eport No. FHWA-RD-75-28

VALUATION OF FIRST GENERATION UTCS/BPS ONTROL STRATEGY

Vol. II. Technical Appendices

J. L. Kay, J. C. Allen, and J. M. Bruggeman

February 1975
Final Report

This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

Prepared for
FEDERAL HIGHWAY ADMINISTRATION
Offices of Research & Development
Washington, D.C. 20590

NOTICE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA-RD-75-28	75-28	

4. Title and Subtitle Evaluation of the First Generation UTCS/BPS Control Strategy ~~Vol. 2 - Technical Appendices~~	5. Report Date February, 1975
	6. Performing Organization Code
7. Author(s) J.L. Kay, J.C. Allen, J.M. Bruggeman	8. Performing Organization Report No.
9. Performing Organization Name and Address JHK & Associates & Peat, Marwick, Mitchell & Co. Suite 1112, 4660 Kenmore Avenue, Alexandria, VA 22304 1025 Connecticut Ave., NW, Washington, D.C. 20036	10. Work Unit No. (TRAIS) FCP 32B4254
	11. Contract or Grant No. DOT-FH-11-8242
12. Sponsoring Agency Name and Address Traffic Systems Division, Office of Research Federal Highway Administration, Department of Transportation Washington, D. C. 20590	13. Type of Report and Period Covered Final Report February 1974 - February 1975
	14. Sponsoring Agency Code T-0119

15. Supplementary Notes

FHWA Contract Manager: H. Bissell, HRS-33

16. Abstract

This report presents the results of the evaluation of the alternatives of the first-generation traffic control strategy for the computerized Urban Traffic Control System and Bus Priority System in Washington, D. C. Five traffic control alternatives were evaluated to assess their relative impact on traffic during typical weekday morning, midday, and evening periods. Significant differences between alternatives were identified through measurements from the UTCS/BPS surveillance system and moving-car studies.

This volume is the second of three reports and contains the technical appendices including the detailed discussion of field studies, sample forms, and the documentation and user guides for the computer programs developed for processing and evaluating detector and moving-car data. The technical report includes a discussion of the overall project and the conclusions. The other volume is:

Vol. No.	FHWA No.	Short Title	NTIS (PB) No.
1	75-	Technical Report	(not yet available)
-	75-	Executive Summary	(not yet available)

The executive summary includes a project overview and a discussion of the conclusions.

17. Key Words	18. Distribution Statement
Traffic Analysis, Computers, Buses, Traffic Control, Signal Control, Traffic Congestion, Vehicle Detectors, Traffic Flow, Time Lapse Photography, Traffic Signals, Fixed Time Traffic Signals	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	188	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

VOLUME 2 TABLE OF CONTENTS

APPENDIX A

DOCUMENTATION OF DETECTOR

DATA ANALYSIS PROGRAMS

INTRODUCTION

This program documentation was developed incrementally during the course of the UTCS/BPS evaluation project. As a consequence, certain of the illustrative job control language (JCL) setups are no longer consistent with the revised programs. In other cases, the parameters shown in the sample setups were chosen to facilitate the original development of the programs on a particular computer operating system. Figures 28 and 29 in the OPERATIONAL PROCEDURES Chapter illustrate the basic JCL statements needed to execute the sequence of current programs.

LISTING OF DOCUMENTATION CONVENTIONS

1. The term "observation" will apply to a 15-minute period, such as 7:15 to 7:30, for which 15-minute measure of effectiveness summaries are prepared.

2. The term "time of day" will apply to one of three periods during a typical weekday: a.m. (7:00-9:45), midday (1:00-3:00), and p.m. (4:00-6:30).

3. The term "traffic control pattern" will apply to the prevailing system of traffic light control being used when a particular set of detector readings were made. A particular set of "adjacent" weekdays will generally be associated with the operation of a given traffic control pattern being tested.

4. The term "individual data elements" will apply to a single set of seven MOE's for a specific link during a specific observation on a certain day.

5. The term "data quality status" relates to the level of usability assigned to a particular individual data element on the basis of several criteria.

6. The term "case" shall apply to a specific control pattern/time of day combination and the data files pertaining to it.

7. Whenever "INTEGER*2" appears, the associated and following variables are integer half words. It thus must be kept in mind that the values taken on by these variables cannot exceed the maximum integer half-word size of the machine being used.

DATA FILE SUFFIX CONVENTIONS

1. MMMDD is a general data file name suffix designation which means that the file corresponds to the day specified by month MMM (three letter abbreviation) and date DD (two digit integer).

2. XXY is a general data file suffix designation which means that the file corresponds to the time of day XX (AM, MD for MIDDAY or PM) and traffic control pattern number Y (one digit integer).

3. XXYZ is similar to the suffix XXY, but it occurs when the file involves a comparison of control patterns Y and Z. If XXZY also can occur, the file applies to the traffic control pattern of the first digit while the second digit is the pattern to which it is being compared for purposes of defining the file.

4. In the above cases, and in all other cases where it is desired to indicate a general (variable) suffix, the variable suffix will be underlined:

   i.e., MOE2<u>XXY</u> to indicate the general form of a file type that in a specific case might be MOE2AM1

5. Numbers or characters appearing in double braces represent the current values being used for a parameter (such as TEST(1) [[8]]).

6. All numbers are right justified in their fields and all character strings are left justified in their fields unless otherwise stated.

OVERALL COMPUTER PROGRAM LINKAGE DESCRIPTION

The starting point is the set of daily traffic detector summary files on disk. Each such file contains data on the seven measures of effectiveness for each of 270 detectorized links for each of approximately 30 fifteen-minute periods throughout the day (on occasion, a file will contain data for a fraction of a day instead of the normal single full day). These disk files are the raw input to the chain of computer programs to be described below.

In what follows, the chain of data manipulations typically performed will be briefly presented. This will provide a context for the more detailed documentation which follows it. The flow of data manipulations will be expressed with reference to sample JCL decks and the associated FORTRAN programs being executed.

STEP 1

The first step is to delete any observations from the raw daily detector summary files which are known to be defective or which are suspected of being unusable. This is accomplished through the use of JCL such as appears in Figure 1. For any daily detector summary file having one or more 15-minute observations that are to be deleted, a new daily file must be created that is identical to the original file, except for the deletion of the unwanted observations. The FORTRAN program OBSOUT (see Figure 2) is executed once for each daily file for which deletions are desired and a new permanent modified disk file is created. This modified file will be used in place of the original file in all subsequent work.

STEP 2

The next step is to consolidate the daily files into three "time of day" files for each traffic control pattern being examined. In addition, the process of screening the data for unusable observations to be deleted and to determine data quality status of individual data elements is initiated. The JCL illustrated in Figure 3 is an example of the type of setup that can be used in executing the program CONFIL3 (see Figure 4), which comprises this step for a single traffic control pattern.

For each traffic control pattern, CONFIL3 is executed once to produce three consolidated time-of-day files from the 10 to 12 daily files that typically comprise a specific control pattern period. Aside from the segregation of

```
//OBSOUT EXEC PGM=OBSOUT
//STEPLIB DD DSN=PMMCO.UTCS.LOAD1,DISP=SHR,
// UNIT=SYSDA,VOL=SER=OTHERS
//FT06F001 DD SYSOUT=A
//FT08F001 DD DSN=PMMCO.UTCS.DET.APR04,DISP=SHR,
// UNIT=SYSDA,VOL=SER=OTHERS
//FT09F001 DD *
00000000001000000000000000000000000000000000000000000000000000000000000 00000000
//FT10F001 DD DSN=PMMCO.UTCS.DET.MOD.APR04,
// UNIT=SYSDA,DISP=(NEW,KEEP),
// DCB=(RECFM=VBS,LRECL=4340,BLKSIZE=7294),
// SPACE=(7294,(45,5),RLSE)
```

Figure 1. Sample listing for job control language.

```
                              LISTING OF JHKA.JHKB.PROGLIB
MEMBER NAME OBSOUT

C       PROGRAM OBSOUT
C
C       PROGRAM TO MANUALLY DELETE OBSERVATIONS FROM A DAILY FILE
C
      INTEGER*2 TAB(2160)
      INTEGER EDIT(50),DATE(3),TIME
      KNT=0
      READ(9,100)(EDIT(I),I=1,50)
  100 FORMAT(50I1)
   10 READ(8,END=30)DATE,TIME,TAB
      KNT=KNT+1
      IF(KNT.LE.50) GO TO 20
      WRITE(6,200)
  200 FORMAT(' MORE THAN 50 RECORDS IN DATA SET')
      GO TO 30
   20 IF(EDIT(KNT).EQ.0) GO TO 40
      WRITE(6,300)DATE,TIME
  300 FORMAT(' OBSERVATION',3A4,2X,A4,'  DELETED')
      GO TO 10
   40 WRITE(10)DATE,TIME,TAB
      GO TO 10
   30 CONTINUE
      RETURN
      END
```

Figure 2. Sample listing for program OBSOUT.

```
//CONFIL3 EXEC PGM=CONFIL3
//STEPLIB DD DSN=PMMCO.UTCS.LOAD1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT09F001 DD *
13312132133111111113211332321223232222222222222222111111111111331333331132333 2133
13312112321100313321313321133323131111111113303131110131331111111133131313311 3333
31331113121133203111103233313330132133313121113233131333331111131311111111133123
211113322213311111111111111111
//FT10F001 DD *
         8        50      6000      1500          5         75
//FT11F001 DD DSN=PMMCO.UTCS.DET.APR15,VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA
// DD DSN=PMMCO.UTCS.DET.APR16,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
// DD DSN=PMMCO.UTCS.DET.APR17,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
// DD DSN=PMMCO.UTCS.DET.APR18,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
// DD DSN=PMMCO.UTCS.DET.APR19,VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA
// DD DSN=PMMCO.UTCS.DET.APR22,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
// DD DSN=PMMCO.UTCS.DET.APR23,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
// DD DSN=PMMCO.UTCS.DET.MOD.APR26,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
// DD DSN=PMMCO.UTCS.DET.APR30A,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
// DD DSN=PMMCO.UTCS.DET.APR30B,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
// DD DSN=PMMCO.UTCS.DET.MAY01A,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
// DD DSN=PMMCO.UTCS.DET.MAY01B,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT12F001 DD DSN=PMMCO.UTCS.TEMPAM3,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=4892,BLKSIZE=7294),
// SPACE=(7294,(100,9),RLSE)
//FT13F001 DD DSN=PMMCO.UTCS.TEMPMD3,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=*.FT12F001,
// SPACE=(7294,(70,7),RLSE)
//FT14F001 DD DSN=PMMCO.UTCS.TEMPPM3,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=*.FT12F001,
// SPACE=(7294,(90,8),RLSE)
```

Figure 3. Sample listing of JCL for program CONFIL3.

```
C       PROGRAM CONFIL3
C
C        A PROGRAM TO CREATE THREE TIME OF DAY FILES FROM A SET OF RAW
C        FILES FOR A GIVEN TRAFFIC CONTROL SYSTEM
C
      INTEGER*2 MOE(8,270),INDOUS(270)
      INTEGER*2 NDET(270)
      INTEGER DATE(3),TIME,TIMKEY(2),INDPER,DATKEY(12),TEST(6)
      DATA INDOUS/270*0/
      DATA TIMKEY/'0945','1600'/,DATKEY/' MAR',' APR',' MAY',' JUN',' JU
     1L',' AUG',' SEP',' OCT',' NOV',' DEC',' JAN',' FEB'/
      READ(9,10) (NDET(J),J=1,270)
   10 FORMAT(80I1)
      READ(10,20) (TEST(K),K=1,6)
   20 FORMAT(6I10)
   25 READ(11,END=1000) DATE,TIME,((MOE(I,J),I=1,8),J=1,270)
      NBAD = 0
      NFLAG = 0
      NSDET = 0
      DO 30 I=1,12
      IF(DATE(1).EQ.DATKEY(I)) GO TO 50
   30 CONTINUE
      WRITE(6,40) (DATE(I),I=1,3),TIME,(MOE(I1,1),I1=1,8)
   40 FORMAT(' ','THE START OF A 15 MINUTE FILE IS INCORRECT OR MISPLACE
     1D: ',3A4,5X,A4,/,' ',8I10,//)
      STOP
   50 DO 55 J10=1,270
   55 INDOUS(J10) = 0
      INDPER = 2
      IF(TIME.LE.TIMKEY(1)) INDPER=1
      IF(TIME.GE.TIMKEY(2)) INDPER=3
      IUN = 11 + INDPER
      DO 500 J=1,270
      IF(NDET(J).LE.1) GO TO 57
      IF((MOE(4,J).EQ.0).AND.(MOE(5,J).EQ.0).AND.(MOE(8,J).EQ.0)) GO TO
     157
      GO TO 58
   57 INDOUS(J) = 2
      NSDET = NSDET + 1
   58 IF(MOE(1,J).NE.J) GO TO 300
      IF(MOE(2,J).LT.TEST(1)) GO TO 350
      IF(MOE(6,J).GT.TEST(2)) GO TO 350
      IF(MOE(7,J).GT.TEST(3)) GO TO 350
      DO 60 I=3,8
      IF(MOE(I,J).LT.0) GO TO 350
   60 CONTINUE
      IF(MOE(2,J).GT.TEST(4)) GO TO 350
      GO TO 500
  300 IF(MOE(1,J).GE.0) GO TO 340
      MOE(1,J) = J
      INDOUS(J) = INDOUS(J) + 5
      NFLAG = NFLAG + 1
      IF(INDOUS(J).EQ.7) NSDET = NSDET - 1
      GO TO 500
  340 MOE(1,J) = J
```

Figure 4. Sample listing for program CONFIL3.

```
  350 INDOUS(J) = INDOUS(J) + 4
      NBAD = NBAD + 1
      IF(INDOUS(J).EQ.6) NSDET = NSDET - 1
  500 CONTINUE
      IF((NBAD.LT.TEST(5)).AND.((NBAD+NFLAG).LT.TEST(6))) GO TO 600
      WRITE(6,510) (DATE(K7),K7=1,3),TIME,NBAD,NFLAG,NSDET
  510 FORMAT(' ','THE 15 MINUTE INTERVAL ',3A4,5X,A4,'   WAS DELETED BEC
     1AUSE IT HAD  NBAD = ',I3,';  NFLAG = ',I3,';  NSDET = ',I3,//)
      GO TO 25
  600 WRITE(IUN) DATE,TIME,NBAD,NFLAG,NSDET,((INDOUS(J),(MOE(I,J),I=1,8)
     1),J=1,270)
      GO TO 25
 1000 STOP
      END
```

Figure 4. Sample listing for program CONFIL3 (continued).

observations into different files by time of day and the consolation of observations for all days of the traffic control pattern period into a single file, the resulting files are similar to the original daily files. However, based on several tests made on the data, one or more observations may have been deleted as unusable and index values which specific data quality status are incorporated into the new files created.

STEP 3

The sample string of JCL appearing in Figure 5 executes in a single job a number of FORTRAN and utility programs for a specific control pattern and time of day so as to make more efficient the manipulation of the associated temporary and intermediate data files. Although a number of different programs are involved, they all operate on data for a given control pattern and time of day in a prespecified sequential fashion and so may be treated as a single multipart step or chain.

The first program to be executed is OLOUT (see Figure 6). The primary input to this program and the chain is one of the time of day files produced by a run of the program CONFIL3. Thus, this chain is run three times for every one execution of CONFIL3.

The program OLOUT makes a final pass to determine if any other observations should be deleted, based on another set of tests. A new temporary data set is created, similar to the input data set, but missing any observations deleted. In addition, a vector is computed and stored, which specifies usability of each of the 270 links for that control pattern and time of day, based on the number of detectors functioning for that link and the data quality levels achieved by the various observations on that link.

Following this program, a sort is executed using an IBM SORT/MERGE routine. The temporary data file created by the last program is the basic input and the output is another temporary file, identical to the first, except that the observations have been reordered so that they are in ascending temporal order by clock time, and observations with the same clock time are adjacent. This shuffling of the data file will facilitate several subsequent computations.

The program PLUGNET (see Figure 7) performs three basic functions. First, it calculates the mean values of each of the seven performance MOE's for each link over all usable observations (a) for each 15-minute period separately and

```
//OLOUT EXEC PGM=OLOUT
//STEPLIB DD DSN=PMMCO.UTCS.LOAD1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT09F001 DD *
       20          5        0.9        0.6        0.9
//FT10F001 DD DSN=PMMCO.UTCS.TEMPAM1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT11F001 DD DSN=&&TEMP1,
// UNIT=SYSDA,DISP=(NEW,PASS),
// DCB=(RECFM=VBS,LRECL=4892,BLKSIZE=7294),
// SPACE=(7294,(100,9),RLSE)
//FT12F001 DD DSN=PMMCO.UTCS.NOBS(AM1),LABEL=(,,,OUT),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP),
// DCB=(RECFM=FB,LRECL=72,BLKSIZE=72),
// SPACE=(72,(1,1,10))
//FT13F001 DD DSN=PMMCO.UTCS.INDGAM1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP),
// DCB=(RECFM=VBS,LRECL=3244,BLKSIZE=3248),
// SPACE=(3248,(1,1))
//FT14F001 DD DSN=PMMCO.UTCS.LUSEAM1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP),
// DCB=(RECFM=VBS,LRECL=544,BLKSIZE=548),
// SPACE=(548,(1,1))
//SORT1 EXEC PGM=SORT,COND=(0,LT),PARM='MSG=AP,CORE=100000'
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SYSIN DD *
 SORT FIELDS=(17,4,CH,A),SIZE=E85
//SORTIN DD DSN=&&TEMP1,DISP=(OLD,DELETE),UNIT=SYSDA
//SORTOUT DD DSN=&&TEMP2,
// UNIT=SYSDA,DISP=(NEW,PASS),
// DCB=(RECFM=VBS,LRECL=4892,BLKSIZE=7294),
// SPACE=(7294,(100,9),RLSE)
//PLUG EXEC PGM=PLUGNET,COND=(0,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT09F001 DD DSN=PMMCO.UTCS.LUSEAM1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT11F001 DD DSN=PMMCO.UTCS.NOBS(AM1),LABEL=(,,,IN),VOL=SER=OTHERS,
// DISP=SHR,UNIT=SYSDA
//FT12F001 DD *
 CONTROL PATTERN 2; P.M.
//FT13F001 DD DSN=&&TEMP2,DISP=(OLD,DELETE),UNIT=SYSDA
//FT14F001 DD DSN=PMMCO.UTCS.IBARAM1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=7564,BLKSIZE=7294),
// SPACE=(7294,(40,2),RLSE)
//FT15F001 DD DSN=PMMCO.UTCS.GBARAM1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=7564,BLKSIZE=7294),
// SPACE=(7294,(2,1),RLSE)
//FT16F001 DD DSN=&&TEMP3,
// UNIT=SYSDA,DISP=(NEW,PASS),
```

Figure 5. Sample JCL for program OLOUT.

```
// DCB=(RECFM=VBS,LRECL=4904,BLKSIZE=7294),
// SPACE=(7294,(100,9),RLSE)
//SCTRGM EXEC PGM=SCTRGM,COND=(0,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT09F001 DD *
 CONTROL PATTERN 2; P.M.
//FT10F001 DD DSN=PMMCO.UTCS.NOBS(AM1),LABEL=(,,,IN),VOL=SER=OTHERS,
// DISP=SHR,UNIT=SYSDA
//FT11F001 DD DSN=PMMCO.UTCS.LUSEAM1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT12F001 DD DSN=&&TEMP3,DISP=(OLD,PASS),UNIT=SYSDA
//SORT2 EXEC PGM=SORT,COND=(0,LT),PARM='MSG=AP,CORE=100000'
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SYSIN DD *
 SORT FIELDS=(17,4,CH,A,33,4,BI,A),SIZE=E85
//SORTIN DD DSN=&&TEMP3,DISP=(OLD,DELETE),UNIT=SYSDA
//SORTOUT DD DSN=&&TEMP4,
// UNIT=SYSDA,DISP=(NEW,PASS),
// DCB=(RECFM=VBS,LRECL=4904,BLKSIZE=7294),
// SPACE=(7294,(100,9),RLSE)
//BRKFIL EXEC PGM=BRKFIL,COND=(0,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT06F001 DD SYSOUT=A,DCB=BUFNO=1
//FT09F001 DD DSN=PMMCO.UTCS.NOBS(AM1),LABEL=(,,,IN),VOL=SER=OTHERS,
// DCB=BUFNO=1,DISP=SHR,UNIT=SYSDA
//FT10F001 DD DSN=&&TEMP4,DISP=SHR,UNIT=SYSDA,DCB=BUFNO=1
//FT11F001 DD DSN=PMMCO.UTCS.HEADAM1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=48,BLKSIZE=2498,BUFNO=1),
// SPACE=(2498,(2,1),RLSE)
//FT12F001 DD DSN=PMMCO.UTCS.INDSAM1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
//FT13F001 DD DSN=PMMCO.UTCS.MOE2AM1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
//FT14F001 DD DSN=PMMCO.UTCS.MOE3AM1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
//FT15F001 DD DSN=PMMCO.UTCS.MOE4AM1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
//FT16F001 DD DSN=PMMCO.UTCS.MOE5AM1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
```

Figure 5. Sample JCL for program OLOUT (continued).

```
// SPACE=(7294,(9,1),RLSE)
//FT17F001 DD DSN=PMMCO.UTCS.MOE6AM1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
//FT18F001 DD DSN=PMMCO.UTCS.MOE7AM1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
//FT19F001 DD DSN=PMMCO.UTCS.MOE8AM1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
//ENDPRNT EXEC PGM=ENDPRNT,COND=(0,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT09F001 DD DSN=PMMCO.UTCS.NOBS(AM1),LABEL=(,,,IN),VOL=SER=OTHERS,
// DISP=SHR,UNIT=SYSDA
//FT10F001 DD DSN=PMMCO.UTCS.LUSEAM1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT11F001 DD DSN=PMMCO.UTCS.HEADAM1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT12F001 DD DSN=PMMCO.UTCS.MOE2AM1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT13F001 DD DSN=PMMCO.UTCS.MOE8AM1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
```

Figure 5. Sample JCL for program OLOUT (continued).

```
MEMBER NAME OLOUT

C       PROGRAM OLOUT
C
C       PROGRAM TO REMOVE ANY REMAINING BAD OBSERVATIONS AND BUILD
C       A LINK USAGE VECTOR, LUSE(J)
C
      INTEGER*2 MOE(8,270),INDOUS(270),LUSE(270),INDG(6,270)
      INTEGER DATE(3),TIME,TEST2(2)
      REAL FRTST(3)
      DATA LUSE/270*1/,INDG/1620*0/,NOBS/0/
      READ(9,10) (TEST2(I),I=1,2),(FRTST(I),I=1,3)
   10 FORMAT(2I10,3F10.5)
   20 READ(10,END=1000) DATE,TIME,NBAD,NFLAG,NSDET,((INDOUS(J),(MOE(I,J)
     1,I=1,8)),J=1,270)
      DO 800 J=1,270
      INDOUS(J) = INDOUS(J) + 1
      I999= INDOUS(J)
      GO TO(100,700,200,700,300,400,500,600,700),I999
  100 INDG(1,J)= INDG(1,J) + 1
      GO TO 800
  200 INDG(2,J)= INDG(2,J) + 1
      GO TO 800
  300 INDG(3,J)= INDG(3,J) + 1
      GO TO 800
  400 INDG(4,J)= INDG(4,J) + 1
      GO TO 800
  500 INDG(5,J)= INDG(5,J) + 1
      GO TO 800
  600 INDG(6,J)= INDG(6,J) + 1
      GO TO 800
  700 INDOUS(J)= INDOUS(J) - 1
      WRITE(6,750) J,INDOUS(J),(DATE(I1),I1=1,3),TIME
  750 FORMAT(' ','INDOUS(',I3,') = ',I2,'  FOR ',3A4,5X,A4,' IS INVALID.
     1  JOB TERMINATED',/)
      STOP
  800 CONTINUE
      GO TO 20
 1000 CONTINUE
      REWIND 10
 1100 READ(10,END=2000) DATE,TIME,NBAD,NFLAG,NSDET,((INDOUS(J),(MOE(I,J)
     1,I=1,8)),J=1,270)
      NSMLT = 0
      DO 1150 J=1,270
      IF((INDG(1,J)+INDG(2,J)).LT.1) GO TO 1150
      IF((((INDG(1,J)+0.)/(INDG(1,J)+INDG(2,J)+0.)).GT.FRTST(1)).AND.(IN
     1DOUS(J).EQ.2)) NSMLT = NSMLT + 1
 1150 CONTINUE
      IF(NSMLT.LT.TEST2(1)) GO TO 1800
      DO 1500 J=1,270
      INDOUS(J) = INDOUS(J) + 1
      I999= INDOUS(J)
      GO TO(1200,1450,1250,1450,1300,1350,1370,1400,1450), I999
 1200 INDG(1,J)= INDG(1,J) - 1
      GO TO 1500
 1250 INDG(2,J)= INDG(2,J) - 1
      GO TO 1500
```

Figure 6. Sample listing for program OLOUT.

```
 1300 INDG(3,J)= INDG(3,J) - 1
      GO TO 1500
 1350 INDG(4,J)= INDG(4,J) - 1
      GO TO 1500
 1370 INDG(5,J)= INDG(5,J) - 1
      GO TO 1500
 1400 INDG(6,J)= INDG(6,J) - 1
      GO TO 1500
 1450 INDOUS(J)= INDOUS(J) - 1
      WRITE(6,1460) J,INDOUS(J),(DATE(I1),I1=1,3),TIME
 1460 FORMAT(' ','INDOUS(',I3,') = ',I2,'  FOR ',3A4,5X,A4,' IS INVALID.
     1  JOB TERMINATED ON SECOND PASS',/)
      STOP
 1500 CONTINUE
      WRITE(6,1550) (DATE(I1),I1=1,3),TIME,NSMLT
 1550 FORMAT(' ','THE 15 MINUTE INTERVAL ',3A4,5X,A4,'   WAS DELETED BEC
     1AUSE IT HAD NSMLT = ',I3,//)
      GO TO 1900
 1800 NOBS=NOBS+1
      WRITE(11) DATE,TIME,NBAD,NFLAG,NSDET,((INDOUS(J),(MOE(I,J),I=1,8))
     1,J=1,270)
 1900 GO TO 1100
 2000 CONTINUE
      WRITE(12,2050) NOBS
 2050 FORMAT(I10)
      WRITE(13) ((INDG(I,J),I=1,6),J=1,270)
      DO 3000 J=1,270
      IF((INDG(1,J)+INDG(2,J)+0.).LT.(NOBS*FRTST(2))) GO TO 2300
      IF((INDG(3,J)+INDG(5,J)).GT.TEST2(2)) GO TO 2300
      IF(((INDG(1,J)+0.)/(INDG(1,J)+INDG(2,J)+0.)).LT.FRTST(3)) GO TO
     1 2400
      GO TO 3000
 2300 LUSE(J) = 0
      GO TO 3000
 2400 LUSE(J) = 2
 3000 CONTINUE
      WRITE(14) (LUSE(J1),J1=1,270)
      STOP
      END
```

