AUTOMATED SKID DATA SYSTEM

by

S. N. Runkle
Research Analyst

and

C. T. Gundlach
Graduate Research Assistant

Virginia Highway & Transportation Research Council
(A Cooperative Organization Sponsored Jointly by the Virginia
Department of Highways & Transportation and the University of Virginia)

Charlottesville, Virginia

June 1975
VHTRC 75-R69

2496

SUMMARY

The Data Systems and Analysis Section of the Virginia Highway & Transportation Research Council has been involved for sometime in the development of automated information systems for various types of roadway data. One of these systems, and the system discussed in this report,is the one for the collection, storage, and retrieval of skid data collected during normal survey skid testing.

Included in this report is a description of the data files maintained for skid data and the computer programs developed to edit input data, maintain the data files, and provide skid data output listings.

AUTOMATED SKID DATA SYSTEM

by

S. N. Runkle
Research Analyst

and

C. T. Gundlach
Graduate Research Assistant

INTRODUCTION

The data Systems and Analysis Section of the Virginia Highway and Transportation Research Council has been involved for some time in the development of automated information systems for various types of roadway data. A general overview of this work is presented in the Council publication entitled "Integrated Data Systems — Summary Report."(1) As indicated in that report, part of the effort has been to develop an automated data system for the collection, storage, and retrieval of skid data collected during normal survey skid testing.

In essence, three permanent skid data files are maintained. These are the mainline survey skid file, (RCØ.SKIDSURV), the bridge skid file, (RCØ.SKIDBRIJ), and the loop skid file, (RCØ.SKIDLOOP). As the names imply, the mainline survey skid file contains all skid data collected on the mainline portions of a highway (i.e. the service roads, ramps, etc. are excluded), and the bridge skid file contains skid data collected on bridges where the surface is different from that of the adjacent pavement. (When the bridge surface is the same as that of the adjacent pavement the data are contained in the mainline skid file.) The loop skid file contains data from several specified test sites which are frequently tested for data control purposes.

The flow diagram shown in Figure 1 indicates in general terms the total skid system and how it is used interactively with other systems.

PURPOSE AND SCOPE

The purpose of this report is to present the computer programs developed to maintain the skid data system as indicated by solid lines in Figure 1. Instructions regarding the function and use of each of the programs as well as listings for each of the programs are given.

The items in Figure 1 indicated by dotted lines will be completed in ongoing projects and will be discussed in future reports on the pavement system and integrated applications. Each of these items was discussed in some detail in "Integrated Data Systems — Summary Report."

