. Figure 3 shows a small region of (ϕ_0, ρ) space with the segments from a track and the errors shown. Segments from the same track overlap each other, and a χ^2 association can be used to make clusters from the segments.

The routine TSLCLU clusters segments together to make track candidates. Barrel and forward segments are clustered separately as they are described by different parameters. In an iterative scheme, each available segment is tested against all of the previously clustered segments. If there is a match, the segment is added to the cluster, and if not, a new cluster is formed. Two independent χ^2 contributions, χ_r^2 and χ_z^2, are calculated for each (ϕ_0, ρ) and (z_0, s) association, respectively, using the routine TSVCH2, as described in Reference 1. The calculation includes the multiple scattering errors for silicon layers, though it does not fully take into account correlations between layers. For segments from the same Monte Carlo track (which should be associated) the distributions of χ_r and χ_z are shown in Figures 4a and b, respectively. Two segments are associated if both $\chi_r < 3$ and $\chi_z < 3$.

The clustering results in a set of points that make a candidate track as well as an initial guess at the track parameters (assuming zero impact parameter). By default, candidate tracks are required to have 2 or more segments. These tracks are stored in the /RECO/HTRK/TRCT banks. If a track goes through both the barrel and forward modules, the two candidate tracks are passed to the track fitting stage (described in the next section), at which point each is extended to the other region using a road technique.

To be able to cluster together segments from different detector components, the parameters and errors must be convertible to the same format. For the clustering to work most effectively, the errors on the parameters should be comparable. For example, if the parameters of the segments from the straw system are converted to the same format as the silicon barrel, due to the angular resolution in a straw superlayer, the errors are much larger than the silicon equivalent. The result is that the silicon segments from many tracks would associate with one straw segment and the power of the silicon system to resolve tracks in dense environments is wasted. Therefore it was decided that this track finding algorithm is better suited to finding tracks in just the silicon system. A road technique is used in the fitting stage to add information from the outer and intermediate tracker.

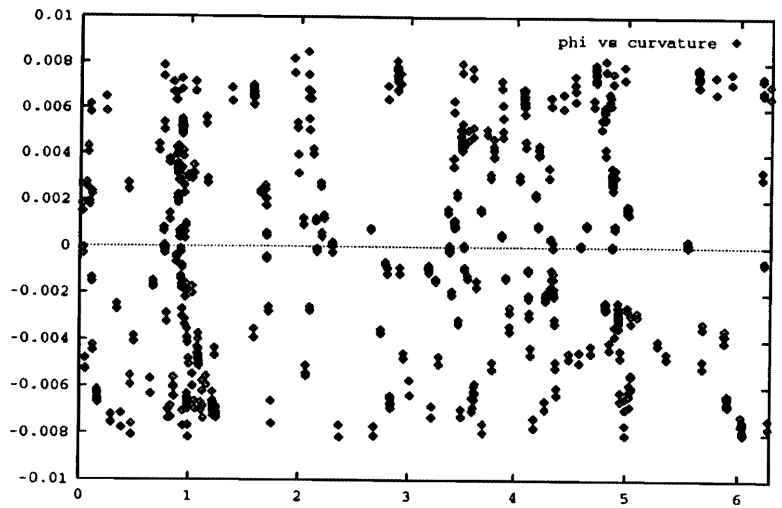

Figure 2. Plot of barrel segments in ϕ_0, ρ space. Tracks appear as dense clusters of segments on this plot. The four tracks very close to the dotted line at $\rho = 0$ are probably the leptons from the Higgs decay.

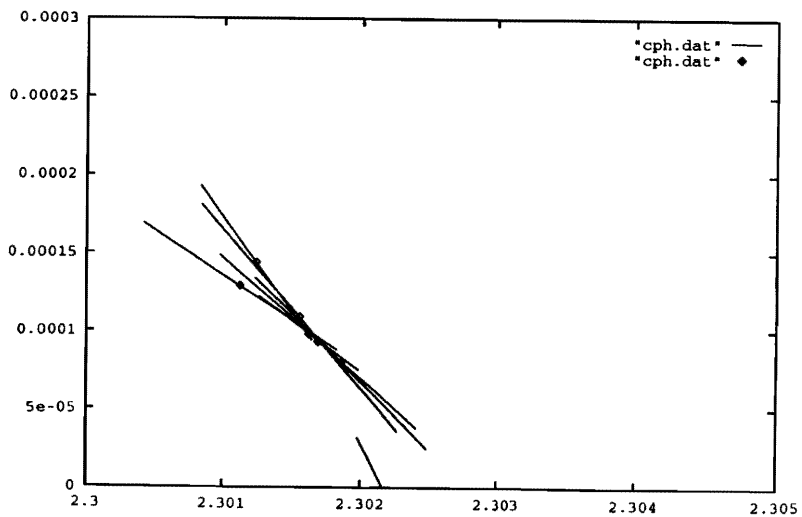

Figure 3. Closeup of barrel segments in ϕ_0, ρ space.

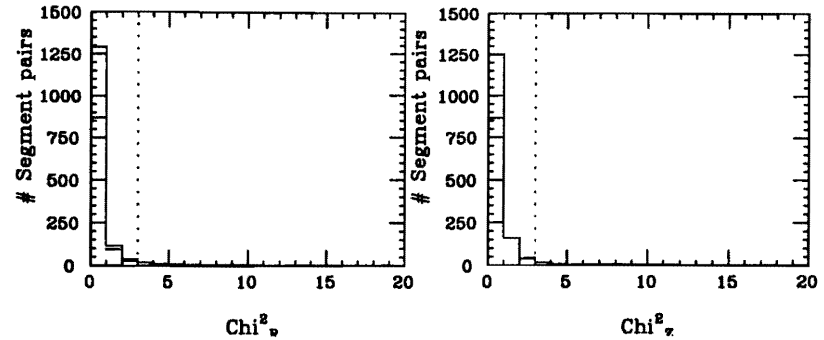

Figure 4 a) Distribution of χ^2_r for segment pairs from the same Monte-Carlo track, for barrel (solid) and forward (dashed) modules. The cut at 3 is indicated with dotted lines. b) Distribution of χ^2_z for segment pairs from the same Monte-Carlo track, for barrel (solid) and forward (dashed) modules.