Figure 6. Sample listing for program OLOUT (continued).

```
C      PROGRAM PLUGNET
C
C      PROGRAM TO PLUG BAD VALUES WITH MEAN VALUES AND CALCULATE
C      NETWORK QUANTITY VALUES FOR EACH OBSERVATION
C
      INTEGER*2 MOE(8,270),INDOUS(270),LUSE(270)
      INTEGER DATE(3),TIME,NUS(270),NGUS(270),NTUS(270),NTGUS(270)
      INTEGER TIMLST,TITLE(18)
      REAL IMEAN(7,270),GMEAN(7,270)
      DATA IMEAN/1890*0./,GMEAN/1890*0./
      DATA NUS/270*0/,NTUS/270*0/,NGUS/270*0/,NTGUS/270*0/
      DATA TIMLST/'1ST '/,I1ST/'1ST '/
      READ(9) (LUSE(J),J=1,270)
      READ(11,5) NOBS
    5 FORMAT(I10)
      READ(12,7) (TITLE(I),I=1,18)
    7 FORMAT(18A4)
      DO 1000 K7=1,NOBS
      READ(13) DATE,TIME,NBAD,NFLAG,NSDET,((INDOUS(J),(MOE(I,J),I=1,8)),
     1J=1,270)
      IF((TIME.EQ.TIMLST).OR.(TIMLST.EQ.I1ST)) GO TO 50
   12 DO 30 J=1,270
      NGUS(J) = NGUS(J) + NUS(J)
      NTGUS(J) = NTGUS(J) + NTUS(J)
      DO 15 I8=1,7
   15 GMEAN(I8,J) = GMEAN(I8,J) + IMEAN(I8,J)
      IF(NUS(J).EQ.0) GO TO 20
      IMEAN(1,J) = IMEAN(1,J)/NUS(J)
      IMEAN(2,J) = IMEAN(2,J)/NUS(J)
      IMEAN(5,J) = IMEAN(5,J)/NUS(J)
      IMEAN(6,J) = IMEAN(6,J)/NUS(J)
      IF(NTUS(J).EQ.0) GO TO 25
      IMEAN(3,J) = IMEAN(3,J)/NTUS(J)
      IMEAN(4,J) = IMEAN(4,J)/NTUS(J)
      IMEAN(7,J) = IMEAN(7,J)/NTUS(J)
      GO TO 30
   20 IMEAN(1,J) = -1.
      IMEAN(2,J) = -1.
      IMEAN(5,J) = -1.
      IMEAN(6,J) = -1.
   25 IMEAN(3,J) = -1.
      IMEAN(4,J) = -1.
      IMEAN(7,J) = -1.
   30 CONTINUE
      WRITE(14) ((IMEAN(L1,L2),L1=1,7),L2=1,270)
      DO 35 J7=1,270
      NUS(J7) = 0
      NTUS(J7) = 0
      DO 35 I7=1,7
   35 IMEAN(I7,J7) = 0.
      IF((K7.EQ.NOBS).AND.(TIME.EQ.TIMLST)) GO TO 1010
   50 TIMLST = TIME
      DO 200 J=1,270
      IF(INDOUS(J).NE.0) GO TO 150
      NTUS(J) = NTUS(J) + 1
      IMEAN(3,J) = IMEAN(3,J) + MOE(4,J)
```

Figure 7. Sample listing for program PLUGNET.

```
      IMEAN(4,J) = IMEAN(4,J) + MOE(5,J)
      IMEAN(7,J) = IMEAN(7,J) + MOE(8,J)
  150 IF(INDOUS(J).GT.2) GO TO 200
      NUS(J) = NUS(J) + 1
      IMEAN(1,J) = IMEAN(1,J) + MOE(2,J)
      IMEAN(2,J) = IMEAN(2,J) + MOE(3,J)
      IMEAN(5,J) = IMEAN(5,J) + MOE(6,J)
      IMEAN(6,J) = IMEAN(6,J) + MOE(7,J)
  200 CONTINUE
      IF(K7.EQ.NOBS) GO TO 12
 1000 CONTINUE
 1010 DO 1100 J=1,270
      IF(NGUS(J).EQ.0) GO TO 1050
      GMEAN(1,J) = GMEAN(1,J)/NGUS(J)
      GMEAN(2,J) = GMEAN(2,J)/NGUS(J)
      GMEAN(5,J) = GMEAN(5,J)/NGUS(J)
      GMEAN(6,J) = GMEAN(6,J)/NGUS(J)
      IF(NTGUS(J).EQ.0) GO TO 1060
      GMEAN(3,J) = GMEAN(3,J)/NTGUS(J)
      GMEAN(4,J) = GMEAN(4,J)/NTGUS(J)
      GMEAN(7,J) = GMEAN(7,J)/NTGUS(J)
      GO TO 1100
 1050 GMEAN(1,J) = -1.
      GMEAN(2,J) = -1.
      GMEAN(5,J) = -1.
      GMEAN(6,J) = -1.
 1060 GMEAN(3,J) = -1.
      GMEAN(4,J) = -1.
      GMEAN(7,J) = -1.
 1100 CONTINUE
      WRITE(15) ((GMEAN(L1,L2),L1=1,7),L2=1,270)
      REWIND 13
      REWIND 14
      TIMLST = I1ST
      READ(14) ((IMEAN(L1,L2),L1=1,7),L2=1,270)
      DO 3000 K7=1,NOBS
      READ(13) DATE,TIME,NBAD,NFLAG,NSDET,((INDOUS(J),(MOE(I,J),I=1,8)),
     1J=1,270)
      NETVOL= 0
      NETDEL= 0
      NETQUE= 0
      IF((TIME.NE.TIMLST).AND.(TIMLST.NE.I1ST)) READ(14) ((IMEAN(L1,L2),
     1L1=1,7),L2=1,270)
      TIMLST = TIME
      DO 2500 J=1,270
      IF(LUSE(J).EQ.0) GO TO 2500
      IF((INDOUS(J).EQ.2).AND.(LUSE(J).EQ.1)) GO TO 2000
      IF( INDOUS(J).LE.2) GO TO 2400
 2000 IF(IMEAN(1,J).EQ.-1.) GO TO 2200
      DO 2100 M=1,7
      IFULL = MAX1(0.,IMEAN(M,J))
 2100 MOE(M+1,J) = IFULL
      GO TO 2400
 2200 DO 2300 M=1,7
      IFULL = MAX1(0.,GMEAN(M,J))
 2300 MOE(M+1,J) = IFULL
```

Figure 7. Sample listing for program PLUGNET (continued).

```
2400 NETVOL = NETVOL + MOE(2,J)
     IF(LUSE(J).EQ.2) GO TO 2500
     NETDEL = NETDEL + MOE(2,J)*MOE(8,J)
     NETQUE = NETQUE + MOE(4,J)
2500 CONTINUE
     WRITE(16) DATE,TIMF,NBAD,NFLAG,NSDET,NETVOL,NETDEL,NETQUE,((INDOUS
    1(J),(MOE(I,J),I=1,8)),J=1,270)
3000 CONTINUE
     STOP
     END
```

Figure 7. Sample listing for program PLUGNET (continued).

(b) for the entire control pattern/time of day file. The former are called "individual" means to distinguish them from the latter, "grand" means.

The program then uses these means to substitute for individual data element values which have been identified as unusable but which correspond to links and observations which in general have been deemed usable. Grand means are used only where an appropriate individual mean could not be calculated. Since total row and column integrity is maintained for the ultimate statistical analysis of the data, this "plugging" of minor "holes" in the data prevents wholesale deletion of all data for whole links and observations that would otherwise be necessary.

Finally, the program calculates the total volume over all usable links for each observation and the total vehicle seconds of delay and average number of vehicles queued over all usable multiple-detector links per cycle for each observation. These values are then incorporated into the temporary data set generated by this program, which, aside from this expansion and the plugging of unusable values with mean values, is equivalent to the input data file.

The next program, SCTRGM (Figure 8), produces a scattergram of all multi-detector individual data elements across all observations in the control pattern/time of day file under consideration. The horizontal axis is the hourly link lane volume in units of eight for a maximum displayable value of 130 X 8 = 1040 vehicles per hour. The vertical axis is average delay per vehicle in units of three seconds for a maximum displayable value of 60 X 3 = 180 seconds. The numbers on the scattergram represent density ranges for the number of data points falling within the corresponding "box" on the plot. The scattergram is intended to give a useful, graphical summary of the data for that control pattern and time of day.

This is followed by a second sort of the latest temporary detector summary files, which orders 15-minute periods in ascending order by network volume (calculated by PLUGNET) <u>within</u> the primary sorting of observations by clock time. This sort is intended to decrease sorting time in later steps, although it is not absolutely necessary.

The program BRKFIL (see Figure 9) is used to disaggregate each large control pattern/time of day temporary file into several smaller permanent files that will be more easily handled in subsequent data manipulations.

```
MEMBER NAME SCTRGM

C       PROGRAM SCTRGM
C
C       PROGRAM TO PRODUCE A SCATTERGRAM OF DELAY VS VOLUME FOR A LINK
C
      INTEGER*2 MOE(8,270),INDOUS(270),MAP(61,131),MAPKEY(8),KEY(10)
      INTEGER*2 ISTAR,LUSE(270)
      INTEGER NOBS,DATE(3),TIME,TITLE(18)
      DATA MAPKEY/1,3,5,10,20,30,50,75/,MAP/7991*0/,KEY/' ','1','2','3',
     1'4','5','6','7','8','9'/,ISTAR/'*'/
      READ(9,10) (TITLE(I),I=1,18)
   10 FORMAT(18A4)
      READ(10,20) NOBS
   20 FORMAT(I10)
      READ(11) (LUSE(J),J=1,270)
      DO 1000 K7=1,NOBS
      READ(12) DATE,TIME,NBAD,NFLAG,NSDET,NETVOL,NETDEL,NETQUE,((INDOUS(
     1J),(MOE(I,J),I=1,8)),J=1,270)
      DO 500 J=1,270
      IF(LUSE(J).NE.1) GO TO 500
      IM = 60 - MIN0(59,(MOE(8,J)/3))
      JM = MIN0(129,(MOE(2,J)/8)) + 2
      MAP(IM,JM) = MAP(IM,JM) + 1
  500 CONTINUE
 1000 CONTINUE
      DO 1300 IM=1,60
      DO 1300 JM=2,131
      IF(MAP(IM,JM).EQ.0) GO TO 1250
      DO 1200 KL=1,8
      IF(MAP(IM,JM).LE.MAPKEY(KL)) GO TO 1260
 1200 CONTINUE
      KL = 9
      GO TO 1260
 1250 KL = 0
 1260 MAP(IM,JM) = KEY(KL+1)
 1300 CONTINUE
      DO 1310 IM=1,61
 1310 MAP(IM,1) = ISTAR
      DO 1320 JM=1,131
 1320 MAP(61,JM) = ISTAR
      WRITE(6,1330) (TITLE(I5),I5=1,18)
 1330 FORMAT('1',18A4,/)
      WRITE(6,1340) ((MAP(IM,JM),JM=1,131),IM=1,61)
 1340 FORMAT(' ',131A1)
      STOP
      END
```

Figure 8. Sample listing for program SCTRGM.

```
MEMBER NAME BRKFIL

C       PROGRAM BRKFIL
C
C       A PROGRAM TO CREATE SEPARATE HEADER AND INDIVIDUAL MOE
C       FILES FOR INPUT TO THE PRE-POST-PROCESSOR
C
      INTEGER*2 MOE(8,270),INDOUS(270)
      INTEGER DATE(3),TIME
      READ(9,10) NOBS
   10 FORMAT(I10)
      DO 1000 K=1,NOBS
      READ(10) DATE,TIME,NBAD,NFLAG,NSDET,NETVOL,NETDEL,NETQUE,((INDOUS(
     1J),(MOE(I,J),I=1,8)),J=1,270)
      WRITE(11) K,DATE,TIME,NBAD,NFLAG,NSDET,NETVOL,NETDEL,NETQUE
      WRITE(12) K,(INDOUS(J),J=1,270)
      DO 500 L=2,8
      IUN = 11 + L
      WRITE(IUN) K,(MOE(L,J),J=1,270)
  500 CONTINUE
 1000 CONTINUE
      STOP
      END
```

Figure 9. Sample listing for program BRKFIL.

Finally, the program ENDPRNT (Figure 10) produces a printed summary and sample listing of some of the detailed data for this control pattern and time of day. This printout is meant to aid checking, analysis, and human data quality control.

STEP 4

The following steps all either prepare for or perform statistical comparisons. Thus, sets of files for two control patterns (assumedly for the same time of day) are involved in each such run and so at least two of the Step 3 chains must be run to provide adequate input for Step 4, which itself has several parts.

The sample JCL in Figure 11 first executes the program NETCON (see Figure 12). Since each control pattern/time of day has a different vector describing the usability of the 270 links, it is necessary for any given comparison to establish a single consistent vector of link usability--a vector which classifies a link as usable for statistical comparison only if it is classified as usable in the two separate vectors and an optional, exogenously-supplied vector of link usability. Based on this new vector, consistent total "network" volumes are calculated for each observation in each of the two control pattern/time of day cases. These two sets of consistent network volumes will be used later in the "matching" process that will define a consistent set of observations for all statistical comparisons beteween these two control pattern/time of day cases. The results of this program are stored in new data files.

Using the IBM SORT/MERGE package again, the file containing the observation descriptors and the corresponding values of network volume for the two control pattern/time of day cases are sorted so as to be in ascending order according to network volume within the primary ordering according to clock time. This sorting will facilitate the "matching" of observations to be used in the statistical analysis.

The sorted files that result are the primary inputs to the program PREPOST (Figure 13). This program insures that both the total number of observations and the number of observations in each 15-minute period for the two control pattern/time of day cases are equal. The program also insures that the volume ranges for observations in each 15-minute period do not differ by more than a certain amount as between the two cases. This matching process is the final step in assuring total data integrity by link and observation for statistical comparisons and also guards

```
MEMBER NAME ENDPRNT

C       PROGRAM ENDPRNT
C
C       A PROGRAM TO PRODUCE A SUMMARY REPORT OF A TIME OF DAY/PATTERN
C       FILE AFTER FILE MANIPULATION IS COMPLETED
C
      INTEGER*2 MOE(270),LUSE(270)
      INTEGER DATE(3),TIME
      READ(9,10) NOBS
   10 FORMAT(I10)
      READ(10) (LUSE(J),J=1,270)
      WRITE(6,20) (LUSE(J),J=1,270)
   20 FORMAT('1','LUSE(J)',/,(' ',72I1))
      WRITE(6,100)
  100 FORMAT(//,' L1   K',6X,'DATE',6X,'TIME',' NBAD NFLAG NSDET  NETVOL
     1  NETDEL  NETQUE',/)
      DO 1000 L1=1,NOBS
      READ(11) K,DATE,TIME,NBAD,NFLAG,NSDET,NETVOL,NETDEL,NETQUE
      WRITE(6,520) L1,K,DATE,TIME,NBAD,NFLAG,NSDET,NETVOL,NETDEL,NETQUE
  520 FORMAT(' ',I2,I4,2X,3A4,2X,A4,I5,2I6,3I8)
 1000 CONTINUE
      REWIND 11
      WRITE(6,1100)
 1100 FORMAT('1',5X)
      DO 2000 L1=1,NOBS
      READ(11) K,DATE,TIME,NBAD,NFLAG,NSDET,NETVOL,NETDEL,NETQUE
      READ(12) K,(MOE(J),J=1,270)
      WRITE(6,1200) K,DATE(1),DATE(2),TIME
 1200 FORMAT(' ',I2,2X,2A4,1X,A4)
      WRITE(6,1300) (MOE(J),J=1,270)
      READ(13) K,(MOE(J),J=1,270)
      WRITE(6,1300) (MOE(J),J=1,270)
 1300 FORMAT(13(' ',19X,20I5,/),' ',19X,10I5,/)
 2000 CONTINUE
      STOP
      END
```

Figure 10. Sample listing for program ENDPRINT.

```
//NETCON EXEC PGM=NETCON
//STEPLIB DD DSN=PMMCO.UTCS.LOAD2,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FTC6F001 DD SYSOUT=A
//FT09F001 DD *
       0
//FT10F001 DD DSN=PMMCO.UTCS.NOBS(AM1),DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA,LABEL=(,,,IN)
//FT11F001 DD DSN=PMMCO.UTCS.NOBS(AM2),DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA,LABEL=(,,,IN)
//FT12F001 DD DSN=PMMCO.UTCS.LUSEAM1,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT13F001 DD DSN=PMMCO.UTCS.LUSEAM2,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT14F001 DD *

//FT15F001 DD DSN=PMMCO.UTCS.LUSGAM12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=544,BLKSIZE=548),
// SPACE=(548,(1,1))
//FT16F001 DD DSN=PMMCO.UTCS.HEADAM1,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
// DD DSN=PMMCO.UTCS.HEADAM2,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT17F001 DD DSN=PMMCO.UTCS.MOE2AM1,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
// DD DSN=PMMCO.UTCS.MOE2AM2,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT18F001 DD DSN=PMMCO.UTCS.VOLAM12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=28,BLKSIZE=1061),
// SPACE=(1061,(3,1))
//FT19F001 DD DSN=PMMCO.UTCS.VOLAM21,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=28,BLKSIZE=1061),
// SPACE=(1061,(3,1))
//SORT3 EXEC PGM=SORT,PARM='MSG=AP,CORE=100000',COND=(0,LT)
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SYSIN DD *
 SORT FIELDS=(21,4,CH,A,25,4,BI,A),SIZE=E85
//SORTIN DD DSN=PMMCO.UTCS.VOLAM12,DISP=(OLD,DELETE,KEEP),
// UNIT=SYSDA,VOL=SER=OTHERS
//SORTOUT DD DSN=PMMCO.UTCS.SRT3AM12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=28,BLKSIZE=1061),
// SPACE=(1061,(3,1))
//SORT4 EXEC PGM=SORT,PARM='MSG=AP,CORE=100000',COND=(0,LT)
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
```

Figure 11. Sample listing of JCL for program NETCON.

```
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SYSIN DD *
 SORT FIELDS=(21,4,CH,A,25,4,BI,A),SIZE=E85
//SORTIN DD DSN=PMMCO.UTCS.VOLAM21,DISP=(OLD,DELETE,KEEP),
// UNIT=SYSDA,VOL=SER=OTHERS
//SORTOUT DD DSN=PMMCO.UTCS.SRT3AM21,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=28,BLKSIZE=1061),
// SPACE=(1061,(3,1))
//PREPST EXEC PGM=PREPOST,COND=(0,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD2,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT10F001 DD DSN=PMMCO.UTCS.NOBS(AM1),DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT11F001 DD DSN=PMMCO.UTCS.NOBS(AM2),DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT15F001 DD DSN=PMMCO.UTCS.SRT3AM12,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT16F001 DD DSN=PMMCO.UTCS.SRT3AM21,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT18F001 DD DSN=PMMCO.UTCS.OBUSAM12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=104,BLKSIZE=108),
// SPACE=(108,(1,1))
```

Figure 11. Sample listing of JCL for program NETCON (continued).

```
'JHKA.JHKB.NETCON
 C       PROGRAM NETCON
 C
 C       A PROGRAM TO ESTABLISH A CONSISTANT SET OF LINKS FOR A NETWORK
 C       AND CALCULATE NETWORK VOLUMES
 C
       INTEGER*2 LUSE1(270),LUSE2(270),LUSEM(270),LUSEG(270),MOE2(270)
       INTEGER DATE(3),TIME
       DATA LUSEM/270*1/
       READ(5,10) ILUSEM
    10 FORMAT(I10)
       READ(10,20) NOBS1
       READ(11,20) NOBS2
    20 FORMAT(I10)
       READ(12) (LUSE1(J),J=1,270)
       READ(13) (LUSE2(J),J=1,270)
       IF(ILUSEM.EQ.1) READ(5,30) (LUSEM(J),J=1,270)
    30 FORMAT(72I1)
       DO 100 J=1,270
       LU1 = LUSE1(J)
       LU2 = LUSE2(J)
       LUM = LUSEM(J)
       LUSEG(J) = MIN0(LU1,LU2,LUM)
       IF(LUSEG(J).NE.0) LUSEG(J) = MAX0(LU1,LU2)
   100 CONTINUE
       WRITE(15) (LUSEG(J),J=1,270)
       WRITE(6,110) (LUSEG(J),J=1,270)
   110 FORMAT(//,' ','LUSEG(J) :',/,(' ',100I1))
       WRITE(6,150)
   150 FORMAT(///,' ','K       DATE     TIME  NETVOL',/)
       DO 500 L=1,NOBS1
       READ(16) K,DATE,TIME,NBAD,NFLAG,NSDET,NETVOL,NETDEL,NETQUE
       READ(17) K,(MOE2(J),J=1,270)
       NETVOL= 0
       DO 400 J=1,270
       IF(LUSEG(J).EQ.0) GO TO 400
       NETVOL = NETVOL + MOE2(J)
   400 CONTINUE
       WRITE(18) K,DATE,TIME,NETVOL
       WRITE(6,510) K,DATE,TIME,NETVOL
   500 CONTINUE
   510 FORMAT(' ',I3,4A4,I8)
       WRITE(6,150)
       DO 1000 L=1,NOBS2
       READ(16) K,DATE,TIME,NBAD,NFLAG,NSDET,NETVOL,NETDEL,NETQUE
       READ(17) K,(MOE2(J),J=1,270)
       NETVOL = 0
       DO 600 J=1,270
       IF(LUSEG(J).EQ.0) GO TO 600
       NETVOL = NETVOL + MOE2(J)
   600 CONTINUE
       WRITE(19) K,DATE,TIME,NETVOL
       WRITE(6,510) K,DATE,TIME,NETVOL
  1000 CONTINUE
       STOP
       END
```

Figure 12. Sample listing for program NETCON.

```
'JHKA.JHKB.PREPOST
      PROGRAM PREPOST

      A PROGRAM TO PREPARE FINAL DATA FOR TWO PATTERNS FOR THE POST
      PROCESSOR

      INTEGER*2 OBUSE1(100),OBUSE2(100)
      INTEGER K1(100),K2(100),DATE1(3,100),DATE2(3,100),TIME1(100)
      INTEGER TIME2(100),NTVOL1(100),NTVOL2(100),NVOL2(100)
      INTEGER NIT1(15),NIT2(15),TIMD1(100),TIMD2(100),NVOL1(100)
      INTEGER KNT1(100),KNT2(100),DATD1(3,100),DATD2(3,100)
      DATA NIT1/15*0/,NIT2/15*0/,I1S/1/,I2S/1/,IT/1/
      DATA OBUSE1/100*1/,OBUSE2/100*1/
      DATA IEND1/0/,IEND2/0/
      READ(10,20) NOBS1
      READ(11,20) NOBS2
   20 FORMAT(I10)
      DO 50 I=1,NOBS1
   50 READ(15) K1(I),(DATE1(K,I),K=1,3),TIME1(I),NTVOL1(I)
      DO 60 I=1,NOBS2
   60 READ(16) K2(I),(DATE2(K,I),K=1,3),TIME2(I),NTVOL2(I)
  200 CALL READIT(I1S,I2S,NIT1,NIT2,IT,OBUSE1,OBUSE2,IEND1,IEND2,TIME1,T
     1IME2,NOBS1,NOBS2,K1,K2)
      I1SP = I1S + NIT1(IT)
      I2SP = I2S + NIT2(IT)
      CALL MATCH(I1S,I2S,NIT1,NIT2,IT,OBUSE1,OBUSE2,NTVOL1,NTVOL2,TIME1,
     1TIME2,IEND1,IEND2,K1,K2)
      I1S = I1SP
      I2S = I2SP
      IF(IEND1*IEND2.EQ.1) GO TO 300
      IT = IT + 1
      GO TO 200
  300 WRITE(18) (OBUSE1(J1),J1=1,NOBS1),(OBUSE2(J2),J2=1,NOBS2)
      WRITE(6,450) (OBUSE1(K1(J1)),J1=1,NOBS1)
      WRITE(6,450) (OBUSE2(K2(J2)),J2=1,NOBS2)
  450 FORMAT(//,' ','OBUSE:',/,' ',100I1)
      NE1 = 0
      DO 500 I=1,NOBS1
      IF(OBUSE1(K1(I)).EQ.0) GO TO 500
      NE1 = NE1 + 1
      KNT1(NE1) = K1(I)
      DATD1(1,NE1) = DATE1(1,I)
      DATD1(2,NE1) = DATE1(2,I)
      DATD1(3,NE1) = DATE1(3,I)
      TIMD1(NE1) = TIME1(I)
      NVOL1(NE1) = NTVOL1(I)
  500 CONTINUE
      NE2 = 0
      DO 1000 I=1,NOBS2
      IF(OBUSE2(K2(I)).EQ.0) GO TO 1000
      NE2 = NE2 + 1
```

Figure 13. Sample listing for program PREPOST.

```
      KNT2(NE2) = K2(I)
      DATD2(1,NE2) = DATE2(1,I)
      DATD2(2,NE2) = DATE2(2,I)
      DATD2(3,NE2) = DATE2(3,I)
      TIMD2(NE2) = TIME2(I)
      NVOL2(NE2) = NTVOL2(I)
 1000 CONTINUE
      WRITE(6,1100)
 1100 FORMAT(//,' ','K1      DATE     TIME    VOL1        VOL2    TIME    DATE
     1      K2',/)
      DO 1200 I=1,NE2
 1200 WRITE(6,1300) KNT1(I),(DATD1(J,I),J=1,3),TIMD1(I),NVOL1(I),NVOL2(I
     1),TIMD2(I),(DATD2(J,I),J=1,3),KNT2(I)
 1300 FORMAT(' ',I2,1X,4A4,I8,I10,2X,A4,1X,3A4,I2)
      STOP
      END
C
      SUBROUTINE READIT(I1S,I2S,NIT1,NIT2,IT,OBUSE1,OBUSE2,IEND1,IEND2,T
     1IME1,TIME2,NOBS1,NOBS2,K1,K2)
      INTEGER*2 OBUSE1(100),OBUSE2(100)
      INTEGER NIT1(15),NIT2(15),TIME1(100),TIME2(100)
      INTEGER K1(100),K2(100)
      DATA NINE/'9999'/
   50 IF(TIME1(I1S).NE.TIME2(I2S)) GO TO 500
  100 NIT1(IT) = NIT1(IT) + 1
      IF((I1S+NIT1(IT)-1).EQ.NOBS1) GO TO 150
      IF(TIME1(I1S+NIT1(IT)).EQ.TIME1(I1S)) GO TO 100
      GO TO 200
  150 IEND1 = 1
  200 NIT2(IT) = NIT2(IT) + 1
      IF((I2S+NIT2(IT)-1).EQ.NOBS2) GO TO 250
      IF(TIME2(I2S+NIT2(IT)).EQ.TIME2(I2S)) GO TO 200
      GO TO 300
  250 IEND2 = 1
  300 RETURN
  500 IF(TIME1(I1S).LT.TIME2(I2S)) GO TO 1000
  550 NIT2(IT) = NIT2(IT) + 1
      OBUSE2(K2(I2S+NIT2(IT)-1)) = 0
      IF((I2S+NIT2(IT)-1).EQ.NOBS2) GO TO 1100
      IF(TIME2(I2S+NIT2(IT)).EQ.TIME2(I2S)) GO TO 550
      IT = IT + 1
      I2S = I2S + NIT2(IT-1)
      GO TO 50
 1000 NIT1(IT) = NIT1(IT) + 1
      OBUSE1(K1(I1S+NIT1(IT)-1)) = 0
      IF((I1S+NIT1(IT)-1).EQ.NOBS1) GO TO 1200
      IF(TIME1(I1S+NIT1(IT)).EQ.TIME1(I1S)) GO TO 1000
      IT = IT + 1
      I1S = I1S + NIT1(IT-1)
      GO TO 50
 1100 IEND2 = 1
```