4110

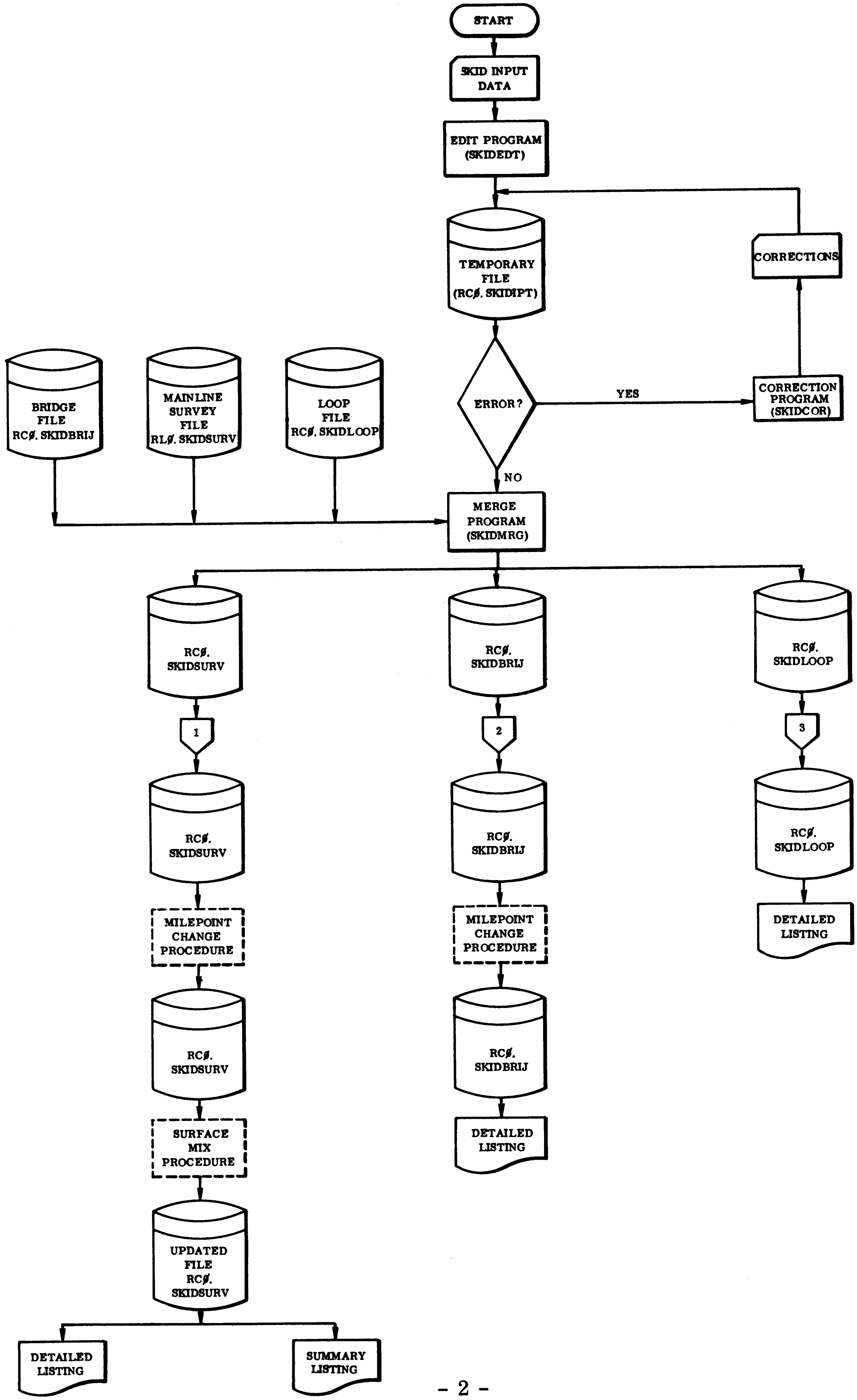

Figure 1. General flow diagram of skid system.

DISCUSSION OF PROGRAMS

This section contains a discussion of the general function and method of use for each of the programs developed thus far, including the JCL (Job Control Language) necessary to run the programs on the IBM-370. It should be noted that the JCL shown is that necessary to run the programs as they exist on the Research Council's library. In order to make the programs completely operational they will have to be placed in the proper library with the necessary changes in the JCL.

Listings for each of the programs discussed are contained in Appendix A.

Program SKIDEDT

The SKIDEDT program reads cards punched in accordance with the coding instructions explained in Appendix A of the report "Test Procedures and Data Input Techniques for Skid Testing" by Stephen N. Runkle.(2) The data items punched and their respective column locations are shown in Table 1. The edit program checks the input data as described in Table 1 and performs a compatibility check to ensure that the district, residency, and county are compatible.

City or Town is coded only when the route tested is a commercial (business) route. Otherwise the County is recoded in these columns. Thus, a compatibility check could be made to ensure that a city or town is coded when a commercial route is coded, but the program presently does not perform this function.

Date type indicates whether the data are mainline survey, bridge, or test loop data. If the data are for a loop as indicated by "L", the program automatically inserts the correct lane and direction corresponding to the test site indicated in columns 6-9.

It should be noted that the program requires input in all 52 columns. For certain data items as indicated in the code manual mentioned above zero is an acceptable input when no data is actually collected.

If an error is detected by any of the edit functions the entire 52 columns of data for the card containing the error are printed along with the sequence number for that card (figured in ascending order), beginning with the first input card and the nature of the error. Upon completion of the processing of all the records, the number of bad records, number of errors, and the total number of records processes are printed. A disk file labeled RCØ. SKIDIPT and containing all of the input records is produced. If there are errors on this file, the program terminates with a message instructing the user to run program SKIDCOR. Otherwise, the file is sorted according to the following keys in the ascending order designated.

1. District
2. Residency
3. County
4. City-Town
5. Route-Aux
6. Route-No-A
7. Direction
8. Lane
9. Milepoint
10. Date-Year
11. Date-Month

Following the sort, a message is printed informing the user that no errors exist on the file and that program SKIDMRG should be run.

Table 2 gives an example of the JCL necessary to run the program.

TABLE 1

Data Items Input and Edit Checks Performed

Columns	Data Item	Edit Performed
1	District	Insures data are numeric in the range 1 ≤ 8.
2-3	Residency	Insures data are numeric in the range 1 ≤ 58.
4-5	County	Insures data are numeric.
6-9	Route Number or Test Loop Site	Insures data are numeric in column 7-9.
10-12	City/Town/County	Insures data are numeric.
13	Test Vehicle	Insures data are numeric in the range 0 ≤ 3.
14	Test Wheel	Insures data are numeric equal to 1.
15-17	Calibration	Insures data are numeric.
18-19	Operators	Insures data are numeric.
20-25	Date	Insures data are numeric.
26-27	Time	Insures data are numeric in the range of 0 ≤ 24.
28	Weather Condition	Insures data are numeric in the range of 0 ≤ 8.
29-31	Air Temperature	Insures data are numeric.
32-34	Surface Temperature	Insures data are numeric.
35-36	Tread Depth	Insures data are numeric in the range of 0 ≤ .35.
37	Direction	Insures data are numeric in the range of 1 ≤ 5.
38	Lane	Insures data are numeric in the range of 1 ≤ 6.
39	Data Type	Insures data are 0, 3, L, X or Y.
40-43	Milepoint	Insures data are numeric in the range 0 ≤ 55.
44-46	Speed	Insures data are numeric in the range 15 ≤ 70.
47-49	Skid Number	Insures data are numeric in the range 0 ≤ 80.
50	Pavement Condition	Insures data are numeric in the range 0 ≤ 3.
51-52	Time Since Last Rain	Insures data are numeric.