The simple clustering approach finds most tracks quickly without confusion from other tracks. There are cases where tracks are very close together (*e.g.*, in jets) and the clustering routine associates them into one track. Two mechanisms are implemented to resolve these ambiguities, which can be selected between using the TS/RESOLVE KUIP command. The first method (the default) is to only allow one segment per layer to be added to a cluster. In this case if a segment is associated with a cluster and there is already a segment from the same superlayer, one with the lowest total χ^2 with all other segments in the cluster is kept. The second method is to break up ambiguous clusters after the clustering is finished. This uses the routine TSLMAM, in which these 'super-clusters' are split into several tracks, with only one segment per pair of layers allowed in each track (with some attempt to use up all the segments). If the resultant split candidate tracks have enough segments in them (usually 3) then they are passed to the fitting stage.

2.4 Track fitting

The TS fitter can be used to fit (or refit) tracks found with the TS or other track finding algorithms. The input to the fitting program are the combined track banks /TRCT, the pointer bank /TRCP, and the raw data, reconstructed segment and/or coordinate banks referred to by /TRCP. As implemented within the TF package, the main fitting driver routine TSIFIT loops over all track banks, attempting to fit each with the routine TSLFIT. The initial fit parameters are retrieved from /TRCT. The individual hit data are unpacked with the routine TSTKHT into the common block /TSFITC/. Then, the fit proceeds in four stages:

1. First, the initially assigned hits are fit. The fit is iterated up to four times or until the χ^2 ceases to improve.
2. If the fit χ^2 is large (greater than $\chi^2_{MAX} = 50$), the point with the largest contribution to the χ^2 is dropped, and the track is refit. This is repeated up to 8 times, or until the χ^2 is below χ^2_{MAX} or there are too few points to refit.
3. Next, using the track parameters from the most recent fit, a road algorithm is used to search for additional hits in the silicon and other systems, refitting after each hit is added. The routine TSROAD controls the road search, using a different routine TSxxRD for each subsystem.
4. Finally, by default, a beam constraint is added to the fit if the impact parameter is less than 2 standard deviations from the beam, using the impact parameter and calculated error from the previous fit iteration. The beam constraint can be disabled with the TS/FIT/BEAM KUIP command.

The main fitting routine called once per iteration is TSITER. The method used is described in Reference 1, but including the multiple scattering errors from the silicon layers. First the residuals from the input track parameters are calculated by extrapolating the trajectory to each measurement layer. Next, the derivatives of the residuals at each layer with respect to each of the five track parameters are calculated. Then, the routine calculates the correlated multiple scattering error between each pair of measurement points due to the material in the silicon layers, which is added to the intrinsic measurement error at each layer. (Other material such as the beam pipe and outer tracking system has not yet been added.) Finally, an improved set of fit parameters and errors is calculated, which involves inverting a matrix with the dimensions of the number of measurement points.

The data which are used for the fit are different for the different detector subsystems. For the silicon strip system, the individual digitizations are fit, so an axial/stereo pair make up two distinct hits. The gas microstrip digitizations are also used, just as the silicon. For the straw tube system, each segment is used as an $r - \phi$ point. The stereo straw superlayers are treated analogously with the silicon stereo hits. A modest improvement might be achieved by fitting the straw digitizations, especially to clean up the final track, but adding too many points would slow the fit down substantially. For the pixel system, reconstructed coordinates are used in the fit, and each coordinate is treated as two uncorrelated measurements, one in $r - \phi$ and one in z. The scintillating fibers are also implemented as coordinates in the fit, though only the $r - \phi$ information is used at present.

After the final fit, a routine TSDUPL goes through all tracks and eliminates duplicates, which are defined as tracks that have nearly identical ϕ_0 and ρ and 50% of the hits from the track with fewer hits are on the other one.

APPENDIX A

KUIP Commands

FIND

Calls TSFIND subroutine

PLOT

Calls TSPLOT subroutine

ANAL

Calls TSANAL subroutine

PRINT print1 print2 print3

```
PRINT1     I 'Print level for segment finding' D=1 R=0:5
PRINT2     I 'Print level for segment clustering' D=1 R=0:5
PRINT3     I 'Print level for track fitting' D=1 R=0:5
```

Set debug print levels for segment finding, clustering, and track fitting.

READGEOM lrdgeom

```
LRDGEO     C 'Whether to read the geometry from the begin run record'
D='Y' R='Y,N'
```

Enable/disable reading silicon geometry from begin run record.

SEGMENTPARAMS

set parameter for finding segments in silicon (stored in sequence TSSEGC)

PTMINV ptminv

```
PTMINV     R 'Minimum pt used to define search road' D=0.5 R=0:
```

Minimum pt used to define search road. Defines the minimum pt used to look in adjacent superlayer for a hit to form a segment.

MKEPTV mkeptv

```
MKEPTV     I 'Maximum number of segments kept per hit' D=3 R=0:10
```

Maximum number of segments kept per hit. A small number of choices of possible segments are kept for each hit to increase efficiency. The code internally sets a limit of 10 due to array sizes.