Figure 13. Sample listing for program PREPOST (continued).

```
      IT = IT + 1
      I2S = I2S + NIT2(IT-1)
      TIME2(I2S) = NINE
      IF(IEND1*IEND2.EQ.1) RETURN
      GO TO 500
 1200 IEND1 = 1
      IT = IT + 1
      I1S = I1S + NIT1(IT-1)
      TIME1(I1S) = NINE
      IF(IEND1*IEND2.EQ.1) RETURN
      GO TO 500
      END
C
      SUBROUTINE MATCH(I1S,I2S,NIT1,NIT2,IT,OBUSE1,OBUSE2,NTVOL1,NTVOL2,
     1TIME1,TIME2,IEND1,IEND2,K1,K2)
      INTEGER*2 OBUSE1(100),OBUSE2(100)
      INTEGER NIT1(15),NIT2(15),NTVOL1(100),NTVOL2(100),TIME1(100)
      INTEGER TIME2(100),IND(80),K1(100),K2(100)
      REAL DEL(80)
      DATA NINE/'9999'/
      IF(((TIME1(I1S).EQ.NINE).OR.(TIME2(I2S).EQ.NINE)).AND.(IEND1*IEND2
     1.EQ.1)) RETURN
      NITA = NIT1(IT)
      NITB = NIT2(IT)
      ICHG = 0
      VBAR1 = 0.
      VBAR2 = 0.
      IF((NITA.EQ.0).OR.(NITB.EQ.0)) GO TO 2000
      MIN = MIN0(NTVOL1(I1S),NTVOL2(I2S))
      MAX = MAX0(NTVOL1(I1S+NITA-1),NTVOL2(I2S+NITB-1))
      NUM = MAX0(NITA,NITB) -1
      IF(NUM.EQ.0) GO TO 50
      ICRIT = (MAX-MIN)/NUM
      GO TO 100
   50 ICRIT = (NTVOL1(I1S)+NTVOL2(I2S))/20
  100 IF((NTVOL1(I1S)-NTVOL2(I2S)).GT.ICRIT) GO TO 500
      IF((NTVOL2(I2S)-NTVOL1(I1S)).GT.ICRIT) GO TO 600
      GO TO 700
  500 OBUSE2(K2(I2S)) = 0
      ICHG = 1
      I2S = I2S + 1
      NITB = NITB - 1
      GO TO 700
  600 OBUSE1(K1(I1S)) = 0
      ICHG = 1
      I1S = I1S + 1
      NITA = NITA -1
  700 IF((NITA.EQ.0).OR.(NITB.EQ.0)) GO TO 2000
      IF((NTVOL1(I1S+NITA-1)-NTVOL2(I2S+NITB-1)).GT.ICRIT) GO TO
     1 800
      IF((NTVOL2(I2S+NITB-1)-NTVOL1(I1S+NITA-1)).GT.ICRIT) GO TO
```

Figure 13. Sample listing for program PREPOST (continued).

```
    1 900
      GO TO 1100
  800 OBUSE1(K1(I1S+NITA-1))= 0
      ICHG = 1
      NITA = NITA - 1
      GO TO 1000
  900 OBUSE2(K2(I2S+NITB-1)) = 0
      ICHG = 1
      NITB = NITB - 1
 1000 IF((NITA.EQ.0).OR.(NITB.EQ.0)) GO TO 2000
 1100 IF(ICHG.EQ.0) GO TO 1200
      ICHG = 0
      GO TO 100
 1200 IF(NITA.EQ.NITB) GO TO 1500
      IF(NITA.EQ.1) GO TO 1450
      IF(NITB.EQ.1) GO TO 1470
      IF(NITA.GT.NITB) GO TO 1300
      VBAR2 = 0.
      DO 1250 I=1,NITB
 1250 VBAR2 = VBAR2 + NTVOL2(I2S+I-1)
      VBAR2 = VBAR2/(NITB+0.)
      DO 1255 I=1,NITB
 1255 DEL(I) = ABS(NTVOL2(I2S+I-1)-VBAR2)
      CALL SORTR(NITB,DEL,IND)
      NDEL = NITB - NITA
      DO 1260 I=1,NDEL
 1260 OBUSE2(K2(I2S+IND(I)-1)) = 0
      GO TO 1500
 1300 VBAR1 = 0
      DO 1350 I=1,NITA
 1350 VBAR1 = VBAR1 + NTVOL1(I1S+I-1)
      VBAR1 = VBAR1/(NITA+0.)
      DO 1355 I=1,NITA
 1355 DEL(I) = ABS(NTVOL1(I1S+I-1)-VBAR1)
      CALL SORTR(NITA,DEL,IND)
      NDEL = NITA - NITB
      DO 1360 I=1,NDEL
 1360 OBUSE1(K1(I1S+IND(I)-1)) = 0
      GO TO 1500
 1450 ITEST = IABS(NTVOL1(I1S)-NTVOL2(I2S))
      KEEP = I2S
      DO 1455 I=2,NITB
      IF(IABS(NTVOL1(I1S)-NTVOL2(I2S+I-1)).LT.ITEST) KEEP = I2S + I - 1
 1455 ITEST = MIN0(ITEST,IABS(NTVOL1(I1S)-NTVOL2(I2S+I-1)))
      DO 1460 I=1,NITB
      IF(I.NE.(KEEP-I2S+1)) OBUSE2(K2(I+I2S-1)) = 0
 1460 CONTINUE
      GO TO 1500
 1470 ITEST = IABS(NTVOL2(I2S)-NTVOL1(I1S))
      KEEP = I1S
      DO 1475 I=2,NITA
```

Figure 13. Sample listing for program PREPOST (continued).

```
      IF(IABS(NTVOL2(I2S)-NTVOL1(I1S+I-1)).LT.ITEST) KEEP = I1S + I - 1
 1475 ITEST = MIN0(ITEST,IABS(NTVOL2(I2S)-NTVOL1(I1S+I-1)))
      DO 1480 I=1,NITA
      IF(I.NE.(KEEP-I1S+1)) OBUSE1(K1(I+I1S-1)) = 0
 1480 CONTINUE
 1500 RETURN
 2000 IF(NITA.NE.0) GO TO 2100
      IF(NITB.NE.0) GO TO 2200
      GO TO 2300
 2100 DO 2150 J79=1,NITA
 2150 OBUSE1(K1(I1S+J79-1)) = 0
      GO TO 2300
 2200 DO 2250 J79=1,NITB
 2250 OBUSE2(K2(I2S+J79-1)) = 0
 2300 RETURN
      END
C
C      SUBROUTINE SORTR
C
      SUBROUTINE SORTR(N,V,IND)
      INTEGER IND(2)
      REAL V(2),DV(100)
      DO 5 I=1,N
    5 DV(I)=V(I)
      NM=N-1
      DO 7 I=1,N
    7 IND(I) = I
   10 K=0
      DO 100 I=1,NM
      IF(DV(I).GT.DV(I+1)) CALL SWITCH(DV(I),DV(I+1),IND(I),IND(I+1),K)
  100 CONTINUE
      IF(K.NE.0) GO TO 10
      RETURN
      END
C
C      SUBROUTINE SWITCH
C
      SUBROUTINE SWITCH(L,M,IL,IM,K)
      REAL L,M
      K=1
      DL=L
      DM=M
      IDL=IL
      IDM=IM
      L=DM
      M=DL
      IL=IDM
      IM=IDL
      RETURN
      END
```

Figure 13. Sample listing for program PREPOST (continued).

against the possibility of comparing traffic control patterns based on two sets of observations which "unfairly" differ in overall volume conditions. The program outputs vectors which are used in statistical comparisons (based on the above match) and also a printed table of the observations in the two cases which have been selected in the matching process.

STEP 5

An optional step that can be appended to the end of the above string is implemented by a JCL deck such as the one illustrated in Figure 14. Should the investigator wish to make a statistical analysis of measures of effectiveness aggregated over specified "subnetworks" of the entire set of usable links, in addition to the standard link-based and observation-based analysis, the execution of program CONDNS (see Figure 15) provides for this capability.

For each of four measures of effectiveness which can be meaningfully aggregated over links, this program constructs a subnetwork level data file. These files are analogous to the permanent MOE link files created in Step 3, but instead of containing 270 individual link MOE values, the four new files contain a specified number of subnetwork MOE aggregate values defined over a corresponding number of pre-specified link aggregates (or subnetworks).

To insure that the network aggregation is always over a consistent set of actually used links, the LUSGXXYZ link usability vector generated by NECTCON must be used and the program CONDNS must be executed twice for each different pair of cases to be compared--once for each of the two cases, given the other with which it is to be compared subsequently.

STEP 6

Of the data preparation completed, it now remains only to make the final statistical analysis in comparing the two traffic control patterns under consideration for a given time of day.

The sample JCL in Figure 16 executes the program POSTPROC (Figure 17), the modified statistical post-processor. It can be applied either to link level data or the subnetwork level data generated in the optional Step 5.

The program makes comparisons on the basis of one measure of effectiveness at a time. It can make its statistical comparisons for each usable link across all observations and/or for each "matched" pair of observation across all usable links.

```
//CONDNS1 EXEC PGM=CONDNS
//STEPLIB DD DSN=PMMCO.UTCS.LOAD2,VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT08F001 DD *
CONTROL PATTERN 1; A.M.
//FT09F001 DD *
         3       270
TOTL              -1
F5L6              11
    1    2    3    4    5  265  266  267  268  269  270
BAD                3
  268  269  270
//FT10F001 DD DSN=PMMCO.UTCS.LUSGAM12,VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA
//FT11F001 DD DSN=PMMCO.UTCS.NOBS(AM1),VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA,LABEL=(,,,IN)
//FT12F001 DD DSN=PMMCO.UTCS.MOE2AM1,VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA
//FT13F001 DD DSN=PMMCO.UTCS.MOE4AM1,VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA
//FT14F001 DD DSN=PMMCO.UTCS.MOE5AM1,VOL=SER=OTHERS,DISP=SHR,
// UNNT=SYSDA
//FT15F001 DD DSN=PMMCO.UTCS.MOE7AM1,VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA
//FT16F001 DD DSN=PMMCO.UTCS.MOE8AM1,VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA
//FT17F001 DD DSN=PMMCO.UTCS.CONQAM12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061),
// SPACE=(1061,(7,1))
//FT18F001 DD DSN=PMMCO.UTCS.CONSAM12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061),
// SPACE=(1061,(7,1))
//FT19F001 DD DSN=PMMCO.UTCS.CONTAM12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061),
// SPACE=(1061,(7,1))
//FT20F001 DD DSN=PMMCO.UTCS.LUSCAM12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=44,BLKSIZE=48),
// SPACE=(48,(1,1))
//FT21F001 DD DSN=PMMCO.UTCS.LBLAM12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
// SPACE=(800,(2,1))
//FT23F001 DD DSN=PMMCO.UTCS.CONDAM12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061),
// SPACE=(1061,(7,1))
//CONDNS2 EXEC PGM=CONDNS
```

Figure 14. Sample listing of JCL for program CONDNS.

```
//STEPLIB DD DSN=PMMCO.UTCS.LOAD2,VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT08F001 DD *
CONTROL PATTERN 1; A.M.
//FT09F001 DD *
        3        270
TOTL              -1
F5L6              11
   1    2    3    4    5  265  266  267  268  269  270
BAD                3
  268  269  270
//FT10F001 DD DSN=PMMCO.UTCS.LUSGAM12,VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA
//FT11F001 DD DSN=PMMCO.UTCS.NOBS(AM2),VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA,LABEL=(,,,IN)
//FT12F001 DD DSN=PMMCO.UTCS.MOE2AM2,VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA
//FT13F001 DD DSN=PMMCO.UTCS.MOE4AM2,VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA
//FT14F001 DD DSN=PMMCO.UTCS.MOE5AM2,VOL=SER=OTHERS,DISP=SHR,
// UNNT=SYSDA
//FT15F001 DD DSN=PMMCO.UTCS.MOE7AM2,VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA
//FT16F001 DD DSN=PMMCO.UTCS.MOE8AM2,VOL=SER=OTHERS,DISP=SHR,
// UNIT=SYSDA
//FT17F001 DD DSN=PMMCO.UTCS.CONQAM21,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061),
// SPACE=(1061,(7,1))
//FT18F001 DD DSN=PMMCO.UTCS.CONSAM21,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061),
// SPACE=(1061,(7,1))
//FT19F001 DD DSN=PMMCO.UTCS.CONTAM21,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061),
// SPACE=(1061,(7,1))
//FT20F001 DD DSN=PMMCO.UTCS.LUSCAM21,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=44,BLKSIZE=48),
// SPACE=(48,(1,1))
//FT21F001 DD DSN=PMMCO.UTCS.LBLAM21,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
// SPACE=(800,(2,1))
//FT23F001 DD DSN=PMMCO.UTCS.CONDAM21,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061),
// SPACE=(1061,(7,1))
```

Figure 14. Sample listing of JCL for program CONDNS (continued).

MEMBER NAME CONDNS

```
C       PROGRAM CONDNS
C
C       A PROGRAM TO CREATE NETWORK MOE FILES
C
      INTEGER*2 MOE2(270),MOE4(270),MOE5(270),MOE7(270),MOE8(270)
      INTEGER*2 LUSEG(270),TABLE(20,150),NLINK(20),NLINK2(20),DUM(270)
      INTEGER LNLBL(20),TITLE(18)
      REAL CONQ(20),CONS(20),CONT(20),COND(20)
      DATA TABLE/3000*0/,NLINK2/20*0/
      READ(8,5) (TITLE(I),I=1,18)
    5 FORMAT(18A4)
      READ(9,10) NNET,NLKNT
   10 FORMAT(2I10)
      READ(10) (LUSEG(J),J=1,NLKNT)
      READ(11,20) NOBS
   20 FORMAT(I10)
      DO 100 I=1,NNET
      READ(9,30) LNLBL(I),NLINK(I)
   30 FORMAT(A4,6X,I10)
      IF(NLINK(I).EQ.-1) GO TO 40
      GO TO 60
   40 I2 = 0
      DO 50 I9=1,NLKNT
      IF(LUSEG(I9).NE.1) GO TO 50
      I2 = I2 + 1
      TABLE(I,I2) = I9
   50 CONTINUE
      NLINK(I) = I2
      GO TO 100
   60 NLNK = NLINK(I)
      READ(9,70) (DUM(J9),J9=1,NLNK)
   70 FORMAT(14I5)
      I88 = 0
      DO 80 I77=1,NLNK
      IF(LUSEG(DUM(I77)).NE.1) GO TO 80
      I88 = I88 + 1
      TABLE(I,I88) = DUM(I77)
   80 CONTINUE
  100 CONTINUE
      WRITE(6,110) (TITLE(I),I=1,18)
  110 FORMAT('1',18A4,//)
      DO 140 I1=1,NNET
      NLNK = NLINK(I1)
      DO 130 I2=1,NLNK
      IF(LUSEG(TABLE(I1,I2)).NE.1) GO TO 130
      NLINK2(I1) = NLINK2(I1) + 1
  130 CONTINUE
      LUSEG(I1) = 1
      IF(NLINK2(I1).LE.0) GO TO 135
      GO TO 140
  135 LUSEG(I1) = 0
      WRITE(6,136) I1,LNLBL(I1)
  136 FORMAT(//,' ','NETWORK # ',I2,', ',A4,'  HAD NO USABLE LINKS',/)
  140 CONTINUE
      WRITE(20) (LUSEG(J),J=1,NNET)
```

Figure 15. Sample listing for program CONDNS.

```
      WRITE(21,141) (LNLBL(J),J=1,NNET)
  141 FORMAT(18A4)
      WRITE(6,145)
  145 FORMAT(///,' ','OBSERV.  NETWORK', 9X,'QUEUES',11X,'STOPS',4X,'VEH
     1. MINUTES',3X,'MIN. OF DELAY',/)
      DO 500 I=1,NOBS
      READ(12) K,(MOE2(J),J=1,NLKNT)
      READ(13) K,(MOE4(J),J=1,NLKNT)
      READ(14) K,(MOE5(J),J=1,NLKNT)
      READ(15) K,(MOE7(J),J=1,NLKNT)
      READ(16) K,(MOE8(J),J=1,NLKNT)
      DO 150 I9=1,20
      CONQ(I9) = 0.
      CONS(I9) = 0.
      CONT(I9) = 0.
  150 COND(I9) = 0.
      DO 300 I1=1,NNET
      IF(LUSEG(I1).EQ.0) GO TO 300
      NLNK = NLINK2(I1)
      DO 250 I2=1,NLNK
      CONQ(I1) = CONQ(I1) + MOE4(TABLE(I1,I2)) + 0.
      CONS(I1) = CONS(I1) + MOE5(TABLE(I1,I2)) + 0.
      CONT(I1) = CONT(I1) + (MOE7(TABLE(I1,I2))*MOE2(TABLE(I1,I2))+0.)/1
     1920.
      COND(I1) = COND(I1) + (MOE8(TABLE(I1,I2))*MOE2(TABLE(I1,I2))+0.)/6
     10.
  250 CONTINUE
  300 CONTINUE
      DO 325 K7=1,NNET
  325 WRITE(6,350) K,LNLBL(K7),CONQ(K7),CONS(K7),CONT(K7),COND(K7)
  350 FORMAT(' ',I6,5X,A4,4F16.0)
      WRITE(17) K,(CONQ(J),J=1,NNET)
      WRITE(18) K,(CONS(J),J=1,NNET)
      WRITE(19) K,(CONT(J),J=1,NNET)
      WRITE(23) K,(COND(J),J=1,NNET)
  500 CONTINUE
      STOP
      END
```

Figure 15. Sample listing for program CONDNS (continued).

```
//POSTP EXEC PGM=POSTPROC,TIME=(4,0)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD2,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT09F001 DD *
    1    2 A.M.   80   75    1    2    1  270    3
//FT10F001 DD *
AVERAGE DELAY
//FT11F001 DD DSN=PMMCO.UTCS.LBLAM12,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT12F001 DD DSN=PMMCO.UTCS.LUSGAM12,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT13F001 DD DSN=PMMCO.UTCS.OBUSAM12,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT15F001 DD DSN=PMMCO.UTCS.MOE8AM1,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT16F001 DD DSN=PMMCO.UTCS.MOE8AM2,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
```

Figure 16. Sample listing of JCL for program POSTPROC

```
'JHKA.JHKB.POSTPROC
C       PROGRAM POSTPROC
C
C      A PROGRAM TO PERFORM 3 SETS OF STATISTICAL TESTS ON A LINK AND/OR
C      LINK AGGREGATE BASIS FOR THE PURPOSE OF COMPARING THE EFFECTIVE-
C      NESS OF TWO CONTROL PATTERNS FOR A GIVEN TIME OF DAY
       REAL A(540),B(270),D(540)
       INTEGER*2 LUSEG(270),OBUSE1(100),OBUSE2(100)
       REAL MOE1(90,135),MOE2(90,135)
       REAL*8 NMSTAT,LNKNAM,IAGNAM,IOBSNM
       DIMENSION LAB1(20),LAB2(20)
       INTEGER PTRNA,PTRNB,TOD,NOBS,NOBS1,NOBS2,LNKNT,LNLBL(270)
       INTEGER NLKNT,TWIND
       INTEGER*2 IHMOE1(90,270),IHMOE2(90,270),IHA
       INTEGER IMOE1(90,135),IMOE2(90,135),NMOE,TESTBS
       INTEGER MOENAM(5,8),SIGP(5),SIGKS(3)
       NAMELIST /PARAM/ NMOE,TESTBS,LNKNT,TWIND
       EQUIVALENCE (IHMOE1,IMOE1,MOE1),(IHMOE2,IMOE2,MOE2)
       DATA SIGP/'    ','*   ','**  ','*** ',' NT '/,SIGKS/'<','>',' '/
       DATA LNKNAM/'LINKS   '/,IAGNAM/'AGGREG. '/,IOBSNM/'OBSERV. '/
       READ(5,PARAM)
       READ(5,10) LAB1,LAB2,NOBS1,NOBS2,NLKNT
    10 FORMAT(20A4/20A4/I10/I10/I10)
       WRITE(6,15) LAB1,LAB2,NOBS1,NOBS2,NLKNT,NMOE,TESTBS,LNKNT,TWIND
    15 FORMAT('1POST PROCESSOR CONTROL PARAMETERS'/'0CONDITION A ',20A4/
      1 '0CONDITION B ',20A4/'0NOBS1=',' NOBS2=',I6,' NLINKS=',I5/
      2 '0NMOE=',I7,' TESTBS=',I5,' LNKNT=',I6,' TWIND=',I6)
       READ(5,20) ((MOENAM(I,J),I=1,5),J=1,NMOE)
    20 FORMAT(5A4)
       IF(LNKNT.EQ.2) READ(11,40) (LNLBL(I),I=1,NLKNT)
    40 FORMAT(18A4)
       READ(12) (LUSEG(J),J=1,NLKNT)
       NUSLNK = 0
       DO 42 J=1,NLKNT
       IF(LUSEG(J).EQ.1) NUSLNK = NUSLNK + 1
    42 CONTINUE
       READ(13) (OBUSE1(J1),J1=1,NOBS1),(OBUSE2(J2),J2=1,NOBS2)
       NOBS = 0
       DO 45 I=1,NOBS1
       IF(OBUSE1(I).EQ.1) NOBS = NOBS + 1
    45 CONTINUE
       DO 1000 I=1,NMOE
       IF(TWIND-2) 200,300,400
   200 J = 0
       DO 210 J1=1,NOBS1
       IF(OBUSE1(J1).EQ.0) GO TO 209
       J = J + 1
       READ(15) K,(MOE1(J,I9),I9=1,NLKNT)
       GO TO 210
   209 READ(15) K,(AA,I9=1,NLKNT)
   210 CONTINUE
```

Figure 17. Sample listing for program POSTPROC.

```
      J = 0
      DO 220 J1=1,NOBS2
      IF(OBUSE2(J1).EQ.0) GO TO 219
      J = J + 1
      READ(16) K,(MOE2(J,I9),I9=1,NLKNT)
      GO TO 220
  219 READ(16) K,(AA,I9=1,NLKNT)
  220 CONTINUE
      GO TO 480
  300 J = 0
      DO 310 J1=1,NOBS1
      IF(OBUSE1(J1).EQ.0) GO TO 309
      J = J + 1 .
      READ(15) K,(IMOE1(J,I9),I9=1,NLKNT)
      GO TO 310
  309 READ(15) K,(IA,I9=1,NLKNT)
  310 CONTINUE
      J = 0
      DO 320 J1=1,NOBS2
      IF(OBUSE2(J1).EQ.0) GO TO 319
      J = J + 1
      READ(16) K,(IMOE2(J,I9),I9=1,NLKNT)
      GO TO 320
  319 READ(16) K,(IA,I9=1,NLKNT)
  320 CONTINUE
      GO TO 480
  400 J = 0
      DO 410 J1=1,NOBS1
      IF(OBUSE1(J1).EQ.0) GO TO 409
      J = J + 1
      READ(15) K,(IHMOE1(J,I9),I9=1,NLKNT)
      GO TO 410
  409 READ(15) K,(IHA,I9=1,NLKNT)
  410 CONTINUE
      J = 0
      DO 420 J1=1,NOBS2
      IF(OBUSE2(J1).EQ.0) GO TO 419
      J = J + 1
      READ(16) K,(IHMOE2(J,I9),I9=1,NLKNT)
      GO TO 420
  419 READ(16) K,(IHA,I9=1,NLKNT)
  420 CONTINUE
  480 KDO = 0
  490 KDO = KDO + 1
      NMSTAT = LNKNAM
      IF((TESTBS.EQ.3).OR.(KDO.EQ.2)) GO TO 495
      IF(NOBS.LT.3) GO TO 950
      IF(LNKNT.EQ.2) NMSTAT = IAGNAM
      GO TO 498
  495 IF(NUSLNK.LT.3) GO TO 970
      NMSTAT = IOBSNM
```

Figure 17. Sample listing for program POSTPROC (continued).

```
      IF((ISIGP.GE.1).AND.(ISIGP.NE.4).AND.(ISIGM.EQ.0)) IDIR=1
      IF((ISIGM.GE.1).AND.(ISIGM.NE.4).AND.(ISIGP.EQ.0)) IDIR=2
      ISIGX=ISIG2
      IF(IDIR.NE.3) ISIGX=MAX0(ISIGP,ISIGM)
      DO 540 I9=1,NMBR
  540 A(NMBR+I9)= B(I9)
      CALL CALC(A,B,NMBR,ABAR,BBAR,VARA,VARB)
      IF(NMBR.LT.4) GO TO 600
      IF(NMBR.LT.9) GO TO 580
      CALL TTEST(A,B,NMBR,ISIGT,D)
      TT=D(1)
  580 CALL UTEST(A,D,NMBR,NMBR,ISIGU)
      TU=D(1)
  600 STDVA = SQRT(VARA)
      STDVB = SQRT(VARB)
      DELBAR = ABAR - BBAR
      IF((ABAR+BBAR).EQ.0.0) GO TO 650
      DELPER = 200.*(DELBAR/(ABAR+BBAR))
      GO TO 700
  650 DELPER = 0.0
  700 IF((LNKNT.EQ.2).AND.(KDO.EQ.1)) GO TO 800
      WRITE(6,710)      J,ABAR,STDVA,BBAR,STDVB,DELPER,TT,SIGP(ISIGT+1),
     1TU,SIGP(ISIGU+1),ITP,ITM,IT2,SIGKS(
     2IDIR),SIGP(ISIGX+1)
  710 FORMAT(' ',I5,2X,4F10.1,F11.1,' %',2(F9.3,1X,A3),
     13X,3I4,2X,A1,A3)
      GO TO 900
  800 WRITE(6,810) LNLBL(J),ABAR,STDVA,BBAR,STDVB,DELPER,TT,SIGP(ISIGT+1
     1)          ,TU,SIGP(ISIGU+1),ITP,ITM,IT2,SIG
     2KS(IDIR),SIGP(ISIGX+1)
  810 FORMAT(' ',A5,2X,4F10.1,F11.1,' %',2(F9.3,1X,A3),
     13X,3I4,2X,A1,A3)
  900 CONTINUE
      IF((KDO.EQ.1).AND.(TESTBS.EQ.2)) GO TO 490
      GO TO 1000
  950 WRITE(6,960) NOBS
  960 FORMAT(////' ','TOO FEW OBSERVATIONS AVAILABLE TO TEST : ',I5,///)
      IF((KDO.EQ.1).AND.(TESTBS.EQ.2)) GO TO 490
      GO TO 1000
  970 WRITE(6,980) NUSLNK
  980 FORMAT(////' ','TOO FEW USABLE LINKS AVAILABLE TO TEST : ',I5,///)
 1000 CONTINUE
      STOP
      END
C
C
      SUBROUTINE READER(A,B,MOE1,MOE2,IMOE1,IMOE2,IHMOE1,IHMOE2,TWIND,
     1 LUSEG,NLKNT,NOBS,J,NMSTAT,IOBSNM)
      REAL A(540),B(270),MOE1(90,135),MOE2(90,135)
      INTEGER IMOE1(90,135),IMOE2(90,135),TWIND
      REAL*8 NMSTAT,IOBSNM
```