TABLE 2

JCL for SKIDEDT

Card No.	Job Control Language
1	//SKIDEDT JOB (Accounting Information)
2	//GO EXEC PGM=SKIDEDT
3	//STEPLIB DD DSNAME=RCØ.PROGLIB,DISP=SHR
4	//OUTPUT DD SYSOUT=A,DCB=(RECFM=FB,LRECL=120,BLKSIZE=120)
5	//TAPE1 DD DSN=RCØ.SKIDIPT,UNIT=SYSDA, DISP=(NEW,CATLG,DELDE),
6	// DCB=(LRECL=80,RECFM=FB,BLKSIZE=1200),SPACE=(TRK,(5,2),RLSE),
7	// VOL=SER=VHDPØØ
8	//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
9	//SORTWKØ1 DD UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE)
10	//SORTWKØ2 DD UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE)
11	//SORTWKØ3 DD UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE)
12	//SORTWKØ4 DD UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE)
13	//SORTWKØ5 DD UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE)
14	//SORTWKØ6 DD UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE)
15	//SYSOUT DD SYSOUT=A
16	//CARDS DD *
17 through n	DATA DECK
n+1	/*
n+2	//

Program SKIDCOR

The SKIDCOR program corrects records on the temporary file created by the SKIDEDT program. It also allows for records to be added to or deleted from the temporary file as will be discussed below.

To make a correction, two cards must be input with the first card being an exact duplicate of the entire incorrect record (all 52 columns) shown on the error listing produced by SKIDEDT. The second card must immediately follow the first card (incorrect record), which seems obvious, but is important since all corrections may be made in a single run, and thus, at times, several input cards are required. All 52 columns of data must be contained in their correct coded form on the second card (even if only one column is in error). In addition to having correct data following the incorrect record, each incorrect record must be corrected in the order it appears on the file. For instance, the second record in error on the file must be matched to the second correction made to the file. In the event the correction cards are not paired as discussed, the program terminates and prints this information.

To delete a record, only one card of input is necessary. This card should contain all 52 columns of data for the record to be deleted exactly as the record occurs on the file, and, in addition, must have the word "DELETE" punched in columns 70-75. Deletion input cards must be intermingled with correction cards such that the correct sequence of corrections and/or deletions is maintained. For instance, if two records in a file are to be corrected and a following record deleted, the correction inputshouldoccur first and the deletion input would be third. Since two cards are required per correction, the deletion card would be the fifth input card.

To add a record, a single input card is needed. It would contain the correct 52 columns of data and the word "ADD" in columns 70-72. The addition cards should follow all correction and deletion cards..

Once corrections, deletions and additions are made to the temporary file, it is again edit checked in the same manner as explained above for SKIDEDT if any errors are listed, SKIDCOR must be rerun to make corrections. For each run of SKIDCOR the number of bad records, number of errors, and number of records processed are listed.

When no errors aredetectedon the temporary file, SKIDCOR sorts the file in the same manner as discussed above for SKIDEDT, and a message is printed instructing the user to run SKIDMRG.

Sample JCL for executing SKIDCOR is shown in Table 3.

TABLE 3

JCL for SKIDCOR

Card No.	Job Control Language
1	//SKIDCOR JOB (Accounting Information)
2	//GO EXEC PGM=SKIDCOR
3	//STEPLIB DD DSNAME=RCØ.PROGLIB,DISP=SHR
4	//OUTPUT DD SYSOUT=A,DCB=(RECFM=FB,LRECL=120,BLKSIZE=120)
5	//TAPE1 DD DSN=RCØ.SKIDIPT,UNIT=SYSDA,DISP=(OLD,KEEP)
6	//TAPE2 DD DSN=RCØ.TEMP1,UNIT=SYSDA,DISP=(NEW,DELETE),
7	// DCB=(LRCEL=80,RECFM=FB,BLKSIZE=1200),SPACE=(TRK,(5,2)),
8	// VOL=SER=VHDPØØ
9	//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
10	//SORTWKØ1 DD UNIT=SYSDA,SPACE=(TRK,(5,2),RLSE)
11	//SORTWKØ2 DD UNIT=SYSDA,SPACE=(TRK,(5,2),RLSE)
12	//SORTWKØ3 DD UNIT=SYSDA,SPACE=(TRK,(5,2),RLSE)
13	//SORTWKØ4 DD UNIT=SYSDA,SPACE=(TRK,(5,2),RLSE)
14	//SORTWKØ5 DD UNIT=SYSDA,SPACE=(TRK,(5,2),RLSE)
15	//SORTWKØ6 DD UNIT=SYSDA,SPACE=(TRK,(5,2),RLSE)
16	//SYSOUT DD SYSOUT=A
17	//CARDS DD *
18 through n	DATA DECK
n+1	/*
n+2	//

Program SKIDMRG

The purpose of SKIDMRG is to read the sorted file created by SKIDEDT and corrected as necessary by SKIDCOR, to write the respective loop, survey, and bridge records onto three separate temporary files, and then to merge these temporary files with the permanent files RCØ.SKIDSURV, RCØ.SKIDLOOP, and RCØ.SKIDBRIJ. The temporary files are labled RCØ.SKIDTSUR for survey data, RCØ.SKIDTLUP for loop data, and RCØ.SKIDTBDG for bridge data. During the creation of the temporary file for loop data it is resorted to interchange the date and milepoint keys. The program consists of a source deck that calls on various utility routines available through the JCL.