TWOHITSEP i2sep

```
I2SEP      I 'Forced two hit separation (in number of strips)' D=0
```

Forced two hit separation (in number of strips). Before a hit is used in the reconstruction, it must be I2SEP number of strips from its neighbors. Was used to account for pulse spreading in old simulation.

STEREOMATCH tstbar tstfor

```
TSTBAR     R 'Stereo match parameter for barrel' D=50 R=0:
TSTFOR     R 'Stereo match parameter for forward' D=100 R=0:
```

Cut on the extrapolated z0 for possible combination. There is a separate cut for the barrel and forward. NOTE: This is no longer working in the current version in the code, a more complicated set of cuts is being used.

MULTSCAT lsegms

```
LSEGMS     C 'Flag to include mult scat contribution to errors' D='Y'
R='Y,N'
```

Flag to include mult scat contribution to errors. For the track clustering method of track finding, you want to include multiple scattering in the track parameter errors. Other methods don't want that, so allow it to be turned off.

USEAMB useamb

```
USEAMB     C 'Whether to use ambiguous coordinates to make segments'
D='Y' R='Y,N'
```

This command selects whether to use ambiguous coordinates to make segments. An ambiguous coordinate is if more than one stereo strip can be associated with an axial strip, in which case there may be two correct and two wrong coordinates for the two particles.

CLUSPARAMS

Set params for track finding (clustering method). (stored in sequence TSFNDC)

NONSTEREO lnonst chiflg

```
LNONST     C 'Whether to include non-stereo segments in finding' D='N'
R='Y,N'
CHIFLG     R 'Value of CHIZ that flags a non-stereo segment' D=999
```

Enable/disable including non-stereo (ie axial only) segments in the finding.

NOAMBIG lnoamb

```
LNOAMB     C 'Flag set when only one segment in superlayer is allowed'
D='Y' R='Y,N'
```

Flag set when only one segment in superlayer is allowed. This flag should be set true if one wants to allow only one segment per superlayer in a candidate track.

CHIMATCH chirmx chizmx

```
   CHIRMX     R 'Maximum chisq value allowed for Rvariables to join
cluster' D=10 R=0:
   CHIZMX     R 'Maximum chisq value allowed for Zvariables to join
cluster' D=10 R=0:
```

Max chisq for R,Z to allow segment to join cluster.

NSEGMENTS nsgclu

```
   NSGCLU     I 'Minimum number of segments in cluster to make a track'
D=3 R=0:
```

Minimum number of segments in a cluster to call it a track.

RESOLVE iresolv

```
   IRESOL     I 'Whether to resolve ambiguities during clustering' D=1
R=0:2
```

Enable code to attempt to resolve ambiguities in segment clustering. Options:

```
    0: Do not attempt to resolve ambiguities
    1: Attempt to resolve ambiguities as they arise during clustering in
TSLCLU.
       If two segments are found in a superlayer, choose between them
       rather than adding them both.
    2: Attempt to sort out ambiguities after clustering, in TSLMAM.
       If a cluster appears to be made from more than one track, break it
apart.
```

FITPARAMS

Set control parameters and flags for track fitting

ITERATE mxiter delch2

```
MXITER     I 'Maximum number of iterations' D=3 R=1:10
DELCH2     R 'Reduction of chi**2 to stop iteration' D=0
```

Set the maximum number of fit iterations. At each stage, fit will be iterated up to MXITER times, or until the reduction of chi**2 per degree of freedom is less than DELCH2.

ROAD usroad roadct [ladd1 ladd2 ladd3 ladd4 ladd5 ladd6 ladd7 ladd8 ladd9

```
USROAD     C 'Whether to use road to pick up hits' D='Y' R='Y,N'
ROADCT     R 'Maximum # sigma for hit to be added' D=100
LADD1      C 'Whether to add ST hits within track' D='Y' R='Y,N'
LADD2      C 'Whether to add ST hits inwards' D='Y' R='Y,N'
LADD3      C 'Whether to add ST hits outwards' D='Y' R='Y,N'
LADD4      C 'Whether to add PX hits' D='N' R='Y,N'
LADD5      C 'Whether to add axial WR segments' D='Y' R='Y,N'
LADD6      C 'Whether to add stereo WR segments' D='Y' R='Y,N'
LADD7      C 'Whether to add class 7 hits' D='N' R='Y,N'
LADD8      C 'Whether to add class 8 hits' D='N' R='Y,N'
LADD9      C 'Whether to add class 9 hits' D='N' R='Y,N'
LADD10     C 'Whether to add class 10 hits' D='N' R='Y,N'
```

Enable/disable road technique and set cut. After an initial fit using the originally assigned hits, other hits may be added along a road defined by the track parameters. Each type of hits to be added can be independently enabled or disabled.

DROP drphts chsqct maxdel

```
DRPHTS     C 'Whether to drop hits to improve a fit' D='Y' R='Y,N'
CHSQCT     R 'Chi**2 above which hits may be dropped' D=50
MAXDEL     I 'Maximum number of hits to be dropped' D=8
```

Enable/disable dropping of hits and set chi**2 cut. Choose whether to drop hits to improve a track fit, and set the minimum chi**2 to drop hits and maximum number of hits to drop before giving up.

BEAM lbmcon bmcon

```
LBMCON     C 'Whether to impose beam constraint' D='Y' R='Y,N'
BMCON      R 'Beam transverse size (cm)' D=0.1000000E-02
```

Enable/disable beam constraint for primary track candidates. Also set the size of the beam (default 10 microns).

EXTRAP lextrp

```
LEXTRP     C 'Whether to use outwards extrapolation feature' D='N'
R='Y,N'
```

Enable feature to alter the multiple scattering calculation to extrapolate outwards.