Figure 17. Sample listing for program POSTPROC (continued).

```
498 CONTINUE
    WRITE(6,50)
 50 FORMAT('1',24X,'STATISTICAL COMPARISON OF A MEASURE OF EFFECTIVENE
   1SS FOR CONDITIONS "A" AND "B"',/)
    WRITE(6,60) LAB1
    WRITE(6,70) LAB2
 60 FORMAT(' ',10X,'CONDITION A : ',20A4)
 70 FORMAT(' ',10X,'CONDITION B : ',20A4)
    IF(LNKNT.EQ.2) GO TO 90
    WRITE(6,80)
 80 FORMAT(' ',10X,'COMPARISONS MADE BY LINK',/)
    GO TO 110
 90 WRITE(6,100)
100 FORMAT(' ',10X,'COMPARISONS MADE BY LINK AGGREGATE',/)
110 IF((KDO.EQ.1).AND.(TESTBS.NE.3)) GO TO 115
    WRITE(6,120) NUSLNK
    GO TO 118
115 WRITE(6,120) NOBS
120 FORMAT(' ',10X,'STATISTICAL SAMPLE SIZE USED : ',I5,/)
118 WRITE(6,130) (MOENAM(J,I),J=1,5),NMSTAT
130 FORMAT(' ',10X,'BASIS OF COMPARISONS : ',5A4,'  BY ',A8,///)
    WRITE(6,140)
140 FORMAT(' ','LNKNT /',6X,'CONDITION A',9X,'CONDITION B',6X,'DIFFERE
   1NCE     T TEST        U TEST     KOLMOGOROV-SMIRNOV')
    WRITE(6,150)
150 FORMAT(' ','OBSERV.       MEAN       STDV       MEAN       STDV    OF
   1MEANS           T            U/Z          T+  T-    T2  SIG.')
    WRITE(6,160)
160 FORMAT(' ',7('-'),4(3X,'-------'),3(3X,10('-')),    3X,12('-')
   1,2X,'----')
    NTEST = NLKNT
    IF(NMSTAT.EQ.IOBSNM) NTEST = NOBS
    NMBR = NOBS
    IF(NMSTAT.EQ.IOBSNM) NMBR = NUSLNK
    DO 900 J=1,NTEST
    IF((NMSTAT.NE.IOBSNM).AND.(LUSEG(J).NE.1)) GO TO 900
    TT=0.
    TU=0.
    ITP=0
    ITM=0
    IT2=0
    ISIGT=4
    ISIGU=4
    ISIGP=4
    ISIGM=4
    ISIG2=4
    ISIGX=4
    CALL READER(A,B,MOE1,MOE2,IMOE1,IMOE2,IHMOE1,IHMOE2,TWIND,LUSEG,
   1 NLKNT,NOBS,J,NMSTAT,IOBSNM)
    CALL KSTEST(NMBR,A,B,ITP,ITM,IT2,ISIGP,ISIGM,ISIG2)
    IDIR=3
```

Figure 17. Sample listing for program POSTPROC (continued).

```
      INTEGER*2 IHMOE1(90,270),IHMOE2(90,270),LUSEG(270)
      IF(NMSTAT.EQ.IOBSNM) GO TO 500
      IF(TWIND-2) 100,200,300
100   DO 110 I=1,NOBS
      A(I) = MOE1(I,J)
110   B(I) = MOE2(I,J)
      RETURN
200   DO 210 I=1,NOBS
      A(I) = IMOE1(I,J) + 0.
210   B(I) = IMOE2(I,J) + 0.
      RETURN
300   DO 310 I=1,NOBS
      A(I) = IHMOE1(I,J) + 0.
310   B(I) = IHMOE2(I,J) + 0.
      RETURN
500   IF(TWIND-2) 600,700,800
600   I = 0
      DO 610 I1=1,NLKNT
      IF(LUSEG(I1).NE.1) GO TO 610
      I = I + 1
      A(I) = MOE1(J,I1)
      B(I) = MOE2(J,I1)
610   CONTINUE
      RETURN
700   I = 0
      DO 710 I1=1,NLKNT
      IF(LUSEG(I1).NE.1) GO TO 710
      I = I + 1
      A(I) = IMOE1(J,I1) + 0.
      B(I) = IMOE2(J,I1) + 0.
710   CONTINUE
      RETURN
800   I = 0
      DO 810 I1=1,NLKNT
      IF(LUSEG(I1).NE.1) GO TO 810
      I = I + 1
      A(I) = IHMOE1(J,I1) + 0.
      B(I) = IHMOE2(J,I1) + 0.
810   CONTINUE
      RETURN
      END
```

Figure 17. Sample listing for program POSTPROC (continued).

In each case, the mean and standard deviation for the MOE (for each link or observation) is calculated for each of the two traffic control patterns. The percent difference of the two means is also calculated. This is followed by a two-tailed T test and U test and a one-tailed Kolmogorov-Smirnov test, all constructed to test for statistically significant differences between the set of MOE values for the two cases. The results of these tests, along with the significance levels achieved, are printed out in tabular form for analysis.

DICTIONARY OF MAJOR FORTRAN VARIABLES

1. DATE(I) INTEGER CHARACTER

A three element vector to store the date of an observation, such as bMARb04b74bb.

2. GMEAN(I,J) REAL

The grand mean of MOE I for link J over all observations in a control pattern/time of day file.

3. IMEAN(I,J) REAL

The individual mean of MOE I for link J over all observations in a given 15-minute period in a control pattern/time of day file. The matrix is reused for successive 15-minute periods.

4. INDOUS(J) INTEGER*2

The index of data quality for link J for a specific individual data element. The vector is reused for different observations. The significance of the index values is as follows:

INDOUS(J)	MEANING
0	Data element O.K.; link to be considered as functioning multi-detector.
2	Data element O.K.; link to be considered as single-detector only.
4	Data element contains data judged to be unlikely to be correct ("BAD"); link considered multi-detector.
5	Data element was flagged in original data files as unusable ("FLAGGED"); link considered multi-detector.
6	As 4, but link considered single detector only.
7	As 5, but link considered single detector only.

5. LUSE(J) INTEGER*2

The link usability status of link J. (Applies across all observations in a case).

LUSE(J)	MEANING
0	Link J is not usable for statistical analysis.
1	Link J is usable for statistical analysis. It is to be considered a multi-detector link.
2	Link J is to be considered a single-detector link. Data elements for this link may be used in all applications for which single-detector link data is acceptable.

6. LUSEG(J) INTEGER*2

As LUE(J), but it is used as the common link usability index when more than one control pattern/time of day file is being considered simultaneously, such as in a statistical comparison. When subnetworks are being used, the J subscript may alternatively also apply to subnetworks instead of individual links.

LUSEG(J)	MEANING
0	At least one relevent LUSE(J) = 0
1	All relevant LUSE(J) = 1
2	All relevant LUSE(J) = 1 or 2 and at least one LUSE(J) = 2

7. MOE(I,J) or MOE<u>I</u>(J) INTEGER*2

The value of measure of effectiveness (MOE) number I for link J for a given observation. When MOE(J) appears with no reference to I, the value of I is implied by context and the data files being used.

I	MEASURE OF EFFECTIVENESS
1	Link I.D. = J (or a negative data flag in original data files)
2	Average hourly vehicle flow rate for the detectorized lane (VOLUME).

3	Percent lane occupancy
4	Average number of cars queued per cycle
5	Number of stops
6	Average spot speed in m.p.h.
7	Average travel time for link in seconds x 32 $(=2^5)$
8	Average vehicle delay in link in seconds.

8. NBAD INTEGER

The number of "bad" (see above for meaning) link data elements for a given observation.

9. NFLAG INTEGER

The number of "flagged" link data elements for a given observation.

10. NOBS INTEGER

The number of observations for a given control pattern/ time of day file. The presence of a subscript or suffix for the variable permits the handling of more than one such file simultaneously.

11. NSDET INTEGER

The number of usable single-detector links for a given observation.

12. OBUSE1(J) and OBUSE2(J) INTEGER*2

The observation usability index for observation number J for control pattern/time of day files no. 1 and 2, respectively. (The numbers 1 and 2 are used merely to distinguish the two cases being dealt with.). A value of 0 means that the corresponding observation should not be used. A value of 1 indicates the observation should be used. (Applies across all links).

13. TIME INTEGER CHARACTER

Contains the four-digit, 24-hour clock time for a given observation. Variable reused for successive observations.

14. TITLE(18) INTEGER

Contains the up-to-72 character title of a given control pattern/time of day file, such as "CONTROL PATTERN 1; MIDDAY".

INDIVIDUAL FORTRAN PROGRAM DOCUMENTATION

PROGRAM OBSOUT

Summary Description

This program deletes observations that are deemed unusable from an original daily MOE detector summary file, and creates a copy of the original daily file with these deletions under a different name. This program must be applied individually to each daily file in which one or more deletions are desired.

Program Limits and Variable Dimensions

The program is designed specifically to edit the standard 15-minute detector summary files described below. Each file must have no more than 50 separate observations.

Definitions of Important Variables

1. TAB(2160) INTEGER*2

A storage variable used in copying the numerical data, one observation (and record) at a time.

2. EDIT(I) INTEGER

The observation deletion index.

Technical/Quantitative Details:

None

Program Parameters and Tests

If EDIT(I)=0, observation I is retained in the new file created. Otherwise, it is deleted.

Subroutines Referenced:

None

Program Input

Permanent Files

NAME: PMMCO.UTCS.DET.MMMDD

DESCRIPTION: 15-minute MOE detector summaries for the day MMMDD

SOURCE: On disk from preliminary processing of original detector summary tapes

DISPOSITION: SHR

I/O FORMAT:
- DATE,TIME,TAB
- INTEGER DATE(3),TIME
  INTEGER*2 TAB(2160)
- No. of records = no. of observations in file

UNIT NO.: FT08F001

Temporary Files:

None

Control/Data Cards

DESCRIPTION: The observation deletion vector for a specified daily file

SOURCE: Following FT09F001 DD * card in JCL

COLUMN DESCRIPTION: Starting in column 1, the I'th column contains the digit value of EDIT(I). As many columns are used as there are observations in the daily file being edited.

UNIT NO.: FT09F001

Program Output

Permanent Files

NAME: PMMCO.UTCS.DET.MOD. MMMDD

DESCRIPTION: Copy of PMMCO.UTCS.DET.MMMDD with deletions made

LOCATION: Disk

DISPOSITION: NEW,KEEP

I/O FORMAT: Just as PMMCO.UTCS.DET.MMMDD, but fewer records

UNIT NO.: FT10F001

Temporary Files

None

Printer, Error Messages

(See Figure 18. for a sample printer output)

DESCRIPTION:	The message: "OBSERVATION [DATE TIME] DELETED" is printed out for every record deleted.
ERROR MESSAGE:	The message "MORE THAN 50 RECORDS IN DATA SET" printed out if the file end has not been reached after reading 50 records. The program then terminates prematurely.
UNIT NO.:	FT06F001

JCL to Run Program

For each run:

Use the appropriate MMMDD wherever the suffix should appear.

Use the appropriate series of zeros (or blanks) and nonzeros on the data card to delete the desired observations from the MMMDD file.

OBSERVATION	APR 26 1974	0745	DELETED
OBSERVATION	MAY 20 1974	1330	DELETED
OBSERVATION	MAY 28 1974	1345	DELETED
OBSERVATION	MAY 30 1974	0915	DELETED
OBSERVATION	MAY 31 1974	0715	DELETED

Figure 18. Sample printout from run of program OBSOUT.

PROGRAM CONFIL3

Summary Description

This program performs four tasks:

It first trisects and then consolidates all of the daily summary files for a given control pattern into three time-of-day files: a.m., midday, and p.m.

It tests the individual data elements and assigns them data quality/usability index values, which are saved.

It calculates the number of links that are (a) bad; (b) flagged; and (c) usable, but single-detector for each observation and saves them.

It deletes observations which exceed specified maximum numbers of either bad or bad + flagged links and prints a message to that effect. Because data can be removed only entire row or column (link across all observations or observation across all links) at a time, deleting "spotty" observations helps keep the number of links that cannot be used for analysis down to a minimum.

Program Limits and Variable Dimensions

CONFIL3 can be applied in its present form only to the standard format daily detector summary files with data for eight MOE's for 270 links. The value of DATE(1) must always be a blank followed by the standard three-letter abbreviation of one of the 12 months.

Definitions of Important Variables

1. NDET(J) INTEGER*2

The number of detectors <u>reported</u> operating on link J.

2. INDPER INTEGER

Equals 1, 2, or 3 depending on whether we are dealing with the a.m., midday, or p.m. periods, respectively.

3. TEST(J) INTEGER

Criterion values used to determine data quality status and whether an observation should be deleted:

TEST(1)	The minimum reasonable lane volume [[8]]
TEST(2)	The maximum reasonable average spot speed [[50]]
TEST(3)	The maximum reasonable link travel time in seconds (x32) [[6000]]
TEST(4)	The maximum reasonable lane volume [[1500]]
TEST(5)	The number of bad links per observation which requires that the observation be deleted [[5]]
TEST(6)	The number of bad + flagged links per observation which requires that the observation be deleted [[75]]

<u>Technical/Quantitative Details</u>

None

<u>Program Parameters and Tests</u>

1. 0945 is the latest a.m. time

   1600 is the earliest p.m. time

2. The program assumes that there is a data file misalignment or similar I/O problem if DATE(1) of any observation does not equal to "bMMM" where MMM is the standard three-letter abbreviation of a month.

3. A link's data element for an observation is classified single-detector if:

   (a) NDET(J) $\leq$ 1, or

   (b) MOE(4,J), MOE(5,J), MOE(8,J) all = 0

   (Since these values are typically zero for single-detector links, for which they cannot be calculated).

4. A link's data element is classified "bad" if:

   (a) MOE(2,J) $<$ TEST(1), or

   (b) MOE(6,J) $>$ TEST(2), or

   (c) MOE(7,J) $>$ TEST(3), or

(d) MOE(I,J) < 0   I = 3,.....,8, or

(e) MOE(2,J) > TEST(4), or

(f) MOE(1,J) ≠ J <u>and</u> MOE(1,J) > 0

5. A link's data element is classified "flagged" if:

   MOE(1,J) ≠ J <u>and</u> MOE(1,J) < 0

6. An observation is deleted if:

   (a) NBAD ≥ TEST(5), or

   (b) NBAD + NFLAG ≥ TEST(6)

<u>Subroutines Referenced</u>

None

<u>Program Input</u>

<u>Permanent Files</u>

NAME:	PMMCO.UTCS.DET.<u>MMMDD</u>* (* all MMMDD for a given control pattern concatenated in chronological order for the unit number listed below. For all files which had observations deleted by program OBSOUT, PMMCO.UTCS.DET.MOD.<u>MMMDD</u> should be used instead.)
DESCRIPTION:	The daily 15-minute MOE detector summary files for all days under the given control pattern, using the output files of OBSOUT instead of the original daily files where necessary.
SOURCE:	Disk files
DISPOSITION:	SHR
I/O FORMAT:	See description under program OBSOUT
UNIT NO.:	FT11F001

<u>Temporary Files</u>

None

Control/Data Cards

DESCRIPTION:	Vector giving the number of detectors that were reported being on each link.
SOURCE:	Following the FT09F001 DD * card in the JCL
COLUMN DESCRIPTION:	Starting in column 1 of each card and continuing to column 80 before continuing on the next card, the single-digit integer representing the number of detectors on link J appears in the J'th column.
UNIT NO.:	FT09F001
DESCRIPTION:	The criterion values, TEST(I).
SOURCE:	Following the FT10F001 DD * card in the JCL.

COLUMN DESCRIPTION:

VALUE	COLUMNS
TEST(1)	1-10
TEST(2)	11-20
TEST(3)	21-30
TEST(4)	31-40
TEST(5)	41-50
TEST(6)	51-60

UNIT NO.: FT10F001

Program Output

Permanent Files

NAME:	PMMCO.UTCS.TEMPAMY, PMMCO.UTCS. TEMPMDY, and PMMCO. UTCS.TEMPPMY, respectively.
DESCRIPTION:	The a.m., midday, and p.m. time-of-day files for control pattern number Y, respecitvely.
LOCATION:	Disk
DISPOSITION:	NEW, KEEP

I/O FORMAT: - DATE,TIME,NBAD,NFLAG,NSDET ((INDOUS(J),MOE(I.J),I=1,8),J=1,270))
- INTEGER DATE(3),TIME,NBAD,NFLAG,NSDET INTEGER*2 INDOUS(270),MOE(8,270)
- No. of Records = no. of observations retained as usable for control pattern/ time of day of file.

UNIT NO.s: FT12F001, FT13F001, FT14F001, respectively.

Temporary Files

None

Printer, Error Messages

(See Figure 19.)

DESCRIPTION: A listing of all observations deleted by the program from its output files, giving the date and time of the observation as well as the values of NBAD, NFLAG, and NSDET.

ERROR MESSAGES: If any of the DATE(1) values is not one of the 12 permitted values, the job terminates prematurely after the following message is printed: "THE START OF A 15-MINUTE FILE IS INCORRECT OR MISPLACED:" followed by what should be the date and time of the observation and the first link worth of MOE values.

UNIT NO.: FT06F001

JCL to Run Program

To make a run for any particular control pattern, the Y suffix must be the appropriate number in all file names in which it appears and the concatenated set of daily summary files must be made those falling in the period of the appropriate control pattern.

```
THE 15 MINUTE INTERVAL  MAR 05 74   1830  WAS DELETED BECAUSE IT HAD NBAD = 91;  NFLAG = 40;  NSDET = 71

THE 15 MINUTE INTERVAL  MAR 12 74   0715  WAS DELETED BECAUSE IT HAD NBAD =144;  NSDET = 30;  NSDET = 41

THE 15 MINUTE INTERVAL  MAR 12 74   0730  WAS DELETED BECAUSE IT HAD NBAD = 53;  NFLAG = 18;  NSDET = 91
```

(This is a typed copy)

Figure 19. Sample printout from run of program CONFIL3.

PROGRAM OLOUT

Summary Description

This program:

1. Counts and stores the number of individual data elements by data quality/usability level for each link over all observations for the control pattern and time of day under consideration.;

2. Makes the final deletion of observations based upon a maximum number of links in any given observation which can have a data quality index indicating a single-detector link, while the nature of the link data over all observations indicates that the link is functioning predominantly as a multiple-detector link. (A large number of such links probably indicates a serious detector problem.);

3. Counts and saves the number of remaining usable observations in the input file.;

4. Constructs a link usability vector which will specify the level of usability of each link for this control pattern and time of day.

Program Limits and Variable Dimensions

The program is made specifically to handle the standard case of eight MOE's and 270 links.

Definitions of Important Variables

1. INDG(I,J) INTEGER*2

The number of individual data elements over all observations recorded for link I falling in the I'th data quality/usability class:

I	DATA QUALITY/USABILITY LEVEL
1	INDOUS(J) = 0
2	INDOUS(J) = 2
3	INDOUS(J) = 4
4	INDOUS(J) = 5
5	INDOUS(J) = 6
6	INDOUS(J) = 7

The maximum attainable value of INDG(I,J) is the number of observations for this case.

2. TEST2(I) INTEGER

Two test criterion values:

TEST2(1) The number of links in an observation which have INDOUS(J) = 2 yet should be multi-detector links, that requires that the observation be deleted. [[20]]

TEST2(2) The maximum number of "bad" data elements for a link in an observation that can be tolerated before designating the link not usable. [[5]]

3. FRTST(I) REAL

Three fractional criterion test values:

FRTST(1) The maximum ratio of INDG(1,J) to (INDG(1,J) + INDG(2,J)) for a link before that link is considered to be "predominantly" multiple-detector. [[0.9]]

FRTST(2) The minimum fraction of observations for a link that must have a 0 or 2 quality status (either multi-detector or single-detector) in order for the link to be considered usable for analysis purposes. [[0.6]]

FRTST(3) The minimum ratio of INDG(1,J) to (INDG (1,J) + INDG(2,J)) before a link is classified as being single-detector for analysis purposes. [[0.9]]

4. NSMLT INTEGER

The number of data elements across all links in an observation which have a single-detector status although their associated links are predominantly multi-detector for the file as a whole.

Technical/Quantitative Details

None

Program Parameters and Tests

1. For a given file, a link is considered to be predominantly multi-detector if:

$$\frac{\mathrm{INDG(1,J)}}{\mathrm{INDG(1,J)} + \mathrm{INDG(2,J)}} > \mathrm{FRTST(1)}$$

2. For a given file, an observation is deleted if the number of links in the observation which had INDOUS(J) = 2, despite the fact that the link is considered predominantly multi-detector, is greater than TEST2(1).

3. LUSE(J) is set = 0 if:

(a) $\frac{\mathrm{INDG(1,J)} + \mathrm{INDG(2,J)}}{\mathrm{NOBS}} < \mathrm{FRTST(2)}$, or

(b) $\mathrm{INDG(3,J)} + \mathrm{INDG(5,J)} > \mathrm{TEST2(2)}$

4. LUSE(J) is set = 2 if:

$$\frac{\mathrm{INDG(1,J)}}{\mathrm{INDG(1,J)} + \mathrm{INDG(2,J)}} < \mathrm{FRTST(3)}$$

5. If the above two sets of tests fail, LUSE(J) is set = 1.

Subroutines Referenced

None

Program Input

Permanent Files

NAME: PMMCO.UTCS.TEMPXXY

SOURCE: Disk, CONFIL3 output file

DISPOSITION: SHR

UNIT NO.: FT10F001

Temporary Files

None

Control/Data Cards

DESCRIPTION: Two integer and three real criterion values for tests in the program.

SOURCE: Following the FT09F001 DD * card in the JCL

COLUMN DESCRIPTION:	VALUE	COLUMNS	
	TEST2(1)	1-10	2I10
	TEST2(2)	11-20	
	FRTST(1)	21-30	3F10.5
	FRTST(2)	31-40	
	FRTST(3)	41-50	

UNIT NO.: FT09F001

Program Output

Permanent Files

NAME: PMMCO.UTCS.NOBS(XXY) (PDS)

DESCRIPTION: The number of observations in the time of day XX for control pattern Y.

LOCATION: Disk

DISPOSITION: NEW,KEEP the first time the PDS is referenced and OLD in subsequent runs when new members are being added

I/O FORMAT:
- NOBS
- FORMAT(I10)
- 1 record

UNIT NO.: FT12F001

NAME: PMMCO.UTCS.INDGXXY

DESCRIPTION: Counts for all links by data quality/ usability level across all observations in the given time of day and control pattern case.

LOCATION: Disk

DISPOSITION: NEW,KEEP

I/O FORMAT:
- ((INDG(I,J),I=1,6,J=1,270)
- INTEGER*2 INDG(6,270)
- 1 record

UNIT NO.: FT13F001

NAME: PMMCO.UTCS.LUSEXXY

DESCRIPTION: The link usability vector for the XXY case

LOCATION: Disk

DISPOSITION: NEW,KEEP

I/O FORMAT:
- (LUSE(J),J=1,270)
- INTEGER*2 LUSE(270)
- 1 record

UNIT NO.: FT14F001

Temporary Files

NAME: &&TEMP1

DESCRIPTION: The version of the 15-minute detector summary data file as edited and expanded by this program.

LOCATION: Disk(scratch)

DISPOSITION: NEW,PASS

I/O FORMAT:
- DATE,TIME,NBAD,NFLAG,NSDET,((INDOUS(J), MOE(I,J),I=1,8)),J=1,270)
- INTEGER DATE(3),TIME,NBAD,NFLAG,NSDET
  INTEGER*2 INDOUS(270),MOE(8,270)
- No. of records - no. of observations

UNIT NO.: FT11F001

Printer, Error Messages

DESCRIPTION: All observations deleted by this program are printed out, as well as the corresponding values of NSMLT.

ERROR MESSAGES: If an invalid value of INDOU(J) is detected at either of two points in the program, an error message to that effect is printed out, including the illegal value and the observation involved. The program terminates prematurely.

UNIT NO.: FT06F001

JCL to Run Program

For a run, all XXY's in file names in the JCL must be the one appropriate for the time of day and control pattern involved.

FIRST FILE SORT

Following the execution of program OLOUT in Step 3 (See the OVERALL COMPUTER PROGRAM LINKAGE DESCRIPTION CHAPTER), the IBM SORT/MERGE utility package is applied to sort the &&TEMP1 file generated by OLOUT and itself output the sorted file as the temporary data set &&TEMP2.

This sort is of observations in a file, so that they appear in ascending order by 24-hour clock time, with all observations having the same clock time adjacent in the sorted file. See the JCL for Step 3, starting with the //SORT1 EXEC PGM=SORT,... card for the proper specification of this sort.

PROGRAM PLUGNET

Summary Description

This program performs the following tasks:

1. For each link and MOE, it calculates the mean value of the MOE over (a) all usable observations for the case under consideration--the "grand means" and (b) all usable observations for the case for each 15-minute period clock time separately--the "individual means". All the means are saved in files.

2. The program then uses these means to substitute for (or "plug") individual data elements that have been identified as unusable, but which correspond to a link and an observation which are both considered usable for analysis in this case. The means used to plug values always correspond to the same MOE and link as the data element involved. Also, unless the appropriate individual mean could not be calculated because of lack of data, the mean used also corresponds to the same 15-minute period as the data element being plugged.

3. The program finally calculated three "network" quantities--actually MOE's aggregated over all usable links--for each observation in the file separately. These network quantities are: total volume, total vehicle seconds of delay (VOLUME*DELAY), and total average number of cars queued per cycle. The latter two measures were aggregated

only over usable multi-detector links since the MOE's in these two cases are not computed for single-detector links. These values are added to the modified summary detector file outputed by this program.

Program Limits and Variable Dimensions

The program works with the standard summary MOE file dimensions of eight MOE's for 270 links. The dimensions of all related variables are based on this standard configuration.

Definitions of Important Variables

1. NUS(J) INTEGER

The number of data elements for a given 15-minute period for link J which was deemed usable (either single- or multi-detector levels of usefulness). The variable is reused for the different 15-minute periods.

2. NTUS(J) INTEGER

The number of data elements for a given 15-minute period for link J which was deemed usable and has multi-detector levels of usefulness (can be used for all seven performance MOE's).

3. NGUS(J) INTEGER

The number of data elements for the entire control pattern/time of day case for link J which was deemed usable (either single- or multi-detector levels).

4. NTGUS(J) INTEGER

The number of data elements for the case for link J which was deemed usable at the multi-detector level.

5. TIMLST INTEGER CHARACTER

The value of TIME for the last observation read in by the program.

6. NETVOL INTEGER

The sum of link volumes over all usable links for a given observation.

7. NETDEL INTEGER

The sum of vehicle seconds of delay over all links usable at the multi-detector level for a given observation.

8. NETQUE INTEGER

The sum of average number of vehicles queued by cycle over all links usable at the multi-detector level for a given observation.