The source program terminates by printing the number of records written on each file, and by issuing a return code that indicates within the JCL which merge utilities will be executed. Thus, if some of the temporary files contain no records the return code will indicate that the merge utility for that file will not be executed. Following the termination of the source programs and prior to the merging, three backup files are created which are copies of the existing permanent files. These files are labled RCØ.SKIDBPLU for loop data, RCØ.SKIDBPSV for mainline survey data, and SKIDBPBG for bridge data.

The JCL necessary to execute the source program and utility programs is shown in Table 4.

Program SKIDUPT

As indicated earlier under purpose and scope, those items in Figure 1 shown by dotted lines will be completed as the work on the pavement system and integrated applications is completed, and they will be discussed in future reports covering that work. In brief terms, the milepoint change procedure will be utilized on the skid system to alter milepoint data on records for routes where milepoint changes have occurred for some reason (relocation, dual-dividing of existing two-lane road, etc.). At times data in the record other than milepoints will be altered as, for instance, the lane and direction data for an existing record when an existing two-lane road is dual-divided. The surface mix procedure will be utilized on the mainline survey skid file for update purposes such that data in the file not relating to a current surface mix will be removed from the file.

Since these processes have not been completed, the program SKIDUPT was developed as a temporary substitute. While SKIDUPT is much less efficient in that it requires manual interpretation of data and detailed input relating to each file record altered or removed, it will perform the milepoint change and update functions and, of course, can be utilized to correct existing file records.

As a matter of interest, no milepoint change procedure is indicated in Figure 1 on the skid loop data file. Milepoint information for loop sites is meaningless in that they are specified sites and thus no need exists to alter milepoint data.

TABLE 4

JCL for SKIDMRG

Card No.	Job Control Language
1	//SKIDMRG JOB (Accounting Information)
2	//GO EXEC PGM=SKIDMRG
3	//STEPLIB DD DSNAME=RCØ.PROGLIB,DISP=SHR
4	//OUTPUT DD SYSOUT=A,DCB=(RECFM=LRECL=120,BLKSIZE=120)
5	//TAPE1 DD DSN=RCØ.SKIDIPT,UNIT=SYSDA,DISP=(OLD,DELETE,KEEP)
6	//TAPE2 DD DSN=RCØ. SKIDTSUR,UNIT=SYSDA,DISP=(OLD,PASS)
7	//TAPE3 DD DSN=RCØ.SKIDTLUP,UNIT=SYSDA,DISP=(OLD,PASS)
8	//TAPE4 DD DSN=RCØ.SKIDTBDG,UNIT=SYSDA,DISP=(OLD,PASS)
9	//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
10	//SORTWKØ1 DD UNIT=SYSDA,SPACE=(TRK,(S,1)
11	//SORTWKØ2 DD UNIT=SYSDA,SPACE=(TRK,(S,1)
12	//SORTWKØ3 DD UNIT=SYSDA,SPACE=(TRK,(S,1)
13	//SYSOUT DD SYSOUT=A
14	/*
15	//STEP1 EXEC PGM=IEBGENER
16	//SYSPRINT DD SYSOUT=A
17	//SYSIN DD DUMMY
18	//SYSUT1 DD DSN=RCØ.SKIDLOOP,UNIT=SYSDA,DISP=(OLD,PASS)
19	//SYSUT2 DD DSN=RCØ.SKIDBPLU,UNIT=SYSDA,DISP=(OLD,PASS)
20	/*
21	//STEP2 EXEC PGM=IEBGENER
22	//SYSPRINT DD SYSOUT=A
23	//SYSIN DD DUMMY
24	//SYSUT1 DD DSN=RCØ.SKIDSURV,UNIT=SYSDA,DISP=(OLD,PASS)
25	//SYSUT2 DD DSN=RCØ.SKIDBPSV,UNIT=SYSDA,DISP=(OLD,PASS)

4123

TABLE 4 (continued)