MULTSCAT fitmsc

```
FITMSC     C 'Whether to include multiple scattering in error matrix'
D='Y' R='Y,N'
```

Enable/disable including multiple scattering in the fit.

MINHITS mnhits

```
MNHITS     I 'Minimum number of hits + segments to start a fit' D=8
```

Set the minimum number of hits to start a fit.

STRAWSEGS cutrphi cutalfa maxalfa lambig

```
CUTRPH     R 'Maximum delta r*phi to pick up segments' D=0.5
CUTALF     R 'Maximum delta alpha to pick up segments' D=0.1
MAXALF     R 'Maximum abs(alpha) at a superlayer to pick up segments'
D=0.2
LAMBIG     C 'Whether to pick up the best segment if more than one'
D='Y' R='Y,N'
```

Set some parameters for adding straw segments with the road algorithm.

INEFFICIENCY

Set constants for implementing inefficiency in strips

CONSTANT tseffc

```
TSEFFC     R 'Constant inefficiency over whole silicon detector' D=0
R=0:1
```

Inefficiency applied uniformly to all digitizations.

RADIUS tseffr imod ilay [irunit]

```
TSEFFR     R 'Inefficiency value for the given mod, lay (readout
unit)' D=0 R=0:1
IMOD       I 'Module number, 0 means all' D=0 R=0:3
ILAY       I 'Layer number for this module, 0 meansall' D=0
IRUNIT     I 'Readout unit (for forward modules),0 means all' D=0
```

Apply inefficiencies as a function of radius: Specify value for given modules, layers and readout units (for forward).

STATUS

Print out current values of inefficiencies

APPENDIX B

Reconstruction bank descriptions

Note: the banks followed by an asterisk (*) are not currently implemented in the reconstruction and are potentially irrelevant.

```
[RECO]
|     |
|   3 |--[HTRK]
|     |   1 |---[STRE>                . silicon barrel tracker
|     |     |    1 |---[STRS>                . reconstructed layers
|     |     |      |    1 |---[STCO]         . space coordinates
|     |     |      |      |    1 |---[STCP]* . pointers to digitizations
|     |     |      |    2 |---[STSG>         . track segments
|     |     |      |      |    1 |---[STSP]  . pointers to digitizations
|     |     |    2 |---[STTK>*               . reconstructed tracks
|     |     |      |    1 |---[STTP]*        . pointers to segs or digis
|     |   2 |---[STIE>                . silicon forward tracker
|     |     |    1 |---[STRS>                . reconstructed layers
|     |     |      |    1 |---[STCO]         . space coordinate
|     |     |      |      |    1 |---[STCP]* . pointers to digitns
|     |     |      |    2 |---[STSG>         . track segments
|     |     |      |      |    1 |---[STSP]  . pointers to digitizations
                        .
                        .
                        .
                        .
|     |  14 |---[TRCT]>               . combined tracks
|     |     |    1 |---[TRCP>                . pointers from tracks
```

Tracking reconstruction header bank: (HTRK)

word #		contents	range	type
1	NCTK	# of tracks from combined subdetectors (# TRCT banks)	(1,10000)	integer
2	NWCT	# words/track in TRCT bank	(1, 50)	integer

Reconstruction subdetector header banks: (xxRE) one per module

word #		contents	range	type
1	NWCO	# words/coordinate in xxCO bank	(1, 10)	integer
2	NWSG	# words/segment in xxSG bank	(1, 50)	integer

3	NWTK	# words/track in xxTK bank	(1, 50)	integer
4	NWSC	# words/supercluster in SFSC only		integer

Reconstructed Superlayer banks: (xxRS) one per superlayer (outer detectors)
one per layer (inner detectors ST,PX)

word #		contents	range	type
1	IZMOD	z module #	(1:3)	integer
2	ISLAY	superlayer #	(1:31)	integer
3	NLAY	# layers in superlayer	(1:10)	integer
4	NCOR	# space coordinates	(1:10000)	integer
5	NSEG	# track segments	(1:5000)	integer

Reconstructed Coordinate banks: (xxCO) one per layer
Some subdetectors may go directly from digitizings to track segments skipping the reconstructed coordinate stage.

word #		contents	range	type
1	IWFDP	inclusive word count	(0:60000)	integer
2	ILAY	layer #	(1:31)	integer
3	NCOR	# of coordinates this layer	(0,10000)	integer

The next block is repeated for all coordinates.
Coordinates could in principle be x,y,z for some detectors; however, in general, the first 6 (real) words are:

....+1	R	radius	(0., 200.)	real
....+2	PHIP	azimuth (position)	(0.,2*pi)	real
....+3	Z	longitudinal coordinate	(-600., 600.)	real
....+4	DPHIP	variance on PHIP	(0., 1.E3)	real
....+5	DZ	variance on Z (barrel)	(0., 1.E3)	real
	DR	variance on R (intermediate)	(0., 1.E3)	real
....+6	DRZ	PHIP - Z covariance (barrel)	(-1.E3,1.E3)	real
	DRZ	PHIP - R covariance (interm)	(-1.E3,1.E3)	real

....+7 IWID Cluster widths and quality bits:

Bits	1: 8	Axial cluster width
Bits	9:16	Stereo cluster width
Bit	17	1 if axial hit used (redundant)
Bit	18	1 if stereo hit used (redundant)
Bit	19	1 if cluster is ambiguous

....+8 IPTR pointers to axial and stereo digitizations for this coordinate:

Bits	1:16	Sequential number of axial digitization in STDI
Bits	17:32	Sequential number of stereo digitization in STDI

At present, since the pointers are contained in STCO, the STCP bank is not created.
For forward detectors, errors will be on PHIP, R, and their correlation, instead of PHIP, Z and their correlation.