Technical/Quantitative Details

There are two different types of MOE's. While queues per cycle (MOE4), stops (MOE5), and delay (MOE8) are only calculated for multi-detector links, the other MOE's are calculated for all links [however, it should be noted that travel time (MOE7) is based in part on delay and thus the figure available in the data file for this MOE will be misleading for single-detector links].

Thus, the calculation of means must be done differently for the two different types of MOE. The number of links summed over to obtain the mean is smaller for the first three MOE's than it is for the others.

The flag value of -1. is assigned to a mean whenever the value cannot be calculated because of a lack of appropriate data. This flagging permits the subsequent plugging process to choose the correct mean (individual or grand) to use.

Program Parameters and Tests

All individual data elements which correspond to a usable link and which have INDOUS(J) > 2 are plugged with appropriate means.

Subroutines Referenced

None

Program Input

Permanent Files

NAME: PMMCO.UTCS.LUSEXXY

SOURCE: Disk, see program OLOUT output

DISPOSITION: SHR

UNIT NO.: FT09F001

NAME: PMMCO.UTCS.NOBS(XXY)

SOURCE: Disk, see program OLOUT output

DISPOSITION: SHR

UNIT NO.: FT11F001

Temporary Files

NAME: &&TEMP2

SOURCE: Disk, see SORT1 utility output

DISPOSITION: OLD,DELETE

UNIT NO.: FT13F001

Control/Data Cards

DESCRIPTION: Title of case being processed, including control pattern number and time of day in character form.

SOURCE: Following the FT12F001 DD * card in the JCL

COLUMN DESCRIPTION: Starting in column 2: "CONTROL PATTERN Y: XXXX where Y is the control pattern number and XXXX = "a.m.", "midday", or "p.m."

UNIT NO.: FT12F001

Program Output

Permanent Files

NAME: PMMCO.UTCS.IBARXXY

DESCRIPTION: The individual means for all MOE's, links, and 15-minute time periods.

LOCATION: Disk

DISPOSITION: NEW,KEEP

I/O FORMAT: - ((MEAN(I,J),I=1,7)J=1,270))
- REAL IMEAN(7,270)
- No. of records = no. of different 15-minute clock times that occur in the detector summary input file.

UNIT NO.: FT14F001

NAME: PMMCO.UTCS.GBARXXY

DESCRIPTION: The grand means for all MOE's and links for this case.

LOCATION: Disk

DISPOSITION: NEW,KEEP

I/O FORMAT: - ((GMEAN(I,J),I=1,7),J=1,270)
- REAL GMEAN(7,270)
- 1 record

UNIT NO.: FT15F001

Temporary Files

NAME: &&TEMP3

DESCRIPTION: Latest version of the detector summary data file for the current case with the data plugs and network quantities added.

LOCATION: Disk

DISPOSITION: NEW,PASS

I/O FORMAT: - DATA,TIME,NBAD,NFLAG,NSDET,NETVOL, NETDEL,NETQUE, ((INDOUS(J),(MOE(I,J), I=1,8)),J=1,270)
- INTEGER DATE(3),TIME through NETQUE INTEGER*2 INDOUS (270),MOE(8,270)
- No. of records = no. of observations

UNIT NO.: FT16F001

Printer, Error Messages

None

JCL to Run Program

To make a run for case XXY, you must change all file suffixes of the type XXY to the appropriate specific case descriptor, and the control pattern number and time of day in the title card following the FT12F001 DD * card must be changed to those appropriate.

PROGRAM SCTRGM

Summary Description

This program produces a scattergram for data in a given control pattern/time of day case file. The horizontal axis is lane hourly vehicle flow rate (volume) in units of eight vehicles per hour. The vertical axis is average link delay in units of three seconds. The numbers plotted represent ranges of frequency of the occurrence of individual data element (a given link during a given observation) which had a volume and delay falling within the ranges of the corresponding box along the two axes.

The resulting scattergrams permit a rapid examination of the relationship between volume and delay for a given case and the change, if any, in this pattern among different time of day/control pattern cases. The fixed units of the scattergram facilitates comparison of scattergrams for different cases.

Program Limits and Variable Dimensions

As currently written, the scattergram can only be of volume versus delay. The maximum displayable volume is 8x130=1040 vehicles per hour and the maximum displayable delay is 3x60=180 seconds. All greater values are displayed as if they were these maximum values. The program is designed to work with the standard configuration of eight MOE's and 270 links in the summary file. Ten frequency ranges are permitted to be distinguished. This includes one for the frequency of 0--nonoccurrence.

Definitions of Important Variables

1. MAP(I,J)　　　INTEGER*2　　　(ULTIMATELY)CHARACTER

The number of individual data elements having a delay falling in the I'th interval and a volume in the (J-1)st (or the character representing the range of frequencies into which this number falls).

2. MAPKEY(I) INTEGER*2

The highest frequency for the (I+1)st frequency range. The first frequency range is simply 0.

3. KEY(I) INTEGER*2

The characters used to represent the various frequency ranges on the scattergram. KEY(I) is the character used to represent the I'th range.

Technical/Quantitative Details

None

Program Parameters and Tests

The frequency ranges are as follows:

Range	Frequencies	Scattergram Character
1	0	BLANK
2	1	1
3	2-3	2
4	4-5	3
5	6-10	4
6	11-20	5
7	21-30	6
8	31-50	7
9	51-75	8
10	$>$ 75	9

Subroutines Referenced

None

Program Input

Permanent Files

NAME: PMMCO.UTCS.NOBS(XXY)

SOURCE: Disk, OLOUT program output

DISPOSITION: SHR

UNIT NO.: FT10F001

NAME: PMMCO.UTCS.LUSEXXY

SOURCE: Disk,OLOUT program output

DISPOSITION: SHR

UNIT NO.: FT11F001

Temporary Files

NAME: &&TEMP3

SOURCE: Disk, PLUGNET program output

DISPOSITION: OLD,PASS

UNIT NO.: FT12F001

Control/Data Cards

DESCRIPTION: Control pattern/time of day title card

SOURCE: Following FT09F001 DD * card in the JCL

COLUMN DESCRIPTION: Just like that following the FT12F001 DD * card in the PLUGNET JCL

UNIT NO.: FT09F001

Program Output

Permanent Files

None

Temporary Files

None

Printer, Error Messages

(See Figure 20.)

DESCRIPTION: The scattergram described above with the appropriate case title should cover one full computer page. Asterisks serve to delineate axes.

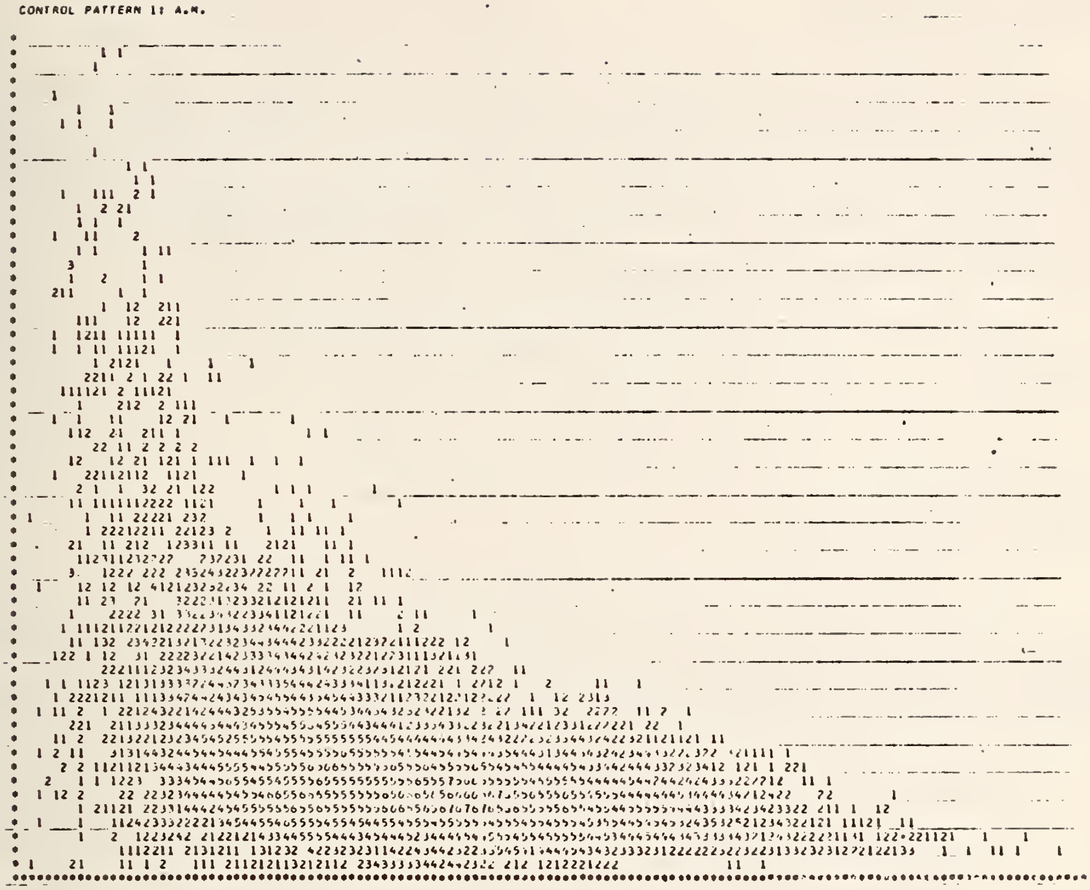

Figure 20. Sample printout from run of program SCTRGM.

SECOND FILE SORT

This IBM utility sort is performed on the latest version of the summary detector file for the case under consideration, &&TEMP3. This sort reorders the observations so that they are in ascending order by network volume, within the primary sort on clock time in ascending order. See the Step 3 JCL starting with the //SORT2 EXEC card for the exact parameters to be used.

The output of this sort is a new temporary summary file, &&TEMP4.

PROGRAM BRKFIL

Summary Description

This program merely disaggregates the final version of the summary detector file, &&TEMP4, into nine smaller permanent files containing the same data when considered together. This disaggregation is for the purpose of making subsequent data manipulation more efficient.

Separate files are created for:

1. The observation level data;
2. The INDOUS(J) vectors for all observations;
3. Each of the seven sets of performance MOE's for all links and observations.

Program Limits and Variable Dimensions

The program is designed for the standard &&TEMP4 file configuration, with 270 links and eight MOE's (seven of which are usable performance measures).

Definitions of Important Variables

1. K INTEGER

The observation sequence number.

K=1,2,...,NOBS

Technical/Quantitative Details

None

Program Parameters and Tests

None

Subroutines Referenced

None

Program Input

Permanent Files

NAME:	PMMCO.UTCS.NOBS(XXY)
SOURCE:	Disk, OLOUT program output
DISPOSITION:	SHR
UNIT NO.:	FT09F001

Temporary Files

NAME:	&&TEMP4
DESCRIPTION:	Latest version of the summary MOE detector data for the case under consideration
SOURCE:	Disk, SORT2 output
DISPOSITION:	SHR
UNIT NO.:	FT10F001

Control/Data Cards

None

Program Output

Permanent Files

NAME:	PMMCO.UTCS.HEADXXY
DESCRIPTION:	The observation level data from the summary file.
LOCATION:	Disk
DISPOSITION:	NEW,KEEP
I/0 FORMAT:	- K,DATE,TIME,NBAD,NFLAG,NSDET,NETVOL, NETDEL,NETQUE - INTEGER K, DATE(3), TIME through NETQUE - No. of records = no. of observations

UNIT NO.: FT11F001

NAME: PMMCO.UTCS.INDSXXY

DESCRIPTION: The values of INDOUS(J) for all J for each observation in the case.

LOCATION: Disk

DISPOSITION: NEW,KEEP

I/O FORMAT:
- K,(INDOUS(J),J=1,270)
- INTEGER K
  INTEGER*2 INDOUS(270)
- No. of records = no. of observations

UNIT NO.: FT12F001

NAME: PMMCO.UTCS.MOEIXXY I = 2,3,...,8

DESCRIPTION: All 270 link values of MOEI for all observations in the case. (Each MOE assigned to a different file--see unit numbers below.)

LOCATION: Disk

DISPOSITION: NEW,KEEP

I/O FORMAT:
- K,(MOE(I,J),J=1,270)
- INTEGER K
  INTEGER*2 MOE(8,270)
- No. of records = No. of observations

UNIT NO.: FT13F001, FT14F001,...FT19F001, respectively

Temporary Files

None

Printer, Error Messages

None

JCL to Run Program

To make a run, make sure that all XXY-type file suffixes are correct.

PROGRAM ENDPRNT

Summary Description

This program produces a printed data summary and detailed data sampling of the final version of the data files. The report produced includes, for the time of day/control pattern case involved:

1. the link usability vector, LUSE(J);

2. a listing, in sorted order, of all observations and the observation level values associated with them: number of bad, flagged, and usable single-detector link values; network volume; network vehicle minutes of delay; and network average queues per cycle.

3. a listing of all link values for the MOE's, (a) volume and (b) delay for all observations.

Program Limits and Variable Dimensions

The program is designed to report on the standard configuration of files with 270 links and seven performance MOE's as defined by the Step 3 programs and JCL.

Definitions of Important Variables

No new variables of importance introduced in this program.

Technical/Quantitative Details

None

Program Parameters and Tests

None

Subroutines Referenced

None

Program Input

Permanent Files

NAME: PMMCO.UTCS.NOBS(XXY)

SOURCE: Disk, OLOUT program output

DISPOSITION: SHR

UNIT NO.: FT09F001

NAME: PMMCO.UTCS.LUSEXXY

SOURCE: Disk, OLOUT program output

DISPOSITION: SHR

UNIT NO.: FT10F001

NAME: PMMCO.UTCS.HEADXXY

SOURCE: Disk, BRKFIL program output

DISPOSITION: SHR

UNIT NO.: FT11F001

NAME: PMMCO.UTCS.MOEIXXY I=2,8

SOURCE: Disk, BRKFIL program output

DISPOSITION: SHR

UNIT NOs.: FT12F001 and FT13F001, respectively

Temporary Files

None

Control/Data Cards

None

Program Output

Permanent Files

None

Temporary Files

None

Printer, Error Messages

(See Figure 21.)

DESCRIPTION: First, the LUSE(J) vector is printed out, 72 single-digit unspaced values per line for a total of 270 values. Next, the observations are listed in sorted order, together with the observation level data appearing under column headings identifying each value.

Finally, for each observation, all link values for MOE2 (volume) and then MOE8 (delay) are printed out in two closely spaced blocks following a marginal indication of the observation involved. Each block contains 270 values, one for each link, including good, plugged, and even bad unplugged values.

JCL to Run Program

To run this program, make sure that the XXY suffixes to all file names in the JCL are correct for the case under consideration.

PROGRAM NETCON

Summary Description

Starting with this program, all runs will be made with respect to a comparison of two cases and will thus refer to files from two different control patterns. NETCON is the first in a chain of steps that paves the way for a meaningful and consistent statistical comparison of two cases for which the chain was run. The actual statistical comparison will occur in a subsequent step, that of the post-processor.

Although previous programs have defined for each case separately a link usability vector and have calculated network volumes based on these vectors, a statistical comparison of two cases requires a single common link usability vector and a "matching" of observations based on two consistent sets of network volumes (network being defined as the same set of usable links in both cases). NETCON provides a single link usability vector and two consistent sets of network volumes for the two control pattern/time of day cases for which the program is run.

```
LUSE(J)
20121011211220222221122111112111111011111111111111222222222222211211101021
11101201202002211122201201120201122111112102222272221121212222012112222222
01121212112211111211222121021110122220111112111021121112121222111111212111
1122222121222222222211211122222221121022222022222222022
```

LI	K	DATE	TIME	NBAD	NFLAG	NSDET	NETVOL	NETDEL	NETQUE
1	1	MAR 14 74	0715	1	31	118	56055	658912	432
2	2	MAR 15 74	0715	3	27	122	56427	628060	409
3	3	MAR 13 74	0715	3	25	122	56640	675197	434
4	4	MAR 06 74	0715	1	22	126	57000	629756	423
5	5	MAR 05 74	0715	3	22	127	57056	641192	429
6	6	MAR 07 74	0715	2	17	134	59477	652378	418
7	7	MAR 07 74	0730	1	24	127	66237	672372	454
8	8	MAR 13 74	0730	0	25	123	67317	698180	451
9	9	MAR 05 74	0730	1	25	124	67690	682516	455
10	10	MAR 15 74	0730	1	27	122	73781	663251	460
11	11	MAR 14 74	0730	0	30	119	74745	669536	446
12	12	MAR 06 74	0730	0	24	124	74965	679048	448
13	13	MAR 05 74	0745	0	26	123	78490	755656	504
14	14	MAR 07 74	0745	0	24	127	79836	729524	500
15	15	MAR 14 74	0745	1	30	119	81142	739342	493
16	16	MAR 06 74	0745	0	24	124	81212	735708	496
17	17	MAR 15 74	0745	0	26	123	81992	734280	501
18	18	MAR 12 74	0745	1	18	127	82472	705078	480
19	19	MAR 13 74	0745	0	25	123	86356	731288	504
20	20	MAR 07 74	0800	0	24	127	88896	830248	560
21	21	MAR 14 74	0800	0	26	123	89904	790772	535
22	22	MAR 13 74	0800	0	25	123	90560	779620	531
23	23	MAR 06 74	0800	0	24	124	91864	848760	555
24	24	MAR 12 74	0800	0	19	126	92163	805388	555
25	25	MAR 15 74	0800	0	26	123	92408	808072	552
26	26	MAR 05 74	0800	0	25	124	94408	815380	540
27	27	MAR 13 74	0815	0	25	123	95520	910429	586
28	28	MAR 06 74	0815	0	24	124	95564	655872	559
29	29	MAR 05 74	0815	0	27	122	96736	868060	575
30	30	MAR 15 74	0815	0	26	123	97478	833598	579
31	31	MAR 12 74	0815	0	18	127	97565	913489	595
32	32	MAR 14 74	0815	0	26	124	98742	932766	608
33	33	MAR 07 74	0815	0	24	127	105384	920830	612
34	34	MAR 12 74	0830	0	18	127	96804	938512	616
35	35	MAR 05 74	0830	0	27	122	98725	969728	596
36	36	MAR 13 74	0830	0	26	123	100070	957378	621
37	37	MAR 07 74	0830	0	24	127	100296	990700	643
38	38	MAR 06 74	0830	0	24	124	107392	939792	608
39	39	MAR 14 74	0830	0	26	123	109242	961866	618
40	40	MAR 15 74	0830	0	26	123	109606	936632	623
41	41	MAR 05 74	0845	0	26	123	98024	960784	622
42	42	MAR 14 74	0845	0	26	127	100321	1024140	577
43	43	MAR 07 74	0845	0	24	128	100496	1039268	623
44	44	MAR 06 74	0845	0	24	124	101004	1041844	634
45	45	MAR 15 74	0845	0	26	123	102105	1036364	651
46	46	MAR 12 74	0845	0	18	127	107952	1007072	642
47	47	APR 08 74	0845	0	25	124	107161	953270	624
48	48	MAR 13 74	0845	0	26	123	107753	1054248	662
49	49	MAR 06 74	0900	1	24	124	96356	1075708	610
50	50	APR 08 74	0900	0	26	123	96682	1102346	635
51	51	MAR 12 74	0900	1	16	127	96790	1052512	656
52	52	MAR 07 74	0900	0	24	127	97360	1097736	638
53	53	MAR 15 74	0900	0	26	124	97781	973274	622
54	54	MAR 14 74	0900	0	27	125	98809	1064250	638
55	55	MAR 13 74	0900	0	26	123	99473	1068650	666
56	56	MAR 05 74	0900	2	25	122	104092	976378	627

Figure 21. Sample printout from run of program BRKFIL.

The single link usability vector is defined as the most restrictive value of the two individual link usability vector elements, for each link independently. Optionally, a third, restraining link usability vector may be read in exogenously for the purpose of "manually" deleting links from later analysis because of external knowledge on detector reliability or for any other purpose.

Network volumes for both cases over all observations are then recalculated based on the single "common" link usability vector just derived. These recalculated network volume values will now be strictly comparable since they will represent volume aggregated over the same links.

Program Limits and Variable Dimensions

This program was intended to apply to the standard 270 link case with the volume MOE file in the form as output by program BRKFIL.

Definitions of Important Variables

1. LUSEM(J) INTEGER*2

= 0 if link J is to be considered unusable for this two-case comparison, independent of the two individual LUSE(J) values.

= 1 if the usability of link J is to be determined by the two individual LUSE(J) values only.

2. ILUSEM INTEGER

= 1 if an exogenous LUSEM(J) vector (with at least one "0" value) is to be read in.

3. The appearance of "1" and "2" following variables such as LUSE and NOBS is merely to distinguish between values for the first and second cases read in for the run.

Technical/Quantitative Details

None

Program Parameters and Tests

The exogenous LUSEM(J) vector is read in if the value of ILUSEM read in is 1.

Subroutines Referenced

None

Program Input

Permanent Files

NAME:	PMMCO.UTCS.NOBS(XXY), PMMCO.UTCS.NOBS(XXZ)
SOURCE:	Disk, OLOUT program output
DISPOSITION:	SHR
UNIT NO.'s:	FT10F001 and FT11F001, respectively
NAME:	PMMCO.UTCS.LUSEXXY, PMMCO.UTCS.LUSEXXZ
SOURCE:	Disk, OLOUT program output
DISPOSITION:	SHR
UNIT NO.'s:	FT12F001 and FT13F001, respectively
NAME:	PMMCO.UTCS.HEADXXY with PMMCO.UTCS.HEADXXZ concatenated to it.
SOURCE:	Disk, BRKFIL program output
DISPOSITION:	SHR
UNIT NO.:	FT16F001
NAME:	PMMCO.UTCS.MOE2XXY with PMMCO.UTCS.MOE2XXZ concatenated to it.
SOURCE:	Disk, BRKFIL program output
DISPOSITION:	SHR
UNIT NO.:	FT17F001

Temporary Files

None

Control/Data Cards

DESCRIPTION:	Switch to indicate specification of exogenous link usability vector.
SOURCE:	Following the FT05F001 DD * Card.
NUMBER:	One card.
CONTENT:	ILUSEM
FORMAT:	(I10)
DESCRIPTION:	Optional exogenous link usability vector
SOURCE:	Concatenated to the above card.
NUMBER:	0; if ILUSEM = 0 4; if ILUSEM = 1
CONTENT:	LUSEM (J): Card 1: J=1, 72 Card 2: J=73, 144 Card 3: J=145, 216 Card 4: J=217, 270

Note: All 270 values must be supplied.

FORMAT:	(72I1)

Program Output

Permanent Files

NAME:	PMMCO.UTCS.LUSGXXYZ
DESCRIPTION:	The single link usability vector to be used in comparisons of case XXY with case XXZ
LOCATION:	Disk
DISPOSITION:	NEW,KEEP
I/O FORMAT:	- (LUSEG(J), J=1,270) - INTEGER*2 LUSEG(270) - 1 record
UNIT NO.:	FT15F001

NAME:	PMMCO.UTCS.VOLXXYZ, PMMCO.UTCS.VOLXXZY
DESCRIPTION:	The re-calculated consistent network volume files for cases XXY and XXZ, respectively for comparison purposes.
LOCATION:	Disk
DISPOSITION:	NEW,KEEP
I/O FORMAT:	- K,DATE, TIME,NETVOL - INTEGER K, DATE(3),TIME,NETVOL - No. of records = No. of observations
UNIT NO.'s:	FT18F001 and FT19F001, respectively

Temporary Files

None

Printer, Error Messages

(See Figure 22.)

DESCRIPTION: First, the single-link usability vector is printed out, 100 elements to a line for 270 link values. This is followed by a listing in tabular form of all observations for the first of the two cases of the comparison inputed. Along with each observation's date and time is the recalculated network volume. This tabular listing is followed by a similar one for the second case of the comparson.

JCL to Run Program

To make a run of this program, the JCL:

1. Must have the correct value of ILUSEM on the control card following the FT05F001 DD * card.

2. If ILUSEM = 1, the cards specifying the LUSEM(J) vector must be correctly inserted following the above card.

3. All XXY, XXZ, XXYZ, and XXZY suffixes of the file names must be correct for the given run.

K	DATE	TIME	NETVOL
1	MAR 14 74	0715	48222
2	MAR 15 74	0715	48620
3	MAR 13 74	0715	48937
4	MAR 06 74	0715	49324
5	MAR 05 74	0715	49489
6	MAR 07 74	0715	51484
7	MAR 07 74	0730	57211
8	MAR 13 74	0730	58249
9	MAR 05 74	0730	58424
10	MAR 15 74	0730	63692
11	MAR 14 74	0730	64617
12	MAR 06 74	0730	65081
13	MAR 05 74	0745	67538
14	MAR 07 74	0745	69040
15	MAR 14 74	0745	70140
16	MAR 06 74	0745	70312
17	MAR 15 74	0745	70844
18	MAR 12 74	0745	71438
19	MAR 13 74	0745	75112
20	MAR 07 74	0800	77068
21	MAR 14 74	0800	77964
22	MAR 13 74	0800	78544
23	MAR 06 74	0800	79480
24	MAR 12 74	0800	79478
25	MAR 15 74	0800	79624
26	MAR 05 74	0800	86132
27	MAR 13 74	0815	82256
28	MAR 06 74	0815	83000
29	MAR 05 74	0815	82564
30	MAR 15 74	0815	83646
31	MAR 12 74	0815	83969
32	MAR 14 74	0815	85418
33	MAR 07 74	0815	91220
34	MAR 12 74	0830	84676
35	MAR 05 74	0830	85249
36	MAR 13 74	0830	85970
37	MAR 07 74	0830	86824
38	MAR 06 74	0830	92800
39	MAR 14 74	0830	94718
40	MAR 15 74	0830	94802
41	MAR 05 74	0845	84444
42	MAR 14 74	0845	86375
43	MAR 07 74	0845	86626
44	MAR 06 74	0845	86888
45	MAR 15 74	0845	88189
46	MAR 12 74	0845	92568

Figure 22. Sample printing from run of program NETCON.

THIRD AND FOURTH FILE SORTS

Following the program NETCON in the next to last required sequential job string are two IBM utility sort steps. They are to sort the PMMCO.UTCS. VOLXXYZ and PMMCO.UTCS. VOLXXZY files, respectively, created by NETCON in the previous step. The sorted files are stored in the files named PMMCO.UTCS.SRT3XXYZ and PMMCO.UTCS.SRT3XXZY, respectively.

The sort puts the observations in ascending order according to the recalculated, consistent network volumes, within the primary sort on clock time. The amount of position changes from the previously-existing sort should be minimal, since the previous sorts on inconsistent network volumes should not have resulted in greatly different orderings.

See the JCL starting with the //SORT3 EXEC and //SORT4 EXEC cards in Step 4 for the detailed sort format to be used.

PROGRAM PREPOST

Summary Description

The final program in the Step 5 chain is PREPOST. Now that link integrity has been assured for the statistical comparison through the definition of a single common link usability vector, it remains only to guarantee observation integrity and assure that the observations in the two cases to be compared reflect roughly similar travel demand conditions. Both of these needs, as well as the requirements of equal statistical samples sizes for the two cases, are satisfied by an observation "matching" process performed by this program.