Card No.	Job Control Language
26	/*
27	//STEP3 EXEC PGM=IEBGENER
28	//SYSPRINT DD SYSOUT=A
29	//SYSIN DD DUMMY
30	//SYSUT1 DD DSN=RCØ.SKIDBRIJ,UNIT=SYSDA,DISP=(OLD,PASS)
31	//SYSUT2 DD DSN=RCØ.SKIDBPBG,UNIT=SYSDA,DISP=(OLD,PASS)
32	/*
33	//STEP4 EXEC SORTD,COND=((100,EQ,GO),(400,EQ,GO),(500,EQ,GO))
34	//SORTINØ1 DD DSN=RCØ.SKIDTLUP,UNITS=SYSDA,DISP=(OLD,KEEP)
35	//SORTINØ2 DD DSN=RCØ.SKIDBPLU,UNIT=SYSDA,DISP=(OLD,KEEP)
36	//SORTOUT DD DSN=RCØ.SKIDLOOP,UNIT=SYSDA,DISP=(OLD,KEEP)
37	//SYSIN DD *
38 39	MERGE FIELDS=(1,5,A,10,3,A,6,4,A,37,2,A,24,2,A,20,2,A,40,4,A), FORMAT=BI
40	END
41	/*
42	//STEPS EXEC SORTD,COND=(400,LT,GO)
43	//SORTINØ1 DD DSN=RCØ.SKIDTSUR,UNIT=SYSDA,DISP=(OLD,KEEP)
44	//SORTINØ2 DD DSN=RCØ.SKIDVPSV,UNIT=SYSDA,DISP=(OLD,KEEP)
45	//SORTOUT DD DSN=RCØ.SKIDSURV,UNIT=SYSDA,DISP=(OLD,KEEP)
46	//SYSIN DD *
47 48	MERGE FIELDS=(1,5,A,10,3,A,6,4,A,37,2,A,40,4,A,24,2,A,20,4,A), FORMAT=BI
49	END
50	/*
51	//STEP6, EXEC SORTD,COND=((200,EQ,GO),(400,EQ,GO),(600,EQ,GO))
52	//SORTINØ1 DD DSN=RCØ.SKIDTBDG,UNIT=SYSDA,DISP=(OLD,KEEP)
53	//SORTINØ2 DD DSN=RCØ, SKIDBPBG,UNIT=SYSDA,DIST=(OLD,KEEP)
54	//SORTOUT DD DSN=RCØ.SKIDBRIJ,UNIT=SYSDA,DISP=(OLD,KEEP)
55	//SYSIN DD *
56 57	MERGE FIELDS=(1,5,A,10,3,A,6,4,A,37,2,A,40,4,A,24,2,A,20,4,A), FORMAT=BI
58	END
59	/*
60	//

The program SKIDUPT has the facility to alter or delete records, but cannot add records to a file. Only one file can be manipulated per run, and the user must know the information listed below for each record to be altered or deleted.

1. District
2. Residency
3. County
4. Route Number/Loop Site
5. City/Town/County
6. Date
7. Direction
8. Lane
9. Milepoint
10. Speed
11. Data-Type
12. Skid Number

To delete a record, an input card must be punched with the information listed above in the appropriate columns as indicated in Table 1. It is not necessary to fill all 52 columns, but it is acceptable to do so. The word DELETE must be punched in columns 70-75. If two or more records exist on the file with information equivalent to that on the delete card, only the first record will be deleted.

To alter a record, two cards are necessary. The first card contains the information as described above with the word UPDATE punched starting in column 70. The information to be changes is punched in the appropriate column on the second card along with UPDATE punched starting in column 70. Whole fields of information must be altered at a time; i.e., if changing only the month, the whole dat must be punched (a field of information exists for each item listed in Table 1).

Since several alterations and deletions may be desired in a single run of SKIDUPT, these two types of inputcardsmay be interspersed. In this event they must be arranged in the order that the records to be altered or deleted occur on the file.

Input data to SKIDUPT are edit checked as indicated in Table 1, and in addition the user is informed:

1. that the DELETE or UPDATE commands were omitted,

2. that the record to be deleted or updated cannot be found on the file,

3. that the input data are not in the proper sequence,

4. that insufficient information is contained on one or more input cards, and

5. that an attempt has been made to update more than one file.

The program terminates with information outputs indicating the number of previous records on a file, the number of updated and deleted records, the number of records which now exist on the file, and the total number of errors encountered while processing the input data.

The JCL necessary to execute SKIDUPT is shown in Table 5.

TABLE 5

JCL for SKIDUPT

Card No.	Job Control Language
1	//SKIDUPT JOB (Accounting Information)
2	//GO EXEC PGM=SKIDUPT
3	//STEPLIB DD DSNAME=RCØ.PROGLIB,DISP=SHR
4	//OUTPUT DD SYSOUT=A,DCB=(RECFM=FB,LRECL=120,BLKSIZE=120)
5	//SVFL DD DSN=RCØ.SKIDSURV,UNIT=SYSDA,DISP=(OLD, PASS,KEEP)
6	//BRFL DD DSN=RCØ.SKIDBRIJ,UNIT=SYSDA,DISP=(OLD, PASS, KEEP)
7	//LPFL DD DSN=RCØ.SKIDLOOP,UNIT=SYSDA,DISP=(OLD,PASS, KEEP)
8	//TAPE2 DD DSN=BCKFLE,UNIT=SYSDA,DISP=(NEW,PASS,DELETE),
9	// DCB=(LRECL=80,RECFM=FB,BLKSIZE=1200),SPACE=(TRK,(10,5),RLSE),
10	// VOL=SER=VHDPØØ
11	//TAPE3 DD DSN=NEWMTR,UNIT=SYSDA,DISP=(NEW,PASS,DELETE),
12	// DCB=(LRECL=80,RECFM=FB,BLKSIZE=1200),SPACE=(TRK,(10,5),RLSE),
13	// VOL=SER=VHDPØØ
14	//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
15	//SORTWKØ1 DD UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE)
16	//SORTWKØ2 DD UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE)
17	//SORTWKØ3 DD UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE)
18	//SORTWKØ4 DD UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE)
19	//SORTWKØ5 DD UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE)
20	//SORTWKØ6 DD UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE)
21	//SYSOUT DD SYSOUT=A
22	//CARDS DD *
23 through n	DATA DECK