Pointers from Coordinates to digitizings (xxCP)

word #		contents	range	type
1	IWFDP	inclusive word count	(0:60000)	integer
2	ILAY	layer #	(1:31)	integer
3	NDIGL	# coordinates in this layer	(0:10000)	integer
for each coordinate in layer:				
.. +1	NHIT	# of digitizations contributing to this coordinate	(0:100)	integer
for each contributing digitizing:				
....+4	IDIG	# of digitizing within layer in the xxDI bank.	(0:65535)	integer

repeat for each digitizing, coordinate.
This depends on the order of pointers being the same as the order of coordinates in the xxCO bank.
digitizing address in xxDI bank would be 3 + (IDIG-1)*NWDI

Segment banks
The segments from the silicon system differ from the outer systems primarily because the information measured is different. The measured quantities for the silicon are essentially space points from the two hits in a wafer (one from each side). These are then paired to create the segments in a superlayer (two adjacent physical layers). On the other hand, the outer tracker proposals have physical superlayers which measure the phi position and the direction of a segment at a given radius. Therefore we have chosen to store different parameters in the banks for these cases.
/RECO/HTRK/STRE/STRS/STSG

```
CA)=================================================================
C     STSG : silicon detector segment bank (barrel)
CB)=================================================================
C   version no : 1
C     no. type  contents                                      range
C-------------------------------------------------------------------
       1   I     ISEG   segment identification number        (0,10000)
       2   I     NPT    number of points (axial+stereo)      (0,200)
```



```
 3   I     FITV   set bit for each variable used in fit.
                  (CERNLIB convention for bit numbering as
                  in SBIT and JBIT). Bits 1 to 6 correspond
                  to  R,B0,PHI0,RHO,Z0, TANL respectively.
                  At least one of these should be fixed.
                  (Usually B0 is.)
                  (Note: since an attempt is always made to estimate
                  the stereo information (by just using the center of
                  the wafer if no stereo hits are matched) the Z0,
                  TANL bits will be on. To check to see if this seg-
                  ment had both stereo hits, look at bit 17
                  (if = 1, had 2 stereo) or can check the number of
                  hits in the pointer bank.)
                  Bit 8 means the beam constraint was used.
                  By implication, # degrees of freedom
                  is NPT - (# of parameters in FITV).
                  Bits 13-16 are used to flag particular
                  conventions for choice of fit parameters:
                  13  parameters at centre of track (fixed R)
                  14  parameters at centre of track (fixed Z)
usual  -------->  15  parameters at closest point of approach to
                      origin (B0 = distance of closest approach)
                  16  parameters for forward silicon detector
                  bits 17-32 are fit procedure
                  (reconstruction package) specific.
                  For TS package:
                   bit 17=1 implies 2 stereo hits used
                     .
                     32 segment used on track?
 4   I     INDMC  MC indices for /PTRK/
                  track number in bits 1-16, ( JBYT etc
                  vertex number in bits 17-32.  convention)
     (The following evaluated at the point which is fixed)
     (Undetermined parameters (eg Z, TANL, RHO) are left zero.)
 5   R     R      radius at segment center **not a param(0., 200.)
 6   R     B0     (usually = 0)
 7   R     PHI0   azimuth at point of closest approach  (0., 2*pi)
 8   R     RHO    curvature (signed)                   (-1.E6, 1.E6)
 9   R     Z      longitudinal coordinate              (-600., 600.)
```

```
     10   R     TANL    tan(dip angle)                    (-1.E6, 1.E6)
     11   R     CHI2    segment chi-squared(axial+stereo)     (0., 1.E6)
     12   R     ALEN    projected segment length  (x-y)       (0., 200.)
     13   R     T0      fitted time (ns) (not used for ST) (-100., 100.)
     14   R     DT      error on fitted T0 (ns)               (0., 100.)
     15-  R     ERRM    (error matrix elements)**2 from fit
      29                lower triangular form, fit
                        parameters in same order as FITV.
                        ( Rows corresponding to fixed
                        parameters are suppressed). Note
                        diagonal terms are error squared.
 CC)-----------------------------------------------------------------------
C
C  daughter bank 1 :  STSP
C
C  created in     :    STFLSG
C  referenced in :    xxUPSG
C
C  originally defined by :  M. Corden, K. O'Shaughnessy
C  history :
C
C  long description :
C
CD)=====================================================================
/RECO/HTRK/STIE/STRS/STSG
CA)=====================================================================
C     STSG : silicon detector segment bank (forward)
CB)=====================================================================
C   version no : 1
C     no. type  contents                                        range
C-----------------------------------------------------------------------
       1   I     ISEG    segment identification number          (0,10000)
       2   I     NPT     number of points (axial+stereo)        (0,200)
       3   I     FITV    set bit for each variable used in fit.
                         (CERNLIB convention for bit numbering as
                         in SBIT and JBIT). Bits 1 to 6 correspond
                         to  R,B0,PHI0',RHO',Z0',TANL' respectively.
                         At least one of these should be fixed.
                         (Usually B0 is.)
                         (Note: since an attempt is always made to estimate
```



```
                         the stereo information (by just using the center of
                         the wafer if no stereo hits are matched) the Z0,
                         TANL bits will be on. To check to see if this seg-
                         ment had both stereo hits, look at bit 17
                         (if = 1, had 2 stereo) or can check the number of
                         hits in the pointer bank.)
                         Bit 8 means the beam constraint was used.
                         By implication, # degrees of freedom
                         is NPT - (# of parameters in FITV).
                         Bits 13-16 are used to flag particular
                         conventions for choice of fit parameters:
                         13  parameters at centre of track (fixed R)
                         14  parameters at centre of track (fixed Z)
                         15  parameters at closest point of approach to
                             origin (B0 = distance of closest approach)
usual  -------->         16  parameters for forward silicon detector
                         bits 17-32 are fit procedure
                         (reconstruction package) specific.
                         For TS package:
                          bit 17=1 implies 2 stereo hits used
                            .
                            32 segment used on track?
       4   I     INDMC   MC indices for /PTRK/
                         track number in bits 1-16, ( JBYT etc
                         vertex number in bits 17-32.  convention)
     (The following evaluated at the point which is fixed)
     (Undetermined parameters (eg Z, TANL, RHO) are left zero.)
       5   R     R       radius at segment center **not a param(0., 200.)
       6   R     B0      (usually = 0)
       7   R     PHI0 -RHO*Z/(2*S) transformed equivalent to
                         azimuth at point of closest approach  (0., 2*pi)
       8   R     RHO/S   curvature (signed)/tan(lambda)     (-1.E6, 1.E6)
       9   R     -Z/S    longitudinal coordinate/tanl       (-600., 600.)
      10   R     1/S     1/tan(dip angle)                   (-1.E6, 1.E6)
      11   R     CHI2    segment chi-squared(axial+stereo)     (0., 1.E6)
      12   R     ALEN    projected segment length  (x-y)       (0., 200.)
      13   R     T0      fitted time (ns) (not used for ST) (-100., 100.)
      14   R     DT      error on fitted t0 (")                (0., 100.)
```

```
     15-  R     ERRM    (error matrix elements)**2 from fit
      29                lower triangular form, fit
                        parameters in same order as FITV.
                        ( Rows corresponding to fixed
                        parameters are suppressed). Note
                        diagonal terms are error squared.
 CC)-----------------------------------------------------------------------
C
C  daughter bank 1 :  STSP
C
C  created in     :    STFLSG
C  referenced in :    xxUPSG
C
C  originally defined by :  K. O'Shaughnessy
C  history :
C
C  long description :
C
CD)=====================================================================
-----------------------------------------------------------------------
Reconstructed Track segment pointers:  (xxSP)
word #          contents                          range      type
 1    IWFDP    inclusive word count             (0:60000)  integer
 2    NCOR     # coordinates/digitizings this    (0,255)   integer
              track segment
for all coordinates/digitizings this segment
 3    ICOR     sequence # of coordinate/digitizing (0:65535) bits 1:16
              within layer in xxCO or xxDI bank
              layer #                              (1:10) bits 17:20
              superlayer #                         (1:31) bits 21:25
              z module #                           (1: 3) bits 26:27
              stereo flag (ST only)                (0: 1) bit     32
repeated for each coordinate/digitizing
so coordinate address in xxCO bank would be  3 + (ICOR-1)*NWCO
-----------------------------------------------------------------------
Reconstructed Combined Track banks:  (TRCT)
This is still before vertex finding and association of tracks
to individual vertices. Track quantities therefore remain in
"tracking" variables.
NB  it may be desirable to insert a dummy bank in between the
```



```
TRRE and TRCT banks, so that combined and individual subdetector
tracks remain at the same tree depth.
/RECO/HTRK/TRCT
CA)=====================================================================
C     TRCT : track parameter bank for distance of closest approach
CB)=====================================================================
C   version no : 1
C     no. type  contents                                          range
C---------------------------------------------------------------------
       1   I     ITRK   track identification number            (0,10000)
       2   I     NPT    number of points (axial+stereo)         (0,200)
       3   I     FITV   set bit for each variable used in fit.
                        (CERNLIB convention for bit numbering as
                        in SBIT and JBIT). Bits 1 to 6 correspond
                        to ALPHA, B0,PHI0,RHO,Z0,TANL respectively.
                        At least one of these should be fixed.
                        Bit 8 means the beam constraint was used.
                        (Not being set by TS algorithm)
                        By implication, # degrees of freedom
                        is:NPT - (# of parameters in FITV).
                        Bits 13-16 are used to flag particular
                        conventions for choice of fit parameters:
                        13  parameters at centre of track (fixed R)
                        14  parameters at centre of track (fixed Z)
      usual -------->   15  parameters at closest point of approach to
                            origin (B0 = distance of closest approach)
                        16  parameters for forward silicon detector
                        bits 17-32 are fit procedure
                        (reconstruction package) specific.
                        eg 17 road used to pick up extra hits?
                           .
                          32 segment used on track?
       4   I     INDMC   MC indices for /PTRK/
                         track number in bits 1-16, ( JBYT etc
                         vertex number in bits 17-32.  convention)
          (The following evaluated at distance of closest approach)
       5   R     ALPHA   alpha=(+-pi/2)                      (-pi/2,pi/2)
       6   R     B0      impact parameter (signed)
       7   R     PHI0    azimuth at DCA0                       (0., 2*pi)
```

```
 8   R    RHO     curvature (signed)                (-1.E6, 1.E6)
 9   R    Z0      longitudinal coordinate at DCA0   (-600., 600.)
10   R    TANL    tan(dip angle)                    (-1.E6, 1.E6)
11   R    CHI2    track   chi-squared                 (0., 1.E6)
12   R    ALEN    projected track length  (x-y)       (0., 200.)
13   R    T0      fitted time (ns)                  (-100., 100.)
14   R    DT      error on fitted t0                  (0., 100.)
15-  R    ERRM    (error matrix elements)**2 from fit
 29               lower triangular form, fit
                  parameters in same order as FITV.
                  ( Rows corresponding to fixed
                  parameters are suppressed). Note
                  diagonal terms are error squared.
```