The "matching" process is briefly as follows:

1. One starts with one set of observations for each of the two cases to be compared. They are in ascending order by 24-hour clock time and within a given clock time they are ordered by network volume. The number of observations for the two cases is generally different.

2. All "matching" of observations between these groups occurs only within the same clock time. No matches of observations with different clock times are permitted.

3. For each clock time, we compare the volume ranges for the two cases. Extreme observations for any clock time,

as measured by its network volume, are "discarded" if they are further from center than their corresponding extreme by a certain criterion value. This range truncation procedure provides protection against the possibility of comparing two sets of observations which differ greatly in network volume values (as a surrogate for travel demand conditions).

4. Following the truncation procedure, it is necessary to insure that the number of observations as between the two cases is equal for each clock time separately, which will make the remaining samples for each clock time equal as between the two cases.

This is done by removing the "n" observations in the larger of the two (if they are not already of equal size) which have network volumes closest to the mean network volume for its respective clock time sample of observations following truncation. This procedure attempts to minimize the impact of equalizing sample sizes on the original truncated samples.

In cases where one of the two cases has only one observation remaining after truncation, the procedure is then, instead, to match it with the observation in the other case having the closest network volume to its own. All other observations are then discarded.

5. Since the original files were sorted by network volume within clock time, it is necessary only to specify which of the original observations should not be used to specify the "match". Each remaining observation is simply matched with the corresponding remaining observations in the other case in succession.

6. Although this procedure could be considered as producing an actual observation-to-observation match. this is not its objective. The "matching" procedure is performed to obtain two sample sets of equal sample size and representing roughly comparable travel demand conditions. Since the primary statistical tests to be used will essentially pool all observations for a given control pattern and time of day for the analysis, the one-to-one "matching" of observations is not used in these (link-by-link) tests.

The primary output of this program is thus two vectors specifying which observations are to be used and which not for statistical purposes when these two cases are being compared.

Program Limits and Variable Dimensions

The variables are dimensioned so as to permit 100 observations for each of the two cases. However, not more than 15 distinct clock times (present in either of the two samples) will be permitted by the present dimensioning; also, no more than 15 observations in any given clock time group for a case will be allowed. Since this program concerns itself only with observations, the number of links or subnetworks involved is irrelevant.

Definitions of Important Variables

1. K1(I), K2(I) INTEGER

The ranking of the I'th observation of the final resorting in the files created by program BRKFIL and all following programs up until the last two sorts in Step 4. These permit accurate manipulation of these files in subsequent steps without the added steps involved in physically resorting them. The K1 and K2 refer to the first and second cases to be compared, respectively.

2. The DATE1, TIME1, NTVOL1, etc., variables are merely observation-subscripted versions of DATE, TIME, and NTVOL. DATD1, TIMD1, NVOL1, etc., are versions of DATE1, TIME1, NTCOL1, etc. used to hold sequentially only values for observations that will actually be used in the statistical comparisons.

3. NIT1(I), NIT2(I) INTEGER

The number of observations in the I'th (15 minute) clock time group for the first and second cases, respectively.

4. KNT1(L), KNT2(L) INTEGER

The counterparts of K1(I) and K2(I), respectively, storing successively the ranks in the pre-final-sort files of only these observations that will be used in the statistical comparisons.

5. I1S, I2S INTEGER

The observation number of the first observation in the clock time group presently under consideration for cases 1 and 2, respectively.

6. IT INTEGER

The clock time group presently under consideration.

7. IEND1, IEND2 INTEGER

These indices are set equal to one to indicate that the last of all observations for cases one and two, respectively, has been reached in the matching process.

Technical/Quantitative Details

None in the body of the main program.

Program Parameters and Tests

None in the body of the main program.

Subroutines Referenced

The main program calls two subroutines directly: READIT and MATCH. Subroutine MATCH in turn calls subroutine SORTR, which, itself calls subroutine SWITCH. Briefly, their functions are as follows:

Subroutine READIT

This subroutine is used to advance the main program through to the next clock time group for which there are observations for both cases. The number of observations in this group for both cases is stored, all observations in any unpaired groups that were passed over are deleted, appropriate counters are updated, and if the end of the observations in either or both cases is encountered in this pass, this fact is denoted. Basically, the subroutine prepares for the matching procedure for the next paired clock time group, which occurs in subroutine MATCH.

Subroutine MATCH

This subroutine carries out both the truncation and sample size equalization steps of the "matching" procedure for the clock time period under consideration. Both of these steps are translated into the two vectors of observation usability which summarize the matching process. Although the two cases generally have different number of observations, when the matching process is complete, the two vectors have the same number of "1"s by definition.

The major variables of interest introduced by this subroutine are:

1. IND(I) INTEGER

A dummy index used to sort the differences between individual values and the mean into ascending order.

2. DEL(I) REAL

The absolute value of the difference between the network volume of the I'th observation in a clock time group and that of the mean of the group.

3. NITA, NITB INTEGER

Unsubscripted substitutes for NIT1(IT) and NIT2(IT), respectively, within the subroutines.

4. ICHG INTEGER

A flag that is assigned the value of 1 if any truncations were made in the last pass. If ICHG=0 after a pass, no further truncation passes are necessary.

5. VBAR1, VBAR2 REAL

The mean network volume of all observations in a clock time group after truncation for cases 1 and 2, respectively.

6. ICRIT INTEGER

The criterion value for the truncation procedure. This is the maximum amount an extreme value of one case can be further from the center than its counterpart in the other case. Its value is determined as follows:

(a) The smaller of the two LOWEST network volumes for a clock time group is subtracted from the larger of the two HIGHEST network volumes. This gives the combined simultaneous volume range for that clock time. This, of course, is done prior to truncation.

(b) This volume range is then divided by the larger of the number of observations in the two groups, minus one. The result is the value of ICRIT for the corresponding clock time.

7. NDEL INTEGER

Is the number of observations that must be "deleted" from the larger of the two clock time groups so as to achieve an equal number of remaining observations for both.

8. KEEP INTEGER

This is the observation that is to be matched with the one observation in the other group for that situation in which one of the cases has only one remaining observation in the clock time group.

9. ITEST INTEGER

The absolute value of the difference between the network volume of the observation for the case with one remaining observation and the observation for the other case in that clock time group with the closest network volume.

In the truncation process, observations are stripped off either end until the maximum extreme divergence criterion is no longer violated.

In the sample size equalization procedure where the remaining sample for the clock time is greater than one for both cases, the larger sample is decreased as follows:

(a) The mean of the volumes for the remaining observations for the larger sample case is computed.

(b) The absolute values of the difference between this mean and the individual remaining volumes are sorted in ascending order.

(c) The observations corresponding to the NDEL first (smallest) absolute differences in this sort are "deleted" and thus the samples sizes are made equal.

The value NINE = '9999' is assigned to the TIME variable for the observation following the last (an imaginary value for testing purposes only) if the last observation is reached for a case but the clock time for this last observation is still less than the last clock time scanned for the other case. Testing for this flag permits the subroutine to deal with all possible occurrences of unpaired clock time groups.

Subroutine SORTR

This subroutine is used to sort the elements of the input vector in the argument list into ascending order and return the original positions of these elements, in sorted order in the vector IND(I). The vector V is itself left unchanged, since the actual sorting manipulations are performed on the dummy vector, DV.

The sort is achieved through a simple interchanging of adjacent elements that are out of order within a double DO loop.

Subroutine SWITCH

This subroutine simply interchanges two vector elements and their corresponding position indices. It is the manipulative basis of the sort performed in subroutine SORTR.

All four subroutines discussed above are physically appended to the end of the main program, PREPOST, in the source file which contains it.

Program Input

Permanent Files

NAME:	PMMCO.UTCS.NOBS(XXY), PMMCO.UTCS.NOBS(XXZ)
SOURCE:	Disk, OLOUT program output
DISPOSITION:	SHR
UNIT NO.'s:	FT10F001 and FT11F001, respectively
NAME:	PMMCO.UTCS.SRT3XXYZ, PMMCO.UTCS.SRT3XXZY
SOURCE:	Disk, third and fourth IBM sort output files (SORT3 and SORT4 output)
DISPOSITION:	SHR
UNIT NO.'s:	FT15F001 and FT15F001, respectively

Temporary Files

None

Control/Data Cards

None

Program Output

Permanent Files

NAME:	PMMCO.UTCS.OBUSXXYZ

DESCRIPTION: The observation usage vectors for cases XXY and XXZ, when compared.

LOCATION: Disk

DISPOSITION: NEW,KEEP

I/O FORMAT:
- (OBUSE1(J1),J1 = 1,NOBS1), (OBUSE2(J2),J2 = 1,NOBS2)
- INTEGER*2 OBUSE1(100), OBUSE2(100)
- 1 record

UNIT NO.: FT18F001

Temporary Files

None

Printer, Error Messages

(See Figure 23.)

DESCRIPTION: First, the two observation usage vectors, OBUSE1 and OBUSE2, are printed out in that order under the title "OBUSE:". This is followed by a tabular representation of the final observation match in a mirror image format centering on the network volumes that served as the basis of the matching process within each clock time. Each line of the table gives the original file position, date, time, and network volume of two observations (one from each case) "matched" together. The lines are in order of increasing network volume within the overall ordering according to clock time. The observations that were thrown out by the matching process do not appear in this table.

PROGRAM CONDNS

Summary Description

This program constitutes an optional step which is executed when statistical analysis over subnetworks or other aggregates of links is desired, in addition to the basic analysis by individual link. The program creates for each case for which it is run a set of four MOE subnetwork

OBUSE:
1001101111111111111101111111111110111110111111111000111100111111101110100011101111111

OBUSE:
11100111011111011111111111111111111100111111101101110111111111111111111100011111111

K1	DATE	TIME	VOL1	VOL2	TIME	DATE	K2
1	MAR 14 74	0715	56055	56834	0715	APR 05 74	1
4	MAR 06 74	0715	57000	57073	0715	APR 02 74	3
5	MAR 05 74	0715	57056	57316	0715	MAR 28 74	2
7	MAR 07 74	0730	66237	67388	0730	APR 04 74	6
8	MAR 13 74	0730	67317	69804	0730	APR 03 74	8
9	MAR 05 74	0730	67690	70004	0730	MAR 27 74	7
10	MAR 15 74	0730	73761	73552	0730	MAR 28 74	10
11	MAR 14 74	0730	74745	75202	0730	APR 02 74	11
12	MAR 06 74	0730	74965	75796	0730	APR 05 74	12
13	MAR 05 74	0745	78490	78044	0745	APR 04 74	13
14	MAR 07 74	0745	79336	81144	0745	MAR 27 74	14
15	MAR 14 74	0745	81142	81372	0745	APR 03 74	16
16	MAR 06 74	0745	81212	81684	0745	MAR 29 74	17
17	MAR 15 74	0745	81992	82136	0745	APR 05 74	18
18	MAR 12 74	0745	82422	82636	0745	MAR 28 74	19
20	MAR 07 74	0800	83896	89016	0800	APR 04 74	20
21	MAR 14 74	0800	89904	91124	0800	MAR 27 74	21
22	MAR 13 74	0800	90560	92064	0800	APR 02 74	22
23	MAR 06 74	0800	91864	92108	0800	APR 05 74	23
24	MAR 12 74	0800	92163	92328	0800	MAR 29 74	24
25	MAR 15 74	0800	92408	92968	0800	MAR 28 74	25
26	MAR 05 74	0800	99408	99016	0800	APR 03 74	26
27	MAR 13 74	0815	95620	95636	0815	APR 02 74	27
28	MAR 06 74	0815	95664	96772	0815	MAR 28 74	28
29	MAR 05 74	0815	96236	96772	0815	APR 03 74	29
30	MAR 15 74	0815	97478	97104	0815	APR 05 74	30
32	MAR 14 74	0815	98742	101648	0815	APR 04 74	31
33	MAR 07 74	0815	105384	106820	0815	MAR 27 74	32
34	MAR 12 74	0830	98604	97372	0830	APR 03 74	35
35	MAR 05 74	0830	98925	100686	0830	MAR 27 74	36
36	MAR 13 74	0830	100070	102052	0830	MAR 29 74	37
38	MAR 06 74	0830	107392	103372	0830	APR 02 74	38
39	MAR 14 74	0830	109242	108760	0830	MAR 28 74	39
40	MAR 15 74	0830	109606	109112	0830	APR 05 74	40
41	MAR 05 74	0845	93024	96976	0845	APR 04 74	42
42	MAR 14 74	0845	100021	99184	0845	MAR 28 74	43
43	MAR 07 74	0845	100496	100029	0845	APR 03 74	46
44	MAR 06 74	0845	101004	100432	0845	MAR 27 74	45
45	MAR 15 74	0845	102105	100803	0845	APR 05 74	47
49	MAR 06 74	0900	96356	96748	0900	APR 04 74	49
50	APR 08 74	0900	96682	98844	0900	APR 05 74	51
51	MAR 12 74	0900	96790	98920	0900	MAR 28 74	50
52	MAR 07 74	0900	97360	99644	0900	MAR 29 74	52
55	MAR 13 74	0900	99473	100480	0900	MAR 27 74	53
56	MAR 05 74	0900	104092	104096	0900	APR 03 74	54
57	MAR 06 74	0915	89596	88180	0915	APR 02 74	55
58	MAR 05 74	0915	90618	91724	0915	APR 03 74	56
59	MAR 12 74	0915	91508	94012	0915	APR 04 74	57
60	APR 06 74	0915	92297	95472	0915	APR 05 74	58
62	MAR 13 74	0915	93865	96052	0915	MAR 28 74	59
63	MAR 14 74	0915	93969	100664	0915	MAR 29 74	60
64	MAR 07 74	0915	100312	101436	0915	MAR 27 74	61

Figure 23. Sample printout from run of program PREPOST.

level data files, analogous to the seven MOE individual link level data files created by program BRKFIL.

The four "MOE's" for the subnetwork level files are four MOE's or arithmetic combinations of MOE's which were deemed to be meaningfully aggregatable over links. These are: average number of cars queued per cycle, number of stops, vehicle minutes of travel time, and vehicle minutes of delay.

Based on a set of control cards, the program constructs a number of subnetworks or link aggregations defined as one or more of the original 270 links. A subnetwork usability vector is constructed based on the usability of links in each subnetwork and it is stored in the LUSEG(J) variable for output. The aggregate values for the four special MOE's are calculated for each subnetwork and these are then stored in four subnetwork MOE files, where each entry represents the value of the "MOE" over the total corresponding subnetwork instead of the value of a MOE for a single link. The computed aggregate "MOE" values are also printed out in tabular form.

Program Limits and Variable Dimensions

The program assumes that the input files contain no more than 270 links in total and that each of the following MOE's exist in the general format of the seven MOE files created by each execution of the program BRKFIL: hourly vehicle flow rate, average number of vehicles queued per cycle, number of stops, link travel time in seconds (times 32), and average vehicle delay in seconds.

With respect to the subnetwork aggregates constructed internally, the variable dimensions are such that: (a) no more than 20 subnetworks can be defined in any given run, including the subnetwork of all usable links, if their program option is used and (b) no more than 150 usable links can be specified in defining any one subnetwork (this was assumed adequate since this exceeds the number of usable multi-detector links found in runs thus far). More than 20 subnetworks can be defined if the program is run more than once. However, it is then necessary to carefully record the different output file names used and the subnetworks associated with each set of such files.

Definitions of Important Variables

1. TABLE(I,J)   INTEGER*2

This gives the integer code name for the J'th link in subnetwork I. I and J vary from 1 to their respective limits (the limits on J varying themselves with I).

2. NLINK(I) INTEGER*2

The number of links used to define subnetwork I on input.

3. NLINK(I) INTEGER*2

The number of usable multi-detector links actually used for aggregation purposes. NLINK2(I) ≤ NLINK(I).

4. DUM(J) INTEGER*2

A dummy input vector used to read in the J'th integer link code for a given subnetwork. This is to be distinguished from a corresponding row of TABLE(I,J) by the fact that all unusable links that may have been specified on input do not appear in the latter.

5. LNLBL(I) INTEGER CHARACTER

The four-letter subnetwork name used to identify subnetwork or link aggregation I.

6. CONQ(I) REAL

The average total number of vehicles queued per cycle on subnetwork I.

7. CONS(I) REAL

The total number of stops over subnetwork I.

8. CONT(I) REAL

The total number of vehicle minutes of travel over subnetwork I.

9. COND(I) REAL

The total number of vehicle minutes of delay over subnetwork I.

10. NNET INTEGER

The number of subnetworks to be defined in a given CONDNS run.

11. NLKNT INTEGER

The total number of links (including unusable) on input. In subsequent programs, this variable may also stand for the number of subnetworks on input to be analyzed if subnetwork, rather than individual link analysis, is being undertaken in that run.

Technical/Quantitative Details

Since the travel time MOE variable is scaled up by a factor of 32 in the original data files, this program divides by 32 when defining the aggregate travel time MOE. Both travel time and delay aggregate MOE's are divided also by 60 to obtain minutes instead of seconds. These divisions (by 1920. and 60., respectively) were performed so as to make the resulting values more meaningful for inspection purposes.

Program Parameters and Tests

When the value of NLINK(I) is -1, this indicates that the associated subnetwork is to consist of all usable multi-detector links--the total statistically-usable network.

Subroutines Referenced

None

Program Input

Permanent Files

NAME:	PMMCO.UTCS.LUSGXXYZ
SOURCE:	Disk, NETCON program output
DISPOSITION:	SHR
UNIT NO.:	FT10F001

NAME:	PMMCO.UTCS.NOBS(XXY)
SOURCE:	Disk, OLOUT program output
DISPOSITION:	SHR
UNIT NO.:	FT11F001

NAME:	PMMCO.UTCS.MOEIXXY I=2,4,5,7,8
SOURCE:	Disk, BRKFIL program output
DISPOSITION:	SHR
UNIT NO.'s:	FT12F001, FT13F001, FT14F001, FT15F001, FT16F001, respectively.

Temporary Files

None

Control/Data Cards

DESCRIPTION:	The control pattern/time of day title card
SOURCE:	Following the FT08F001 DD * card in the JCL
COLUMN DESCRIPTION:	Starting in column 1: "CONTROL PATTERN Y: XXXX where Y is the pattern number and XXXX is either "a.m.", "midday", or "p.m.".
UNIT NO.:	FT08F001
DESCRIPTION:	The subnetwork/link aggregation definition data.
SOURCE:	Following the FT09F001 DD * card in the JCL
COLUMN DESCRIPTION:	On the first card: the number of networks to be defined, including the total usable network (if that option is used), is placed in columns 1-10. The total number of links (including unusable) in the input files appears in columns 11-20.

Following this card is one set of cards for each network to be defined. The first card in each set contains the character label for the corresponding subnetwork in columns 1-4 and the number of links to be used on input to define the subnetwork in columns 11-20. Following this card in each set are as many cards as is necessary to specify all of the links in the corresponding subnetwork. Each card may hold up to 14 link numbers in subsequent adjacent fields of five columns each, starting in column one and not extending past column 70. No internal blank fields are permitted. No link definition cards follow the first card of a set if the flag value of -1 is used to specify the number of links in the network. The computer program will automatically make the proper assignment in this case.

UNIT NO.: FT09F001

Program Output

Permanent Files

NAME: PMMCO.UTCS.CON"I"XXYZ "I" = Q,S,T,D

DESCRIPTION: The aggregate subnetwork level "MOE" files for queues, stops, travel time, and delay time, respectively, for case XXY when being compared with case XXZ.

LOCATION: Disk

DISPOSITION: NEW,KEEP

I/O FORMAT:
- K,(CON"I"(J),J=1,NNET) "I" = Q,S,T,D, respectively
- INTEGER K REAL CONQ(20), CONS(20), CONT(20),COND(20)
- No. of records = no. of observations

UNIT NO.'s: FT17F001, FT18F001, FT19F001, FT23F001, respectively

NAME: PMMCO.UTCS.LUSCXXYZ

DESCRIPTION: The subnetwork usability vector, which indicates which of the specified subnetworks will be usable for statistical purposes (that is, contains at least one usable multi-detector link) when cases XXY and XXZ are compared. 1 indicates usable, 0 indicates not usable. The value "2" is not used for subnetwork usability purposes.

LOCATION: Disk

DISPOSITION: NEW,KEEP

I/O FORMAT:
- (LUSEG(J),J=1,NNET)
  INTEGER*2 LUSEG(270)
- 1 record

UNIT NO.: FT20F001

NAME: PMMCO.UTCS.LBLXXYZ

DESCRIPTION: The four-letter names of the subnetworks defined for this run of the program.

LOCATION: Disk

DISPOSITION: NEW,KEEP

I/O FORMAT:
- (LNLBL(J),J=1,NNET)
- FORMAT(18A4)
- 1 set of records, as many records as are needed to exhaust one writing of the variable list.

UNIT NO.: FT21F001

Temporary Files

None

Printer, Error Messages

(See Figure 24.)

DESCRIPTION: First, the title of the control pattern/time of day case under consideration is printed. This is followed by a listing of all subnetworks which were found to have no usable multi-detector links, if any. Finally, in tabular form the values for all four aggregate "MOE's" for each subnetwork and each observations is printed out.

JCL to Run Program

To make a run, the following must be done:

CONTROL PATTERN 1; A.M.

OBSERV.	NETWORK	QUEUES	STOPS	VEH. MINUTES	MIN. OF DELAY
1	S 1	27.	247.	1118.	909.
1	S 2	74.	619.	2502.	1940.
1	S 3	230.	1747.	7933.	5635.
1	S 4	93.	611.	3225.	2297.
1	S 12	101.	666.	3621.	2649.
1	S 34	323.	2358.	11159.	7932.
1	EAST	85.	561.	3037.	2009.
1	WEST	79.	638.	2806.	2046.
1	NOR.	76.	561.	2630.	1928.
1	SOU.	63.	598.	2686.	1950.
1	L ST	22.	161.	809.	607.
1	K ST	53.	398.	1720.	1030.
1	PENN	33.	253.	1223.	955.
1	CON	41.	254.	1477.	1007.
1	17ST	20.	108.	673.	453.
1	18ST	21.	166.	744.	568.
1	19ST	26.	209.	848.	665.
1	TOTL	424.	3224.	14779.	10781.
2	S 1	28.	239.	1065.	659.
2	S 2	71.	588.	2457.	1889.
2	S 3	217.	1664.	7723.	5492.
2	S 4	68.	570.	3021.	2162.
2	S 12	99.	827.	3542.	2748.
2	S 34	305.	2234.	10744.	7594.
2	EAST	78.	487.	2819.	1773.
2	WEST	69.	552.	2479.	1778.
2	NOR.	77.	568.	2746.	2044.
2	SOU.	81.	627.	2698.	2003.
2	L ST	20.	120.	654.	472.
2	K ST	45.	366.	1730.	1004.
2	PENN	26.	191.	965.	744.
2	CON	38.	227.	1356.	903.
2	17ST	25.	162.	654.	609.
2	18ST	16.	143.	604.	425.
2	19ST	24.	212.	877.	732.
2	TOTL	404.	3061.	14266.	10342.
3	S 1	26.	245.	1115.	900.
3	S 2	72.	561.	2361.	1791.
3	S 3	235.	1872.	8281.	5948.
3	S 4	93.	665.	3418.	2473.
3	S 12	100.	806.	3476.	2691.
3	S 34	328.	2537.	11699.	8421.
3	EAST	63.	551.	3058.	1991.
3	WEST	78.	687.	2785.	2095.
3	NOR.	75.	576.	2741.	2039.
3	SOU.	92.	723.	3115.	2297.
3	L ST	20.	130.	686.	508.
3	K ST	52.	457.	1922.	1227.
3	PENN	26.	220.	1043.	769.
3	CON	40.	261.	1470.	1007.
3	17ST	22.	137.	822.	572.
3	18ST	19.	167.	668.	493.
3	19ST	29.	257.	958.	753.

Figure 24. Sample printout from run of program CONDNS.

1. The title card must identify the appropriate control pattern and time of day.

2. The subnetwork aggregation data cards must be correctly specified.

3. XXYZ file subscripts must pertain to the correct control pattern and time of day, as well as that of the case with which it is being compared.

PROGRAM POSTPROC

Summary Description

The program POSTPROC is executed to perform the statistical computations and tests of the final step. In this final step, the statistical comparison of the system's performance under two different control patterns is made for a given time of day. Each run of the program, together with its associated subroutines, produces one or more tables which present the statistical calculations and tests made, either on an individual link or subnetwork basis, depending upon the control cards and type of MOE files used on input.

Because the program has many output options and because it produces only printed output, POSTPROC can be meaningfully run a number of times for the same pair of control pattern/time of day cases.

Each row in a table represents a separate set of calculations and statistical tests. A row may represent a usable link or subnetwork for which the calculations and tests are made on a sample of different observation values for that link or subnetwork. Alternatively (by changing control parameters), each row can represent a matched pair of observations and the calculations and tests are made across a sample of all usable link values for the respective observations of that row.

By reference to a number of subroutines, the program makes the following calculations and statistical tests (once for each row in each table in the printed output):

1. The mean and standard deviation for the MOE under consideration over the appropriate sample is calculated for the two cases under consideration. The signed percentage difference of these two means is calculated.

2. A two-tailed T test is performed to test the null hypothesis that the mean values of the MOE for the two cases were not significantly different.

3. A two-tailed Mann-Whitney U test is performed to test the same null hypothesis.

4. A Kolmogorov-Smirnov two-sample test is performed. It is intended to be a one-tailed test of the null hypothesis that either the first case's distribution of MOE values is not significantly "greater" than that of the second or vice versa, whichever tail is appropriate in light of the data. However, if a level of significance in rejecting the null hypothesis in both directions is achieved, the significance test defaults to the two-tailed type. In any case, all test statistics are calculated and made available in the printed output.

By varying the control cards, it is possible to obtain tables for one, several, or all of the available MOE's in a given run. Also, it is possible to obtain tables with rows by link/subnetwork and/or by observation for each MOE treated.

Program Limits and Variable Dimensions

The program can handle up to 100 observations for each case on input, but the variables are dimensioned so that no more than 90 observations can be retained as usable for the statistical comparisons (the program assumes that the input data files contain all observations and links; the program will use the appropriate observation and link/subnetwork usability vectors to choose the proper subset of these for statistical analysis internally).

For MOE input data in half-word integer form, up to 270 links or subnetworks are permitted on input. For data in full-word integer or real form, no more than 135 links or subnetworks are allowed. For a given run of the program, only one type of MOE input data is permitted. Unlike previous programs described, POSTPROC reads in the appropriate number of links or subnetworks instead of assuming a standard 270 link format. Thus, this program can be applied to MOE data generated from other sources.

Tests on up to eight different MOE's (if that many MOE files exist) can be made in a given run, on either or both the link/subnetwork and observation bases. However, a single run must pertain to either single links or subnetworks, but not both. All calculations and statistical tests described above are made unless the sample size specified is below the minimum value specified within the program for that test or calculation to be made.

Definitions of Important Variables

1. A(J) REAL

The variable used to hold the MOE value for the Jth member of the sample for a given "row" for the first case. This is the variable used in the actual row-by-row statistical calculations and tests. The element A(J + NMBR) is also used to store the value of B(J) [see below for definitions of NMBR and B(J)].

2. B(J) REAL

As A(J), but for the second case.