TABLE 5 (continued)

Card No.	Job Control Language
n+1	/*
n+2	//STEP1 EXEC SORTD,COND=(1,NE,GO)
etc	//SORTINØ1 DD DSN=BCKFLE,UNIT=SYSDA,DISP=(OLD,DELETE)
	//SORTINØ2 DD DSN=NEWMTR,UNIT=SYSDA,DISP=(OLD,DELETE)
	//SORTOUT DD DSN=RCØ.SKIDSURV,UNIT=SYSDA,DISP=(OLD,KEEP)
	//SYSIN DD *
	MERGE FIELDS=(1,5,A,10,3,A,6,4,A,37,2,A,40,4,A,24,2,A,20,4,A), FORMAT=BI
	END
	/*
	//STEP2 EXEC SORTD,COND=(2,NE,GO)
	//SORTINØ1 DD DSN=BCFLE,UNIT=SYSDA,DISP=(OLD,DELETE)
	//SORTINØ2 DD DSN=MEWMTR,UNIT=SYSDA,DISP=(OLD,DELETE)
	//SORTOUT DD DSN=RCØ.SKIDBRIV,UNIT=SYSDA,DISP=(OLD, KEEP)
	//SYSIN DD *
	MERGE FIELDS=(1,5,A,10,3,A,6,4,A,37,2,A,40,4,A,24,2,A,20,4,A), FORMAT=BI
	END
	/*
	//STEP3 EXEC SORTD,COND=(3,NE,GO)
	//SORTINØ1 DD DSN=BCKFLE,UNIT=SYSDA,DISP=(OLD,DELETE)
	//SORTINØ2 DD DSN=NEWMTR,UNIT=SYSDA,DISP=(OLD,DELETE)
	//SORTOUT DD DSN=RCØ.SKIDLOOP,UNIT=SYSDA,DISP=(OLD,KEEP)
	//SYSIN DD *
	MERGE FIELDS=(1,5,A,10,3,A,6,4,A,37,2,A,24,2,A,20,4,A,40,4,A), FORMAT=BI
	END
	/*
n+29	//

1132

Program SKIDLST

A detailed listing from any of the three skid files can be obtained by employing the program SKIDLST. An example of this output is shown in Figure 2. As shown, the listing contains the district, residency, county, route, direction, lane, milepoint, date, test speed, actual trailer skid number obtained, the trailer skid number corrected to a 40 mph test speed, and the predicted car stopping distance skid number.

It should be noted that the detailed listing prints one line per each record on the file and skips to the top of a new page for any change in district, residency, county, route, direction, or lane. Therefore, there will be a considerable amount of output associated with any run involving the whole file or a large section of the file.

The use of the detailed output and other desirable potential skid outputs is discussed fully in the report "Integrated Data Systems — Summary Report." Thus, only the method of use of the program SKIDLST is discussed in this report. As indicated in the above mentioned report, the program SKIDLST ultimately will be used most often to produce detailed skid output for specified road sections by request.

To run SKIDLST the user inputs data as specified in Table 6. Several options are available, as discussed below, pertaining to the mainline survey file. The use of the program with the bridge and loop files will be discussed later.

(1) By punching "ALL" in columns 1-3 and 16-18 and leaving the remaining columns blank, the entire file will be listed. If a starting date is punched in columns 31-35, only data with a test date later than or equal to the starting date will be listed. In either case, columns 21-29 must be left blank.

(2) By punching "ALL" in columns 1-3, a Route Number (preceded by I for Interstate or O for Primary) in columns 16-19, and leaving the remaining columns blank, all data for the route punched will be listed. As for option number 1 above, a starting date can be specified in columns 31-35 if desired. Columns 21-29 must also be left blank for this option.

(3) By punching a county name in columns 1-14 (left justified), "ALL" in columns 16-18, and leaving the remaining columns blank, data for all routes in the specified county will be listed. Again, a starting date may be specified and columns 21-29 must be left blank.