The following word has been proposed but not yet implemented in all packages. This would be a backwards incompatible change which would make previously reconstructed data unreadable. Probably, if this is added, a few additional "user" words will be added.

```
 30  R    CHIMATCH Chi**2 match between reconstructed track
                   and Monte-Carlo particle parameters.
 CC)-----------------------------------------------------------------------
C
C  daughter bank 1 :  TRCP
C
C  created in    :
C  referenced in :
C
C  originally defined by :
C  history :
C
C  long description :
C
CD)=====================================================================
Reconstructed Combined Track pointers:  (TRCP)
word #          contents                      range      type
 1    IWFDI   inclusive word count           (0:60000)  integer
 2    NSDET   # of subdetectors                (0, 7)   integer
for each subdetector:
 3    NWSDET  # of words this subdetector      (0,  )   integer
 4    ISDET   subdetector identifier           (1,  )   integer
```



```
              (+ stereo flag?)
 5    NSEG    # segments this subdetector     (0,  63)  integer
 6    NCOR/   # coordinates/digitizings this  (0, 255)  integer
      NDIG    subdetector
for all segments this subdetector,
 7    ISEG    segment #                     (0,16383)   bits  1:14
              superlayer #                    (1:31)    bits 21:25
              z module #                      (1: 3)    bits 26:27
       .
       .
6+NSEG ISEG
for all coordinates/digitizings this subdetector
6+NSEG+1 ICOR coordinate sequence # within  (0,65535)   bits  1:16
              layer in xxCO (or xxDI) bank
              layer #                         (1:10)    bits 17:20
              superlayer #                    (1:31)    bits 21:25
              z module #                      (1: 3)    bits 26:27
       .
       .
6+NSEG+NDIG/NCOR
so coordinate address in xxCO bank for this superlayer would be
   3 + (ICOR-1)*NWCO +
repeated for all subdetectors
```

APPENDIX C

Utility routines

There are a few routines in the package that might prove to be useful to others.

The first set is a collection of statement functions used to access the geometry data. They are documented in the sequence SIFUNC in the blankdek in patch STUTIL in ST.CAR. For example SIMNLA(mod) will give the number of layers in the module mod.

Some other routines unpack the silicon information into temporary space for faster access. They are: TSUPDI - unpack the digitizations TSUPSG - unpack segments (all types indicated by bits in TF control word) TSUPTK - unpack track banks

In addition, there is a routine which converts segments from the outer barrel format or the forward silicon format to the silicon barrel format. It is called TSCONV (see code for arguments).