3. D(J) REAL

This variable is used as a temporary storage variable in various capacities in conjunction with the statistical subroutines. Its first element is used to return the value of the T statistic and then the statistic for the U test to the main program. In the T test subroutine, it is also used to store in its first three elements the three appropriate calculated critical significance test values in cases of "large" sample size. In the U test and RANK subroutines, the variable which corresponds to D(J) in the calling sequence holds ranks for a sorting of the pooled MOE values for both cases for a given "row."

4. MOE1(I,J) and MOE2(I,J) REAL

The input storage variables for a given MOE for cases 1 and 2, respectively. Element I,J is the value for link J on observation I. These variables are used when the output MOE files contain real word data.

5. IMOE1(I,J) and IMOE2(I,J) INTEGER

The full-word integer counterparts of MOE1 and MOE2, used when the input files are in full-word integer format.

6. IHMOE1(I,J) and IHMOE2(I,J) INTEGER*2

The half-word integer counterparts of MOE1 and MOE2.

7. NMSTAT REAL*8 CHARACTER

The eight-letter character name identifying what each row the corresponding output table represents: a link, link aggregation (subnetwork), or observation.

8. LNKNAM REAL*8

Holds the character string value of 'LINKS' that the variable NMSTAT may assume.

9. IAGNAM REAL*8

Holds the character string value of 'AGGREG.' that NMSTAT may assume.

10. IOBSNM REAL*8

Holds the character string value of 'OBSERV.' that NMSTAT may assume.

11. PTRNA INTEGER

Holds the number of the control pattern for the first case in this run.

12. PTRNB INTEGER

As PTRNA, but for the second case in the run.

13. TOD INTEGER CHARACTER

Holds the four-letter designation for the time of day of the two cases in the run.

14. LNKNT INTEGER

= 1 if individual link MOE files are being read in.

= 2 if subnetwork/link aggregation MOE files are being read in.

15. TWIND INTEGER

The control variable used to specify the type of word used on input in the MOE files:

= 1 Real
= 2 Integer full word
= 3 Integer half word

16. NOBS INTEGER

In this program the variable stores the number of usable observations actually used in the statistical analysis, and thus equal for both cases.

17. AA, IA, IHA REAL, INTEGER, INTERGER*2, respectively

Dummy input variables used to skip over unused file records on input.

18. NMOE INTEGER

The number of MOE's to be used as the basis of case comparison in the run.

19. TESTBS INTEGER

This variable indicates whether the analysis should be done with links/subnetworks, observations, or both (in sequential tables) as rows:

= 1 tables with rows as links or subnetworks only
= 2 tables for both alternatives
= 3 tables with rows as observations only.

20. MOENAM(I,J) INTEGER CHARACTER

(MOENAM(I,J),I=1,5) holds the 20-letter character string describing the Jth MOE for which statistical comparisons are made.

21. SIGP(I) INTEGER CHARACTER

The significance flag associated with a significance value of I-1.

22. SIGKS(IDIR) INTEGER CHARACTER

The one-tail direction indicator for the Kolmogorov-Smirnov test significance flag associated with the given value of IDIR.

23. NUSLNK INTEGER

The number of links usable for the statistical comparison.

24. KDO INTEGER

This indicates the table (either 1 or 2) presently being produced for a given MOE. KDO = 2 occurs only if TESTBS = 2.

25. NTEST INTEGER

This stores the total number of elements from which the table rows will be picked.

26. NMBR INTEGER

This stores the actual sample size to be used in the statistical calculations and tests.

27. TT, TU REAL

The values of the test statistics for the T and U tests, respectively.

28. ITP, ITM IT2 INTEGER

The values of the one-tailed "+", one-tailed "-", and two-tailed Kolmogorov-Smirnov test statistics, respectively.

29. ISIGT, ISIGU, ISIGP, ISIGM, ISIG2, ISIGX INTEGER

The significance levels achieved by the T test; U test; one-tailed "+", one-tailed "-", two-tailed, and selected "appropriate" Kolmogorov-Smirnov tests, respectively.

30. IDIR INTEGER

The indicator of the "appropriate" direction of the Kolmogorov-Smirnov test.

= 1 one-tailed "+"
= 2 one-tailed "-"
= 3 two-tailed

31. ABAR, BBAR REAL

The mean value of the MOE for the given row for cases 1 and 2, respectively.

32. VARA, VARB REAL

The variance of the MOE values for the given row for cases 1 and 2, respectively.

33. STDVA, STDVB REAL

The standard deviations for cases 1 and 2, respectively.

34. DELPER REAL

The signed percent difference in the MOE means for the two cases for a given row.

Technical/Quantitative Details

The variables used to store the input MOE data for real, full-word integer and half-word integer data are equivalenced to share the same core, so that only one type of word can be handled in any given run.

Program Parameters and Tests

1. The significance level achieved in rejecting the null hypothesis in a statistical test is graphically presented through the printing of asterisks (stored in SIGP) following the test statistic value(s). If no significance was achieved, no asterisks are printed. If the lowest level of significance is achieved (95 percent), one asterisk is printed. Two asterisks indicate the second significance level (97.5 percent or 98 percent) and three asterisks, the highest significance level (99 percent). If the test could not be made for any reason, an "NT" is printed instead.

2. Just before the significance level indicator for the Kolmogorov-Smirnov in each row of a table is printed, the test's tail direction indicator appears. A "<" indicates that case 1 (condition A) had a distribution of MOE values that was significantly lower than that of case 2 (condition B). A ">" indicates the reverse. The absence of either indicates:

(a) no significance was achieved if no *'s follow this blank, or

(b) a two-tailed significance test was used if one or more *'s do follow the blank.

3. If the sample size is less than three, no statistical test or calculation is performed and this fact is printed out instead of the corresponding table.

4. A significance level of four signifies that the corresponding test was not performed and an "NT" is printed out.

5. The following values of IDIR have the following significances:

IDIR = 1: "+" one-tailed test. Print "<"

IDIR = 2: "-" one-tailed test. Print ">"

IDIR = 3: two-tailed test default. Print a blank

6. The T test is not performed unless the sample size is at least nine. The U test is not performed unless the sample size is at least four.

Subroutines Referenced

The main program POSTPROC directly calls the subroutines READER, KSTEST, CALC, TTEST, AND UTEST. In turn, KSTEST calls SWITCH (which is physically appended to the end of KSTEST in the file containing this first-level subroutine) and UTEST calls RANK (which is physically part of the source file containing all of the subroutines except READER, KSTEST, and SWITCH).

Subroutine READER is physically appended to the end of POSTPROC. KSTEST and SWITCH are in a separate source file under the name of the former. All other subroutines are in another file named POSTSUB. The functions of the subroutines are as follows.

Subroutine Reader

This subroutine is called for every row of every table just before the statistical subroutines are called for that row. Its sole function is to place in the vectors A and B the MOE values for cases 1 and 2, respectively, for all statistically-usable elements in the sample for the current row of the table. The values stored in A and B are the row data values used as input to the statistical subroutines. Depending on the type of analysis being done, these two vectors may correspond to either a row or column in each of the two large MOE data matrices stored earlier by the main program.

Subroutine KSTEST

This subroutine performs the two-sample Kolmogorov-Smirnov test. This is a nonparametric statistical test that is well adapted to testing for general differences between distributions. The one-tailed test statistics are based on the maximum and minimum signed vertical differences between the cumulative MOE distributions for the two cases, respectively. The two-tailed statistic is based on the maximum absolute vertical difference between the two distributions.

For sample sizes of 40 or less, the critical significance level values used by the subroutine were the appropriate values taken from test tables. When the sample size exceeded 40, the significance tests had to use an asymptotic approximation formula (based on $\sqrt{2*N}$).

The one-tailed test was assumed unless both one-tailed tests were significant, in which case the two-tailed, undirectional test was used. For the one-tailed tests, the three significance levels were 95 percent, 97.5 percent, and 99 percent. For the two-tailed tests, the significance levels were 95 percent, 98 percent, and 99 percent.

The subroutine SWITCH is called by KSTEST as part of the process of sorting the set of MOE values pooled from both A and B. This sort precedes the calculation of the test statistics.

<u>Subroutine CALC</u>

This subroutine calculates the mean value and variance for input vectors A and B.

<u>Subroutine TTEST</u>

This subroutine performs a two-tailed T test to see if the null hypothesis that the mean MOE value for the two cases are equal can be rejected. The T statistic is calculated, assuming that the standard deviation of the two populations involved is not in general equal:

$$T = \frac{\left|\bar{A} - \bar{B}\right|}{\sqrt{\dfrac{\sum_{I=1}^{n} [(A(I) - \bar{A}^2 + (B(I) - \bar{B})^2]}{N*(N-1)}}}$$

where N is the common sample size for the two cases (the number of actual elements in vectors A or B). The degrees of freedom used was 2*N-2.

The three significance levels for this test were 95 percent, 98 percent, and 99 percent (corresponding to *, **, and ***, respectively). Critical significance values for df> 25 were estimated using linear interpolation of table values, with a lower bound of the critical values for df = ∞.

Subroutine UTEST

This subroutine performs a two-tailed Mann-Whitney U test, a nonparametric test to determine if the two distributions of MOE values (A and B) are significantly different. The test statistic is based on a sum of ranks from a pooling of A and B values, where tied values are all given a rank equal to the average of all the adjacent integral ranks that these numbers would have assumed if they were all slightly different from each other. The ranking procedure is accomplished by calling subroutine RANK.

The three significance levels are 95 percent, 98 percent, and 99 percent. For sample sizes exceeding 20, a test approximation based on the normal distribution is used instead of the exact critical table values.

Program Input

Permanent Files

NAME:	PMMCO.UTCS.LBLXXYZ
SOURCE:	Disk, CONDNS program output (replace this unit no. file reference with a DD * card if an appropriate CONDNS has not been run and LNKNT≠2
DISPOSITION:	SHR
UNIT NO.:	FT11F001

NAME:	PMMCO.UTCS.LUSGXXYZ for individual link analysis PMMCO.UTCS.LUSCXXYZ for subnetwork analysis
SOURCE:	Disk. NETCON program output and CONDNS program output, respectively
DISPOSITION:	SHR
UNIT NO.:	FT12F001

NAME:	PMMCO.UTCS.OBUSXXYZ
SOURCE:	Disk, PREPOST program output
DISPOSITION:	SHR

UNIT NO.: FT13F001

NAME: PMMCO.UTCS.MOEIXXY for individual link analysis, where all NMOE different I files are concatenated in the correct order for this unit number. PMMCO.UTCS.CON"I"XXY for subnetwork analysis, where as many "I" from Q,S, T, and D as are desired are concatenated for this unit number in the desired order.

SOURCE: Disk. BRKFIL program output and CONDNS program output, respectively

DISPOSITION: SHR

UNIT NO.: FT15F001

NAME:
SOURCE:
DISPOSITION: As with the unit no. FT15F001 file(s), but Y is everywhere replaced by Z

UNIT NO.: FT16F001

Temporary Files

None

Control/Data Cards

DESCRIPTION: Program control parameters

SOURCE: Following the FT05F001 DD * card.

NUMBER: As required, normally one card.

CONTENT: FORTRAN NAMELIST of the following type:

&PARAM	NAMELIST identifier
NMOE = n	number of MOE's for testing
TESTBS = n	type of tests to be performed
	n=1: link or subnetwork statistics only

2: link or subnetwork and observation statistics

3: observation statistics only

LNKNT = n type of analysis

n=1: link analysis

2: subnetwork analysis

TWIND = n type of data

n=1: binary floating point full words

2: binary integer full words

3: binary integer half words

&END NAMELIST terminator

Note: All values must be supplied.

DESCRIPTION: Label for alternatives

SOURCE: Concatenated to the above card.

NUMBER: Two cards

CONTENT: Card 1: Alphanumeric description of the first alternative.
Card 2: Alphanumeric description of the second alternative.

FORMAT: (20A4)

DESCRIPTION: Number of observations

SOURCE: Concatenated to the above cards

NUMBER: Two cards, one for each alternative.

CONTENT: Card 1: NOBS1
Card 2: NOBS2

FORMAT: (I10)

DESCRIPTION:	Number of links or subnetworks
SOURCE:	Concatenated to the above cards
NUMBER:	One card
FORMAT:	(I10)
DESCRIPTION:	MOE labels
SOURCE:	Concatenated to the above card
NUMBER:	NMOE cards
CONTENT:	Card n: contains an alphanumeric level for the n'th MOE.
FORMAT:	(5A4)

Program Output

Permanent Files

None

Temporary Files

None

Printer, Error Messages

(See Figure 25.)

Description

The normal printer output will consist of one or more statistical comparison tables. Each will be described as to the two cases being compared; whether the rows of the table will represent links, subnetworks, or observations; the sample size involved; and the MOE on which the comparison is being made. All tables for a given MOE will be printed before any tables for other MOE's are printed.

Each table will have column headings which will identify the following information that will appear for each row:

1. the link or observation number or subnetwork label for the row;

STATISTICAL COMPARISON OF A MEASURE OF EFFECTIVENESS FOR CONDITIONS "A" AND "B"

CONDITION A : CONTROL PATTERN 3 AM DETECTOR MOES
CONDITION B : CONTROL PATTERN 1 AM DETECTOR MOES
COMPARISONS MADE BY LINK

STATISTICAL SAMPLE SIZE USED : 53

BASIS OF COMPARISONS : AVG. SPEED BY LINKS

LNKNT /	CONDITION A		CONDITION B		DIFFERENCE	T TEST	U TEST	KOLMOGOROV-SMIRNOV			
OBSERV.	MEAN	STDV	MEAN	STDV	OF MEANS	T	U/Z	T+	T-	T2	SIG.
3	20.3	2.0	17.8	2.2	12.6 %	5.921 ***	-5.143 ***	0	24	24	>***
5	19.6	3.2	20.8	3.1	-5.9 %	1.953	-1.640	9	0	9	
7	20.5	1.6	22.7	1.9	-10.0 %	5.959 ***	-5.118 ***	22	0	22	<***
8	16.5	1.7	17.3	1.6	-4.2 %	2.086 *	-1.769	10	0	10	
10	21.7	2.0	21.6	1.7	0.7 %	0.357	-0.269	5	9	9	
20	28.2	1.5	29.6	1.7	-4.8 %	4.434 ***	-4.003 ***	18	0	18	<***
21	24.8	1.6	19.7	1.0	23.0 %	19.673 ***	-6.824 ***	0	52	52	>***
24	23.4	4.3	22.5	4.4	4.0 %	1.077	-0.512	7	9	9	
25	22.3	3.0	21.8	3.4	3.1 %	1.067	-0.663	3	8	8	
27	23.4	1.8	22.6	1.6	3.2 %	2.286 *	-1.902	0	9	9	
30	16.4	1.3	19.6	1.3	-18.1 %	12.750 ***	-8.037 ***	43	0	43	<***
33	21.1	1.5	21.5	1.2	-2.0 %	1.617	-1.267	6	0	6	
34	24.8	4.7	27.1	2.7	-6.5 %	2.963 ***	-3.118 ***	22	6	22	<***
38	21.6	2.5	23.4	2.4	-3.0 %	3.808 ***	-3.551 ***	17	0	17	<***
40	27.0	1.3	25.3	1.5	6.3 %	5.960 ***	-5.156 ***	0	24	24	>***
42	25.6	2.1	23.9	2.9	7.0 %	3.548 ***	-3.112 ***	0	14	14	>**
43	22.6	4.3	22.3	4.1	1.3 %	0.368	-0.543	7	9	9	
44	23.1	1.9	19.8	1.2	15.2 %	10.795 ***	-7.835 ***	0	40	40	>***
46	20.8	1.3	20.5	1.7	1.2 %	0.815	-0.777	2	9	9	
47	23.6	1.6	20.4	1.1	14.4 %	11.776 ***	-7.660 ***	0	39	39	>***
49	26.8	1.7	28.5	2.4	-6.0 %	4.175 ***	-3.766 ***	20	0	20	<***
62	20.7	2.5	19.4	2.5	6.8 %	2.813 ***	-2.407 **	0	10	10	
63	16.5	4.9	17.4	5.2	-5.2 %	0.900	-0.897	8	2	8	
65	23.5	2.5	24.0	2.8	-2.3 %	1.064	-0.976	9	1	9	
66	19.7	4.3	16.9	4.3	15.7 %	3.421 ***	-3.468 ***	2	17	17	>***
67	22.4	4.9	21.4	3.3	4.9 %	1.333	-1.602	7	14	14	>**
69	20.8	2.1	21.4	1.3	-2.6 %	1.704	-0.767	6	2	6	
72	23.1	2.9	22.2	2.8	3.9 %	1.607	-2.054 *	1	10	10	
73	24.9	2.7	22.5	3.2	9.9 %	4.035 ***	-3.536 ***	0	18	18	>***
74	19.3	2.0	21.7	3.6	-11.8 %	4.255 ***	-4.698 ***	26	4	26	<***
75	23.3	2.6	21.5	1.1	7.8 %	4.514 ***	-4.344 ***	1	19	19	>***
77	29.0	2.8	25.7	4.5	12.2 %	4.596 ***	-4.094 ***	0	22	22	>***
80	18.9	4.0	18.8	3.9	0.5 %	0.123	-0.130	7	3	7	
88	22.2	1.6	18.2	4.8	19.7 %	5.794 ***	-5.339 ***	0	27	27	>***
89	22.2	2.3	23.1	2.5	-4.3 %	2.098 *	-1.937	10	0	10	
90	16.6	1.4	15.6	1.9	5.9 %	2.921 ***	-3.232 ***	2	18	18	>***
95	16.0	5.4	15.6	3.7	2.5 %	0.444	-0.085	5	8	8	
98	18.0	3.0	15.9	4.3	12.3 %	2.872 ***	-2.502 **	1	14	14	>**
99	10.6	5.0	13.2	5.0	-21.4 %	2.618 **	-2.938 ***	16	0	16	<***
104	21.5	3.3	23.2	3.1	-7.3 %	2.633 **	-2.711 ***	15	0	15	<**
105	16.2	1.4	17.6	1.1	-8.8 %	6.056 ***	-5.150 ***	23	0	23	<***
108	26.5	2.2	24.9	1.9	5.9 %	3.604 ***	-3.801 ***	1	19	19	>***
109	25.6	5.7	21.1	4.5	19.3 %	4.554 ***	-4.316 ***	2	23	23	>***
110	17.2	1.8	20.2	1.5	-15.9 %	9.911 ***	-7.386 ***	39	0	39	<***
111	22.9	1.5	22.7	1.3	1.1 %	0.899	-1.232	2	8	8	
112	17.6	1.9	17.2	2.4	3.6 %	1.467	-1.700	5	11	11	
114	23.2	4.7	23.2	4.2	0.2 %	0.044	-0.265	3	4	4	
124	23.5	3.0	20.4	2.6	14.0 %	5.626 ***	-4.954 ***	0	28	28	>***
125	16.9	2.0	19.6	2.4	-3.6 %	1.694	-1.681	9	1	9	
127	19.4	3.9	21.5	3.3	-10.2 %	2.997 ***	-3.055 ***	15	0	15	<**

Figure 25. Sample printout from run of program POSTPROC.

2. the mean value and standard deviation for case 1, followed by the same for case 2;

3. the signed percentage difference (using the average as divisor) of the two means;

4. the value of the T test statistics and the significance level it represents (indicated by the number of asterisks following this value);

5. the value of the U test statistic and the significance level it represents;

6. the values of the three Kolmogorov-Smirnov test statistics (one-tailed "+", one-tailed "-", and two-tailed) and the significance level for the "appropriate" one of the three tests. The direction indicator or its absence will specify which of the three tests was chosen as "appropriate".

If the sample size was below the minimum necessary number specified in the program, the message "TOO FEW OBSERVATIONS [or "USABLE LINKS", if appropriate] AVAILABLE TO TEST:", followed by the number involved, will be printed out instead of the usual table.

JCL to Run Program

To make a run with POSTPROC the JCL must be appropriate with respect to:

1. the control/data cards specifying the control parameters and the names of the MOE's to be used;

2. the XXY, XXZ, and XXYZ file name subscripts used;

3. the correct concatenation of MOE files to correspond with the MOE names specified;

4. if the analysis is to be done on a subnetwork basis, the link/subnetwork usability vector and MOE input file names must be those corresponding to the subnetwork files.

OPERATIONAL PROCEDURES

The detector data analysis programs can either be executed individually in the proper sequence or joined together in the form of catalogued procedures. The latter mode of operation reduces the number of permanent data sets, decreases the probability of user error, and facilitates the repetitive application of the programs through the use of symbolic parameters in the data set names.

Two catalogued procedures have been developed to process the detector data. The first procedure (named STRING) should be applied for each time-of-day within each control alternative, e.g. alternative 3, a.m. The second procedure (POST) uses the alternative/time-of-day data sets produced by STRING as input to make comparisons between alternatives, e.g., alternative 3 versus alternative 3, a.m. traffic conditions. The programs contained within the catalogued procedures are illustrated in Figures 26 and 27.

A number of sequential and partitioned data sets were created to facilitate the operation of the procedures. These data sets contain control cards and data which correspond directly to the input formats of the individual programs documented earlier. Although the procedures are currently tied very closely to the operational objectives and data formats of the Phase I evaluation project, modifications can readily be made to the input data and control cards to permit other applications.

PROCEDURE STRING

Input Datasets

PMMCO.UTCS.TEMP<u>XXY</u>

Output Datasets

```
PMMCO.UTCS.NOBS(XXY)
          .LABS(XXY)
          .INDGXXY    (optional)
          .LUSEXXY
          .IBARXXY    (optional)
          .GBARXXY    (optional)
          .HEADXXY
          .INDSXXY    (optional)
          .MOE2XXY
```

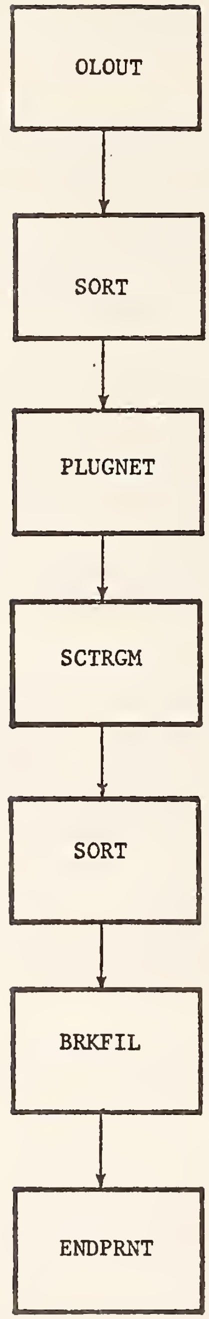

Figure 26. Procedure STRING.

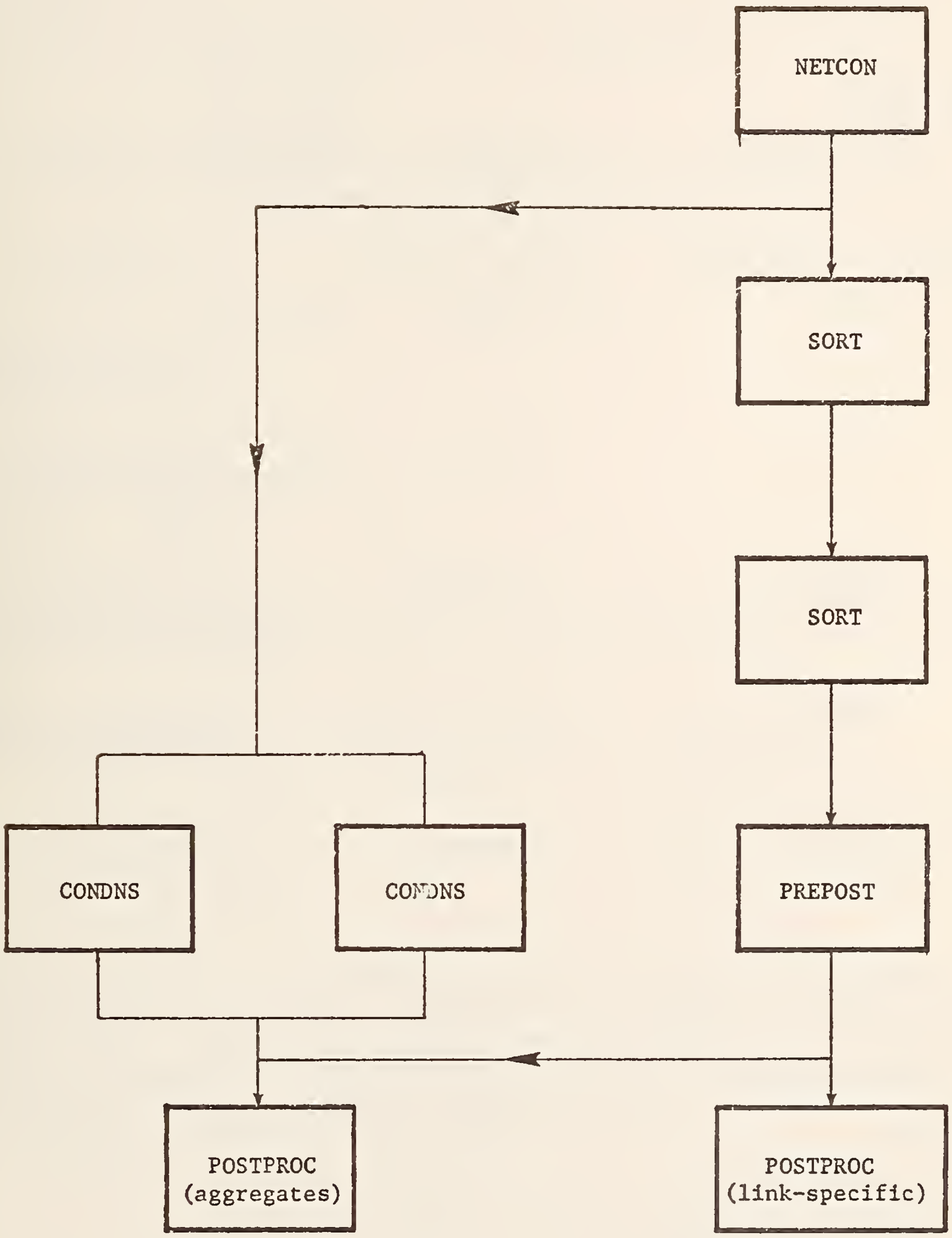

Figure 27. Procedure POST.

.
.
.
.MOE8XXY

Operation

The procedure is invoked by the use of an EXEC card with the appropriate parameters specified:

Symbolic Parameter	Default Value	Description
DDN	none	time-of-day/alternative designation: XXY, e.g., PM3
DINDG	DELETE	disposition of data set PMMCO.UTCS.INDGXXY
DIBAR	DELETE	disposition of data set PMMCO.UTCS.IBAR-XXY
DGBAR	DELETE	disposition of data set PMMCO.UTCS.GBAR-XXY
DINDS	DELETE	disposition of data set PMMCO.UTCS.INDS-XXY

In the most basic mode of operation, the only other input cards are used to produce a unique label to identify the time-of-day/alternative output data sets. The label card can contain up to 72 alphanumeric characters; the format is (18A4). The following sample setup illustrates the basic JCL and control cards needed to execute the catalogued procedure:

```
//EXEC STRING,DDN=PM3,DGBAR=KEEP
//ON.SYSUTI DD *
CONTROL PATTERN 3 PM DETECTOR MOES
```

In this example, the optional output data set PMMCO.UTCS.GBARPM3 has been retained for future use.