(4) By punching a county name in columns 1-14 (left justified), a route number in columns 16-19, and leaving the remaining columns blank, all data for the specified route in the specified county will be listed. A starting date can be specified as in the options above if desired. In addition, a starting milepoint to the nearest one hundredth mile may be punched in columns 21-24 (a decimal point is implied between columns 22 and 23), or an ending milepoint to the nearest one hundredth mile may be punched in columns 26-29 (a decimal point is implied between columns 27 and 28), or both starting or ending

milepoint may be punched. Thus, for example, one may specify that all skid data on Route 29 in Albemarle County beginning at milepoint 12.87 be listed, or that all data up to milepoint 15.35 be listed, or that all data between 12.87 and 15.35 be listed. Again, a starting date may be specified. Thus, in the last example one could specify that all data collected after January 1, 1975, on Route 29 in Albemarle County between the milepoints 12.87 and 15.35 be listed.

TABLE 6

Input Data for SKIDLST Program

Columns	Data
1-14	County Name or "ALL"[(a)]
16-19	Route Number/Loop Number or "ALL"[(b)]
21-24	Starting Milepoint to the nearest one hundredth mile
26-29	Ending Milepoint to the nearest one hundredth mile
31-35	Month - Year with dash in column 33

(a) Left justified with remaining columns left blank.

(b) "ALL" should be left justified with remaining column left blank.

The program can be utilized to produce listings from the bridge and loop files in the same ways as discussed above. Thus, if someone specifies the beginning and ending milepoints of a particular bridge on a particular route, then only data for that bridge will be listed. With regard to the loop file, it obviously makes no sense to specify milepoints since each loop site designation refers to a specific section of roadway.

The JCL necessary to execute SKIDLST is shown in Table 7. As implied above, only one file can be accessed per run, but as many input cards (requests) as desired may be included in each run. The file to be accessed is indicated in the fifth line of the JCL.

TABLE 7

JCL for SKIDLST

Card No.	Job Control Language
1	//SKIDLST JOB (Accounting Information)
2	//GO EXEC PGM=SKIDLST
3	//STEPLIB DD DSNAME=RCØ.PROGLIB,DISP=SHR
4	//OUTPUT DD SYSOUT=A,DCB=(LRECL=120,RECFM=FB,BLKSIZE=120)
5	//TAPE1 DD DSN=RCØ.SKIDfilename,UNIT=SYSDA,DISP=(OLD,KEEP)
6	//CARDS DD *
7 through n	DATA CARDS
n+1	/*
n+2	//

```
DISTRICT   CULPEPER
RESIDENCY  FAIRFAX
COUNTY     FAIRFAX
ROUTE      I  95

           MILE                  TEST    ACTUAL    CORR
LANE       POST       DATE       SPEED     SN       SN      PCSN

SBL1        .10     10-09-74     42.0      40       41       52
            .30                  40.0      42       42       53
            .50                  40.0      44       44       54
            .70                  40.0      42       42       53
           1.00                  40.0      44       44       54
           1.20                  40.0      43       43       53
           1.40                  40.0      45       45       55
           1.50     10-08-74     40.0      47       47       56
           1.60                  42.0      45       46       56
           1.80                  40.0      46       46       56
           2.00                  38.0      46       45       55
           2.20                  42.0      44       45       55
           2.40                  42.0      44       45       55
           2.80                  41.0      42       43       53
           3.00                  41.0      41       42       52
           3.30                  40.0      43       43       53
           3.60                  40.0      45       45       55
           3.80                  40.0      44       44       54
           4.00                  40.0      43       43       53
           4.20                  40.0      46       46       56
           4.40                  40.0      45       45       55
           4.60                  40.0      45       45       55
           4.80                  40.0      43       43       53
           5.00                  40.0      45       45       55
           5.20                  38.0      46       45       55
           5.40                  40.0      46       46       56
           5.60                  38.0      46       45       55
           5.80                  40.0      41       41       52
           6.00                  41.0      42       43       53
           6.20                  39.0      47       46       56
           6.40                  40.0      46       46       56
           6.60                  40.0      45       45       55
           6.80                  39.0      46       45       55
           7.00                  40.0      44       44       54
           7.20                  40.0      44       44       54
           7.30                  40.0      44       44       54
           7.40                  40.0      41       41       52
           7.60     10-01-74     40.0      48       48       57
           7.80                  40.0      48       48       57
           8.00                  40.0      48       48       57
           8.20                  40.0      50       50       59
           8.40                  40.0      50       50       59
           8.60                  40.0      51       51       59
           8.80                  40.0      50       50       59
           9.00                  40.0      50       50       59
           9.20                  40.0      50       50       59
           9.60                  40.0      45       45       55
           9.80                  40.0      42       42       53
```

Figure 2. Example of detailed skid report.

Program SKIDSUM

The program SKIDSUM may be utilized to produce a listing presenting average skid information by lane mile as illustrated in Figure 3. The data shown for each lane mile are average trailer skid number corrected to a 40 mph test speed, the average test speed, the number of tests the average skid number and the average speed are based on, the standard deviation of the average correlated trailer skid number, and the predicted car stopping distance skid number. The program skips to the top of a new page upon a change in district, residency, county, route, or direction, and currently can handle no more than three lanes in a given direction.

The procedures for utilizing SKIDSUM are exactly the same as discussed above for SKIDLST, including the necessary change in the JCL depending on the file to be accessed. However, it is envisioned that SKIDSUM will be run only against the mainline survey skid file. The JCL necessary to run SKIDSUM is shown in Table 8.