Program Flow

```
Initialization:
TSDEFK [defines KUIP commands, called from TFDEFK
  |     written by KUIPC processor from deck TSCDF]
  |
  |-----> KUIP routines KUNWG,KUCMD,KUGUID,KUCMD,KUACT
          Other externals, callable via KUIP commands:
          TSFIND [cluster segments to make tracks]
          TSPLOT [Monitor performance of reconstruction code]
          TSANAL [sample analysis routine]
          TSPSEG [print out the segment finding parameters]
          TSGSEG [read in segment finding parameters]
          TSPFND [print status of track finding parameters]
          TSGFND [read in segment finding parameters]
          TSPFIT [print status of track fitting parameters]
          TSGFIT [set fit flags and parameters]
          TSKUIP [set parameters and logical switches from KUIP]
TSINIT [Program initialization, called from TFINIT]
  |
  |------> STTRKR [in ST package, set up silicon detector geometry]
  |------> TSCONS [define superlayer structure and other constants]
  |------> TSMSM  [calculate moments for multiple scattering errors]
```



```
TSRUNB [Run initialization, called from TFRUNB]
  |
  |------> TSRGEO [read silicon detector geometry from begin run record]
  |------> TSCONS [define superlayer structure and other constants]
  |------> TSMSM  [calculate moments for multiple scattering errors]
  |------> TSPSEG [print out the segment finding parameters]
  |------> TSPFND [print status of track finding parameters]
  |------> TSGFND [read in segment finding parameters]
  |------> TSPFIT [print status of track fitting parameters]
Event by event routines:
TSEVIN  [event initialization for TS reconstruction, called by TFEVNT]
  |
  |------>TSUPDI [unpack digits into temp. common]
  |         |
  |         |------>shell routines SHTSCR,SHTSPR,SZGET,SHTSDL,SUERR
  |
  |------>TSFLHD [fill header banks if they don't exist already]
TSFLCO [reconstruct silicon coordinates from raw data, called by TFEVNT]
  |
  |------>STCOOR [in ST package, to calculate coordinate positions]
  |------>shell routines SHTSCR,SZFILL,SHTSDL
TSSEGF [reconstruct silicon segments from coordinates, called by TFEVNT]
  |
  |------>TSUPCO [unpack coordinate banks into temporary array]
  |------>UTGETP [in UT package, get MC particle parameters]
  |------>PHIDIF [in UT package, delta phi between 2 phi angles]
  |------>TSVPAR [Get vector parameters and error matrix]
  |         |
  |         |----->STCOOR [calculate coordinates from hits]
  |------>TSFLSG [fill segment banks from info in TSFVEC ]
  |         |
  |         |----->STCOOR [calculate coordinates from hits]
  |         |----->shell routines SZFILL,SUERR
  |------>UTMATC [match segment with MC particles]
  |------>shell routines SUHBK,SZFILL,SZGET,SUERR
TSFIND  [find tracks by clustering segments]
  |
  |
```

```
|------>TSUPSG [unpack the segment banks into temp storage]
|         |
|         |----->TSCONV [convert outer tracker segments
|         |      to silicon barrel segments or inverse]
|         |----->shell routines SZNUM,SZGET,SHTSCR
|
|------>TSLCLU [Find clusters amongst the vectored hits]
|         |
|         |------>SHTSCR [allocate temp space]
|         |------>TSPGET [Get pointer,layer for vector]
|         |------>TSVASS [decide if 2 vectors should be associated]
|                  |
|                  |----->TSVCH2 [calculate chisq for 2 or more
|                      vectors to come from the same track]
|
|------>TSLOAM [Now look for 0-ambiguity clusters, make tracks]
          |
          |------>TSPGET [Get pointer,layer for vector]
          |------>TSLMAM [Resolve multiple track cluster ambiguities]
          |------>TSVCH2 [calculate chisq for 2 or more vectors
          |                          to come from the same track]
          |------>TSVTAG [fill track and pointer bank (tag cluster)]
                   |
                   |----->SZFILL
                   |
                   |----->UTMATC
TSIFIT [fit and clean up tracks, called from TFEVNT]
  |
  |------>TSUPTK [unpack found track banks]
  |         |
  |         |------>LNOPAD [string length, excluding trailing blanks]
  |         |------>shell routines SZNUM,SHTSCR,SZGET
  |
  |------>TSLFIT [fit a specific track]
  |         |
  |         |----->TSTKHT [unpack hit information for specified track
  |         |        |     into the TSFITC common block]
  |         |        |
```



```
|         |        |----->TSSTHT [get silicon hit info]
|         |        |----->TSWRHT [get straw hit info]
|         |        |----->TSPXHT [get pixel hit info]
|         |        |----->TSPMHT [get gas microstrip hit info]
|         |        |----->TSSFHT [get sci-fi hit info]
|         |
|         |----->TSITR8 [fit iteration]
|         |        |
|         |        |---->TSITER [do beam constrained fit]
|         |                |
|         |                |----->TSSCAT [calculate mult scat]
|         |                |----->STBPOS [in ST package, extrapolate
|         |                |              to silicon barrel]
|         |                |----->STFPOS [in ST package, extrapolate
|         |                |              to silicon forward]
|         |                |----->PRFPOS [in PR package, extrapolate
|         |                |              to gas microstrips]
|         |                |----->TSPROP [extrapolate to r,phi,z]
|         |                |----->INVERT [invert measurement error matrix]
|         |                |        |
|         |                |        |---->MDINV[invert matrix]
|         |                |----->STPMAT [print a matrix]
|         |
|         |----->TSDELH [kill hit with largest residual]
|         |----->TSSORT [sort hits on track by radius]
|         |----->TSROAD [road search to pick up new hits]
|                  |
|                  |----->TSSTRD [road algorithm in silicon]
|                  |        |
|                  |        |----->TSNPOS [find num. of possible hits for
|                  |                |    track with given helix parameters,
|                  |                |    specify which dets/strips are hit]
|                  |                |----->STBPOS,STFPOS [track position]
|                  |----->TSWRRD [road algorithm in straws]
|                  |----->TSPXRD [road algorithm in pixels]
|                  |----->TSPMRD [road algorithm in gas microstrips]
|                  |----->TSSFRD [road algorithm in sci-fi]
|
```

```
  |------>TSDUPL [drop duplicate tracks]
  |------>shell routine SZFILL
TSRLSE [drop temporary space allocations, called from TFEVNT]
            |
            |------>shell routine SHTSDL
TSPLOT [Monitor performance of reconstruction code, called from TFEVNT]
  |
  |------>shell routines SUFSTA,SUOPEN (first call only)
  |------>UTFILL [in UT package, unpack MC particles]
  |------>UTCLUS [in UT package, cluster MC particles]
  |------>PHIDIF [min. diff. between 2 phi angles]
  |------>TSUPDI [Unpack track banks into temporary storage]
  |------>TSDSEG  [fill diagnostic histograms about the segments]
  |         |
  |         |------>TSUPSG [unpack segments banks to temp storage]
  |         |------>TSPGET [Get pointer,layer for vector]
  |         |------>TSPLSG [plot segments in kappa-phi space]
  |         |------>SHTSCR,SHTSDL to drop temp. space
  |
  |------>TSUPTK [unpack tracks into dynamic common]
  |------>TSTKPR [get requested parameter for specified track ]
  |------>TSTKHT [get hit information for specified track,hit]
  |------>TSNPOS [find expected hits,detectors given helix params]
  |------>TSRLSE [Drop any dynamically allocated commons]
Termination routines:
TSEFF  [divide histograms to get efficiencies, called by TFFINI]
  |
  |------>HDIVE [Divide 2 hists to get eff hist with correct errors]
            |------>HGIVE,HEXIST,HBOOK2,HBARX,HBARY,HBOOK1,HUNPAK,
            |       HPAK,HPAKE
            |------>SHTSCR,SHTSDL
            |------>HCALEF [to do the efficiency calculation]
Miscellaneous routines:
TSERUP  [unpack segment error matrix into full expanded form]
        [called by TSUPSG]
TSANAL [sample analysis job for higgs to 4 leptons]
  |
  |------>UTFILL [in UT package, unpack MC particles]
```



```
  |------>FROMZ  [determine whether lepton comes from a Z]
  |------>TSUPTK [unpack tracks into dynamic common]
TSSTFT [refit track with silicon information only]
  |
  |------>TSLFIT [fit a specific track]
```

REFERENCES

[1] SDC-NOTE-90-076, A.J. Weinstein

[2] S. Kunori, "SDC SHELL User's Guide", file SHELL_USERS_GUIDE.DOC, included with the SDC shell software.

[3] B. Hubbard and K. O'Shaughnessy, *Description of the Silicon Tracking Simulation*, SCIPP 92/17, SDC-92-208, June 2, 1992.