RECOMMENDATION

It is recommended that the skid data system as described in this report be implemented as an operating system to be maintained by the Data Processing Division. Of course, the implementation would be subject to revisions in the system as deemed necessary by the Data Processing Division.

```
DISTRICT  CULPEPER
RESIDENCY FAIRFAX
COUNTY    FAIRFAX
ROUTE     I  95
SUMMARY INFORMATION BEGINNING 10-74

                       SBL-1                               SBL-2                               SBL-3
             -----------------------------       -----------------------------       -----------------------------
             AVG                STD              AVG                STD              AVG                STD
             COR   AVG          DEV              COR   AVG          DEV              COR   AVG          DEV
 MILE        SN    SPEED   N    SN    PCSN       SN    SPEED   N    SN    PCSN       SN    SPEED   N    SN    PCSN

 0- 1        42    40.5    4    1.2   53         43    39.8    4    2.3   54         49    40.0    2    1.4   58

 1- 2        45    40.3    6    1.5   55         44    40.2    6    1.8   54         50    40.0    5    2.0   59

 2- 3        44    40.8    4    1.3   54         44    39.8    4    1.5   54         49    40.0    4    1.0   58

 3- 4        43    40.3    4    1.5   54         45    40.0    4    1.0   55         49    40.0    4    1.0   58

 4- 5        44    40.0    5    1.3   54         44    39.4    5     .7   54         50    40.0    5     .4   59

 5- 6        44    39.2    5    1.9   54         44    39.8    5    1.0   54         49    39.8    5     .8   58

 6- 7        45    39.8    5    1.5   55         45    39.4    5     .9   55         48    40.2    5    1.2   57

 7- 8        45    40.0    6    2.7   55         48    40.0    6    4.0   57         49    39.7    6    1.3   58

 8- 9        50    40.0    5    1.1   59         52    40.0    5     .8   60         50    40.0    1          59

 9-10        47    40.0    4    3.9   56         53    40.0    4    2.5   61         50    40.0    1          59

10-11        42    40.2    5    3.6   52         42    40.0    5    3.3   53         48    40.0    6    2.0   57

11-12        45    39.6    5    1.5   55         45    40.0    5    3.0   55         44    40.0    4    6.9   54

12-13        45    40.0    5    1.3   55         49    40.0    5    1.3   58         42    40.0    5    5.0   53

13-14        43    40.0    5    1.7   53         47    40.0    5    1.0   56         44    40.0    6    2.7   54

14-15        44    40.0    4    1.0   54         44    40.0    4    1.0   54         43    40.0    2    1.4   53
```

Figure 3. Example of summary skid output.

TABLE 8

JCL for SKIDSUM

Card No.	Job Control Language
1	//SKIDSUM JOB (Accounting Information)
2	//GO EXEC PGM=SKIDSUM
3	//STEPLIB DD DSNAME=RCØ.PROGLIB,DISP=SHR
4	//OUTPUT DD SYSOUT=A,DCB=(RECFM=FB,LRECL=120,BLKSIZE=120)
5	//TAPE1 DD DSN=RCØ.SKIDfilename,UNIT=SYSDA,DISP=(OLD,KEEP)
6	//TAPE2 DD DSN=&&TEMP,UNIT=SYSDA,DISP=(NEW,DELETE),
7	// DCB=(RECFM=FB,LRECL=80,BLKSIZE=1200),SPACE=(TRK,(10,5)),
8	// VOL=SER=VHDPØØ
9	//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
10	//SORTWKØ1 DD UNIT=SYSDA,SPACE=(TRK,(10,5),RLSE)
11	//SORTWKØ2 DD UNIT=SYSDA,SPACE=(TRK,(10,5),RLSE)
12	//SORTWKØ3 DD UNIT=SYSDA,SPACE=(TRK,(10,5),RLSE)
13	//SORTWKØ4 DD UNIT=SYSDA,SPACE=(TRK,(10,5),RLSE)
14	//SORTWKØ5 DD UNIT=SYSDA,SPACE=(TRK,(10,5),RLSE)
15	//SORTWKØ6 DD UNIT=SYSDA,SPACE=(TRK,(10,5),RLSE)
16	//SORTWKØ7 DD UNIT=SYSDA,SPACE=(TRK,(10,5),RLSE)
17	//SORTWKØ8 DD UNIT=SYSDA,SPACE=(TRK,(10,5),RLSE)
18	//SORTWKØ9 DD UNIT=SYSDA,SPACE=(TRK,(10,5),RLSE)
19	//SYSOUT DD SYSOUT=A
20	//CARDS DD *
21 through n	DATA CARDS
n+1	/*
n+2	//

REFERENCES

1. Runkle, S. N., "Integrated Data Systems — Summary Report", Virginia Highway & Transportation Research Council, VHTRC 75-R62, May 1975.

2. Runkle, S. N., "Test Procedures and Data Input Techniques for Skid Testing", Virginia Highway and Transportation Research Council, April 1974.

2109