PROCEDURE POST

Input Datasets

```
PMMCO.UTCS.NOBS(XXY)
          .LABS(XXY)
          .LUSEXXY
          .HEADXXY
          .MOE2XXY
          .MOE3XXY    (optional)
              .
              .
              .
          .MOE8XXY    (optional)
```

Output Datasets

```
PMMCO.UTCS.SRT3XXYZ
PMMCO.UTCS.SRT3XXZY
```

Operation

Procedure POST is invoked by the use of an EXEC card containing the appropriate parameters and their associated values:

Symbolic Parameter	Default Value	Description
DN1	none	time-of-day/alternative designation for first alternative to be compared: XXY, e.g., PM3
DN2	none	time-of-day/alternative designation for second alternative to be compared: XXY, e.g., PM4
D12	none	time-of-day/alternative comparison designation: XXYZ, e.g., PM34
D21	none	time-of-day/alternative comparison designation: XXZY, e.g., PM43

D2	2	identification number of first MOE to be processed
.	.	.
.	.	.
.	.	.
D8	8	identification number of eighth MOE to be processed, if any
LUSE	YES	member name of partitioned data set PMMCO.UTCS.LUSE; member should contain control card and data for input on FORTRAN unit 5 in program NETCON (see program documentation)
PTIME	15	time limit (in CPU minutes) for step POSTP

The only other required input cards are associated with the control card for the link-specific execution of program POSTPROC. The parameters on this card are explained in the program description in the preceding section. The following sample setup demonstrates the flexibility of the catalogued procedure:

```
//EXEC POST, DN1=PM3,DN2=PM4,D12=PM34,D21=PM43
//D2=6,D3=7,D4=8
//ON.SYSUT1 DD *
 &PARAM NMOE=3,LNKNT=1,TESTB=1,TWIND=3 &END
```

In this example, statistical comparisons are made between control alternatives 3 and 4 for p.m. traffic conditions. Link-specific analyses are produced for three MOE's: average speed, travel time, and delay.

Figures 28 and 29 show the basic programs for procedures STRING and POST, respectively.

```
//STRING PROC DDN=XXX,DINDG=DELETE,DIBAR=DELETE,DGBAR=DELETE,
// DINDS=DELETE
//ON EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT2 DD DSN=PMMCO.UTCS.LABS(&DDN),LABEL=(,,,OUT),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=OLD,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80),SPACE=(TRK,(1,1,10))
//ZAP EXEC PGM=IEFBR14
//A1 DD DSN=PMMCO.UTCS.INDG&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//A2 DD DSN=PMMCO.UTCS.LUSF&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//A3 DD DSN=PMMCO.UTCS.IBAR&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//A4 DD DSN=PMMCO.UTCS.GBAR&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//A5 DD DSN=PMMCO.UTCS.HEAD&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//A6 DD DSN=PMMCO.UTCS.INDS&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//B2 DD DSN=PMMCO.UTCS.MOE2&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//B3 DD DSN=PMMCO.UTCS.MOE3&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//B4 DD DSN=PMMCO.UTCS.MOE4&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//B5 DD DSN=PMMCO.UTCS.MOE5&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//B6 DD DSN=PMMCO.UTCS.MOE6&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//B7 DD DSN=PMMCO.UTCS.MOE7&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//B8 DD DSN=PMMCO.UTCS.MOE8&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//OLOUT EXEC PGM=OLOUT
//STEPLIB DD DSN=PMMCO.UTCS.LOAD1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT09F001 DD DSN=PMMCO.UTCS.CC(OLOUT),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
//FT10F001 DD DSN=PMMCO.UTCS.TEMP&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT11F001 DD DSN=&&TEMP1,
// UNIT=SYSDA,DISP=(NEW,PASS),
// DCB=(RECFM=VBS,LRECL=4892,BLKSIZE=7294),
// SPACE=(7294,(100,9),RLSE)
//FT12F001 DD DSN=PMMCO.UTCS.NOBS(&DDN),LABEL=(,,,OUT),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=OLD,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80),SPACE=(TRK,(1,1,10))
```

Figure 28. First detector data process procedure STRING.

```
//FT13F001 DD DSN=PMMCO.UTCS.INDG&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,&DINDG,DELETE),
// DCB=(RECFM=VBS,LRECL=3244,BLKSIZE=3248),
// SPACE=(3248,(1,1))
//FT14F001 DD DSN=PMMCO.UTCS.LUSE&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP),
// DCB=(RECFM=VBS,LRECL=544,BLKSIZE=548),
// SPACE=(548,(1,1))
//SORT1 EXEC PGM=SORT,COND=(0,LT),PARM='MSG=AP,CORE=100000'
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SYSIN DD DSN=PMMCO.UTCS.CC(SORT1),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
//SORTIN DD DSN=&&TEMP1,DISP=(OLD,DELETE),UNIT=SYSDA
//SORTOUT DD DSN=&&TEMP2,
// UNIT=SYSDA,DISP=(NEW,PASS),
// DCB=(RECFM=VBS,LRECL=4892,BLKSIZE=7294),
// SPACE=(7294,(100,9),RLSE)
//PLUG EXEC PGM=PLUGNFT,COND=(0,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT03F001 DD SYSOUT=A
//FT09F001 DD DSN=PMMCO.UTCS.LUSE&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT11F001 DD DSN=PMMCO.UTCS.NOBS(&DDN),LABEL=(,,,IN),VOL=SER=OTHERS,
// DISP=SHR,UNIT=SYSDA
//FT12F001 DD DSN=PMMCO.UTCS.LABS(&DDN),LABEL=(,,,IN),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT13F001 DD DSN=&&TEMP2,DISP=(OLD,DELETE),UNIT=SYSDA
//FT14F001 DD DSN=PMMCO.UTCS.IBAR&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,&DIBAR,DELETE),
// DCB=(RECFM=VBS,LRECL=7564,BLKSIZE=7294),
// SPACE=(7294,(40,2),RLSE)
//FT15F001 DD DSN=PMMCO.UTCS.GBAR&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,&DGBAR,DELETE),
// DCB=(RECFM=VBS,LRECL=7564,BLKSIZE=7294),
// SPACE=(7294,(2,1),RLSE)
//FT16F001 DD DSN=&&TEMP3,
// UNIT=SYSDA,DISP=(NEW,PASS),
// DCB=(RECFM=VBS,LRECL=4904,BLKSIZE=7294),
// SPACE=(7294,(100,9),RLSE)
//SCTRGM EXEC PGM=SCTRGM,COND=(0,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT09F001 DD DSN=PMMCO.UTCS.LABS(&DDN),LABEL=(,,,IN),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT10F001 DD DSN=PMMCO.UTCS.NOBS(&DDN),LABEL=(,,,IN),VOL=SER=OTHERS,
// DISP=SHR,UNIT=SYSDA
//FT11F001 DD DSN=PMMCO.UTCS.LUSE&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
```

Figure 28. First detector data process procedure STRING (continued).

```
//FT12F001 DD DSN=&&TEMP3,DISP=(OLD,PASS),UNIT=SYSDA
//SORT2 EXEC PGM=SORT,COND=(0,LT),PARM='MSG=AP,CORE=100000'
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SYSIN DD DSN=PMMCO.UTCS.CC(SORT2),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
//SORTIN DD DSN=&&TEMP3,DISP=(OLD,DELETE),UNIT=SYSDA
//SORTOUT DD DSN=&&TEMP4,
// UNIT=SYSDA,DISP=(NEW,PASS),
// DCB=(RECFM=VBS,LRECL=4904,BLKSIZE=7294),
// SPACE=(7294,(100,9),RLSE)
//BRKFIL EXEC PGM=BRKFIL,COND=(0,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT06F001 DD SYSOUT=A,DCB=BUFNO=1
//FT09F001 DD DSN=PMMCO.UTCS.NOBS(&DDN),LABEL=(,,,IN),VOL=SER=OTHERS,
// DCB=BUFNO=1,DISP=SHR,UNIT=SYSDA
//FT10F001 DD DSN=&&TEMP4,DISP=SHR,UNIT=SYSDA,DCB=BUFNO=1
//FT11F001 DD DSN=PMMCO.UTCS.HEAD&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=48,BLKSIZE=2498,BUFNO=1),
// SPACE=(2498,(2,1),RLSE)
//FT12F001 DD DSN=PMMCO.UTCS.INDS&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,&DINDS,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
//FT13F001 DD DSN=PMMCO.UTCS.MOE2&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
//FT14F001 DD DSN=PMMCO.UTCS.MOE3&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
//FT15F001 DD DSN=PMMCO.UTCS.MOE4&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
//FT16F001 DD DSN=PMMCO.UTCS.MOE5&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
//FT17F001 DD DSN=PMMCO.UTCS.MOE6&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
```

Figure 28. First detector data process procedure STRING (continued).

```
//FT18F001 DD DSN=PMMCO.UTCS.MOE7&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
//FT19F001 DD DSN=PMMCO.UTCS.MOE8&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=548,BLKSIZE=7294,BUFNO=1),
// SPACE=(7294,(9,1),RLSE)
//ENDPRNT EXEC PGM=ENDPRNT,COND=(0,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD1,VOL=SER=OTHERS,DISP=SHR,UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT09F001 DD DSN=PMMCO.UTCS.NOBS(&DDN),LABEL=(,,,IN),VOL=SER=OTHERS,
// DISP=SHR,UNIT=SYSDA
//FT10F001 DD DSN=PMMCO.UTCS.LUSE&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT11F001 DD DSN=PMMCO.UTCS.HEAD&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT12F001 DD DSN=PMMCO.UTCS.MOE2&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT13F001 DD DSN=PMMCO.UTCS.MOF8&DDN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
```

Figure 28. First detector data process procedure STRING (continued).

```
//POST PROC DN1=A,DN2=B,D12=C,D21=D,D2=2,D3=3,D4=4,D5=5,D6=6,
// D7=7,D8=8,LUSE=YES,PTIME=15
//DN EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT2 DD DSN=PMMCO.UTCS.CC(POST),UNIT=SYSDA,VOL=SER=OTHERS,DISP=OLD,
// LABEL=(,,,OUT)
//ZAP EXEC PGM=IEFBR14
//D1 DD DSN=PMMCO.UTCS.LUSG&D12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//D2 DD DSN=PMMCO.UTCS.VOL&D12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//D3 DD DSN=PMMCO.UTCS.VOL&D21,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//D4 DD DSN=PMMCO.UTCS.SRT3&D12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//D5 DD DSN=PMMCO.UTCS.SRT3&D21,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//D7 DD DSN=PMMCO.UTCS.OBUS&D12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//NETCON EXEC PGM=NETCON
//STEPLIB DD DSN=PMMCO.UTCS.LOAD2,DISP=SHR,VOL=SER=OTHERS,UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT10F001 DD DSN=PMMCO.UTCS.NOBS(&DN1),DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA,LABEL=(,,,IN)
//FT11F001 DD DSN=PMMCO.UTCS.NOBS(&DN2),DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA,LABEL=(,,,IN)
//FT12F001 DD DSN=PMMCO.UTCS.LUSE&DN1,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT13F001 DD DSN=PMMCO.UTCS.LUSE&DN2,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT15F001 DD DSN=PMMCO.UTCS.LUSG&D12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=544,BLKSIZE=548),
// SPACE=(548,(1,1))
//FT16F001 DD DSN=PMMCO.UTCS.HEAD&DN1,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
// DD DSN=PMMCO.UTCS.HEAD&DN2,DISP=SHR,VOL=SER=OTHERS,UNIT=SYSDA
//FT17F001 DD DSN=PMMCO.UTCS.MOE2&DN1,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
// DD DSN=PMMCO.UTCS.MOE2&DN2,DISP=SHR,VOL=SER=OTHERS,UNIT=SYSDA
//FT18F001 DD DSN=PMMCO.UTCS.VOL&D12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=28,BLKSIZE=1061),
// SPACE=(1061,(3,1))
//FT19F001 DD DSN=PMMCO.UTCS.VOL&D21,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=28,BLKSIZE=1061),
// SPACE=(1061,(3,1))
//FT05F001 DD DSN=PMMCO.UTCS.LUSE(&LUSE),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
```

Figure 29. Second detector data process procedure POST.

```
//SORT3 EXEC PGM=SORT,PARM='MSG=AP,CORE=100000',COND=(0,LT)
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SYSIN DD DSN=PMMCO.UTCS.CC(SORT34),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
//SORTIN DD DSN=PMMCO.UTCS.VOL&D12,DISP=(OLD,DELETE,DELETE),
// UNIT=SYSDA,VOL=SER=OTHERS
//SORTOUT DD DSN=PMMCO.UTCS.SRT3&D12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=28,BLKSIZE=1061),
// SPACE=(1061,(3,1))
//SORT4 EXEC PGM=SORT,PARM='MSG=AP,CORE=100000',COND=(0,LT)
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)
//SYSIN DD DSN=PMMCO.UTCS.CC(SORT34),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
//SORTIN DD DSN=PMMCO.UTCS.VOL&D21,DISP=(OLD,DELETE,DELETE),
// UNIT=SYSDA,VOL=SER=OTHERS
//SORTOUT DD DSN=PMMCO.UTCS.SRT3&D21,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=23,BLKSIZE=1061),
// SPACE=(1061,(3,1))
//PREPST EXEC PGM=PREPOST,COND=(0,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD2,DISP=SHR,VOL=SER=OTHERS,UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT10F001 DD DSN=PMMCO.UTCS.NOBS(&DN1),DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA,LABEL=(,,,IN)
//FT11F001 DD DSN=PMMCO.UTCS.NOBS(&DN2),DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA,LABEL=(,,,IN)
//FT15F001 DD DSN=PMMCO.UTCS.SRT3&D12,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT16F001 DD DSN=PMMCO.UTCS.SRT3&D21,DISP=SHR,VOL=SER=OTHERS,
// UNIT=SYSDA
//FT18F001 DD DSN=PMMCO.UTCS.OBUS&D12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=104,BLKSIZE=108),
// SPACE=(108,(1,1))
//CONDNS1 EXEC PGM=CONDNS
//STEPLIB DD DSN=PMMCO.UTCS.LOAD2,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT08F001 DD DSN=PMMCO.UTCS.LABS(&DN1),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
//FT09F001 DD DSN=PMMCO.UTCS.AGGRC,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=OLD
```

Figure 29. Second detector data process procedure POST (continued).

```
//FT10F001 DD DSN=PMMCO.UTCS.LUSG&D12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT11F001 DD DSN=PMMCO.UTCS.NOBS(&DN1),LABEL=(,,,IN),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT12F001 DD DSN=PMMCO.UTCS.MOE2&DN1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT13F001 DD DSN=PMMCO.UTCS.MOE4&DN1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT14F001 DD DSN=PMMCO.UTCS.MOE5&DN1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT15F001 DD DSN=PMMCO.UTCS.MOE7&DN1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT16F001 DD DSN=PMMCO.UTCS.MOE8&DN1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT17F001 DD DSN=&&Q&DN1,UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(1,1)),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061)
//FT18F001 DD DSN=&&S&DN1,UNIT=SYSDA,SPACE=(TRK,(1,1)),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061),DISP=(,PASS)
//FT19F001 DD DSN=&&T&DN1,UNIT=SYSDA,SPACE=(TRK,(1,1)),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061),DISP=(,PASS)
//FT23F001 DD DSN=&&D&DN1,UNIT=SYSDA,SPACE=(TRK,(1,1)),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061),DISP=(,PASS)
//FT20F001 DD DSN=&&LUS&DN1,UNIT=SYSDA,SPACE=(TRK,(1,1)),
// DCB=(RECFM=VBS,LRECL=44,BLKSIZE=48),DISP=(,PASS)
//FT21F001 DD DSN=&&LAB&DN1,UNIT=SYSDA,SPACE=(TRK,(1,1)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),DISP=(,PASS)
//CONDNS2 EXEC PGM=CONDNS
//STEPLIB DD DSN=PMMCO.UTCS.LOAD2,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT08F001 DD DSN=PMMCO.UTCS.LABS(&DN2),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
//FT09F001 DD DSN=PMMCO.UTCS.AGGRC,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=OLD
//FT10F001 DD DSN=PMMCO.UTCS.LUSG&D12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT11F001 DD DSN=PMMCO.UTCS.NOBS(&DN2),LABEL=(,,,IN),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT12F001 DD DSN=PMMCO.UTCS.MOE2&DN2,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT13F001 DD DSN=PMMCO.UTCS.MOE4&DN2,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT14F001 DD DSN=PMMCO.UTCS.MOE5&DN2,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT15F001 DD DSN=PMMCO.UTCS.MOE7&DN2,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT16F001 DD DSN=PMMCO.UTCS.MOE8&DN2,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT17F001 DD DSN=&&Q&DN2,UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(1,1)),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061)
```

Figure 29. Second detector data process procedure POST (continued).

```
//FT18F001 DD DSN=&&S&DN2,UNIT=SYSDA,SPACE=(TRK,(1,1)),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061),DISP=(,PASS)
//FT19F001 DD DSN=&&T&DN2,UNIT=SYSDA,SPACE=(TRK,(1,1)),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061),DISP=(,PASS)
//FT23F001 DD DSN=&&D&DN2,UNIT=SYSDA,SPACE=(TRK,(1,1)),
// DCB=(RECFM=VBS,LRECL=88,BLKSIZE=1061),DISP=(,PASS)
//FT20F001 DD DSN=&&LUS&DN2,UNIT=SYSDA,SPACE=(TRK,(1,1)),
// DCB=(RECFM=VBS,LRECL=44,BLKSIZE=48),DISP=(,PASS)
//FT21F001 DD DSN=&&LAB&DN2,UNIT=SYSDA,SPACE=(TRK,(1,1)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),DISP=(,PASS)
//POSTAG EXEC PGM=POSTPROC,TIME=(6,0),COND=(0,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD2,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT11F001 DD DSN=&&LAB&DN1,UNIT=SYSDA,DISP=(OLD,DELETE)
//FT12F001 DD DSN=&&LUS&DN1,UNIT=SYSDA,DISP=(OLD,DELETE)
//FT13F001 DD DSN=PMMCO.UTCS.OBUS&D12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT15F001 DD DSN=&&D&DN1,UNIT=SYSDA,DISP=(OLD,DELETE)
// DD DSN=&&T&DN1,UNIT=SYSDA,DISP=(OLD,DELETE)
// DD DSN=&&Q&DN1,UNIT=SYSDA,DISP=(OLD,DELETE)
// DD DSN=&&S&DN1,UNIT=SYSDA,DISP=(OLD,DELETE)
//FT16F001 DD DSN=&&D&DN2,UNIT=SYSDA,DISP=(OLD,DELETE)
// DD DSN=&&T&DN2,UNIT=SYSDA,DISP=(OLD,DELETE)
// DD DSN=&&Q&DN2,UNIT=SYSDA,DISP=(OLD,DELETE)
// DD DSN=&&S&DN2,UNIT=SYSDA,DISP=(OLD,DELETE)
//FT05F001 DD DSN=PMMCO.UTCS.CC(POSTAG),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.LABS(&DN1),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.LABS(&DN2),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.NOBS(&DN1),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.NOBS(&DN2),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.CC(NUMAG),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.MLABS(M8),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.MLABS(M7),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.MLABS(M4),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.MLABS(M5),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
//POSTP EXEC PGM=POSTPROC,TIME=(&PTIME,0),COND=(0,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD2,DISP=SHR,VOL=SER=OTHERS,UNIT=SYSDA
//FT06F001 DD SYSOUT=A
//FT12F001 DD DSN=PMMCO.UTCS.LUSG&D12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
```

Figure 29. Second detector data process procedure POST (continued).

```
//FT13F001 DD DSN=PMMCO.UTCS.OBUS&D12,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//FT15F001 DD DSN=PMMCO.UTCS.MOE&D2&DN1,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
// DD DSN=PMMCO.UTCS.MOE&D3&DN1,UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
// DD DSN=PMMCO.UTCS.MOE&D4&DN1,UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
// DD DSN=PMMCO.UTCS.MOE&D5&DN1,UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
// DD DSN=PMMCO.UTCS.MOE&D6&DN1,UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
// DD DSN=PMMCO.UTCS.MOE&D7&DN1,UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
// DD DSN=PMMCO.UTCS.MOE&D8&DN1,UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT16F001 DD DSN=PMMCO.UTCS.MOE&D2&DN2,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
// DD DSN=PMMCO.UTCS.MOE&D3&DN2,UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
// DD DSN=PMMCO.UTCS.MOE&D4&DN2,UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
// DD DSN=PMMCO.UTCS.MOE&D5&DN2,UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
// DD DSN=PMMCO.UTCS.MOE&D6&DN2,UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
// DD DSN=PMMCO.UTCS.MOE&D7&DN2,UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
// DD DSN=PMMCO.UTCS.MOE&D8&DN2,UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT05F001 DD DSN=PMMCO.UTCS.CC(POST),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.LABS(&DN1),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.LABS(&DN2),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.NOBS(&DN1),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.NOBS(&DN2),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.CC(NUM),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.MLABS(M&D2),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.MLABS(M&D3),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.MLABS(M&D4),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.MLABS(M&D5),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.MLABS(M&D6),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.MLABS(M&D7),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
// DD DSN=PMMCO.UTCS.MLABS(M&D8),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR,LABEL=(,,,IN)
```

Figure 29. Second detector data process procedure POST (continued).

APPENDIX B

DATA COLLECTION PROCEDURES AND FORMS

DATA COLLECTION PROCEDURES AND FORMS

INTRODUCTION

This appendix provides details of the field data collection procedures and forms for the moving car studies and the bus priority studies. The information expands, as applicable, on that provided in the chapter on SITUATION AND APPROACH presented in Volume 1. In addition to the basic information provided herein, verbal instruction and field practice tests were used for the more complex tasks such as the intersection bus timing procedure and the use of the Greenshield's Traffic Analyzer.

MOVING CAR DATA COLLECTION

The following instructions were provided to the drivers of the vehicles equipped with the analyzers. Supplemental equipment and instructions were provided for the manual collection effort.

These instruction are designed to assist each driver in acquiring the best possible travel time data to be used in evaluating the UTCS/BPS traffic control system. Four routes have been selected for data collection in this project. Two routes are exclusively on primary arterials and are divided into two route notations by routes and random routes representing a loop-type design. The information received from the data that each of the drivers will collect while traveling on the routes will be utilized by the study team in the evaluation of the UTCS/BPS system.

Each automobile is equipped with a Greenshield's Traffic Systems Analyzer. Each analyzer records the following information; speed change, distance, and time.

It is anticipated that each driver will travel each route a minimum of four times for each of the three periods, that is, morning, midday, and evening. Each driver is required to follow the route exactly as it is laid out without variation. Before starting each run, the driver will perform the following tasks:

- Plug analyzer power cord in the cigarette lighter outlet.
- Check to be sure there is sufficient tape on the roll to complete route. Each roll should last approximately one week with normal use.

- The automatic tenth-of-a-mile print switch is turned off. The analyzer has been reset to zero and turned off.

- You have recorded on the paper tape the date, the route number, the direction, when applicable, the run number, and the starting time for each run.

- Driver will begin each run at the prescribed starting point.

- When the run is begun, the driver will turn the machine on as he crosses the starting point and depress the print button.

- After depressing the print button, the driver will advance the tape by hand two to three notches on the machine. This is to gap the print on the tape and identify each check point.

As a driver drives along the pre-defined route, the following instructions are to be followed:

1. At no time will the driver exceed the speed limit of 30 miles per hour unless he is in the middle of a moving platoon and it would be unsafe to reduce his speed. If the driver is driving on the street alone, he will not exceed 30 miles an hour.

2. While driving the route, whichever lane you have selected, each driver should remain in that lane and should avoid unnecessary lane change or weaving from lane to lane.

3. If you are caught behind a left-turning vehicle, for example, you will wait in that lane until this vehicle has cleared the intersection.

4. Each driver will depress the print button whenever he comes to a full stop, regardless of why. He will depress the print button at the point in which he starts moving again.

5. Each time you cross the stop bar or link discharge point as defined, the driver will depress the print button and manually advance the tape two or three notches to create a gap.

6. Since each delay entails two manual button depressions or two print-outs and the discharge point is recorded as one print-out, there will always be an odd number--for example, one, three, five, or nine--entries between each manual pull of the tape.

7. Each driver will continue pressing the print-out button at each discharge point and each time he stops and starts until the route is completed.

8. When the route has been completed, the driver will park in a safe place, manually pull the tape out a sufficient distance to separate each run, and perform the initial check points as previously described.

9. At the end of each time period, the tape is removed from the machine, folded neatly, and placed in an envelope on which will be recorded the date, the time period, and the route number with each driver initialing the envelope. These envelopes are to be turned in at the conclusion of each day to the field crew supervisor.

The drivers are to report each morning at 6:45 in order to be at his appropriate routing point to begin his initial run at 7:00 a.m. At the conclusion of the morning runs, each driver is to return the vehicle to the parking lot located at 3rd and Massachusetts Avenue, N. W., where the car will be parked and locked until 12:45, at which time he will secure the vehicle and prepare to begin the afternoon runs. At the conclusion of his last evening run, the driver will return the vehicle to the parking lot at 3rd and Massachusetts, N. W., park and lock the car, and store the keys inside the central control center. At no time is any driver to take the keys home with him.

Gasoline money for purchasing fuel for the autos will be provided and can be acquired from the field crew supervisor. The driver is responsible for refilling the gas tank as needed, being sure to get a receipt for the purchase of the gasoline. The receipt will be turned in to the field crew supervisor after the purchase.

The route breakdown and distance check list is shown on Figures 30 through 35. These were used during the driver training program, to check the daily runs, and to verify information during the coding of the data.

CHECK POINT NO.	INTERSECTION NAME	NUMBER	LINK DISTANCE	CUMULATIVE DISTANCE
1.	M STREET	1	0.11	0.11
2.	N STREET	2	0.05	0.16
3.	DUMBARTON STREET	3	0.02	0.18
4.	O STREET	4	0.09	0.27
5.	P STREET	5	0.09	0.36
6.	Q STREET, S.	6	0.03	0.39
7.	Q STREET, N.	7	0.21	0.60
8.	R STREET	8	0.14	0.74
9.	34TH STREET	9	0.08	0.82
10.	WHITEHAVEN PKWY.	10	0.015	0.835
11.	WHITEHAVEN STREET	11	0.17	1.005
12.	W PLACE	12	0.09	1.095
13.	HALL PLACE	13	0.18	1.275
14.	CALVERT STREET	14	0.15	1.425
15.	EDMUNDS STREET	15	0.19	1.615
16.	GARFIELD STREET	16	0.06	1.675
17.	MASSACHUSETTS	17	0.23	1.905
18.	WOODLEY ROAD	18	0.10	2.005
19.	MACOMB STREET	19	0.08	2.085
20.	NEWARK STREET	20	0.20	2.285
21.	PORTER STREET			

Figure 30. Wisconsin Avenue, northbound (Route 11).

CHECK POINT NO.	INTERSECTION NAME	NUMBER	LINK DISTANCE	CUMULATIVE NAME
1.	PORTER STREET	1	0.19	0.19
2.	NEWARK STREET	2	0.08	0.27
3.	MACOMB STREET	3	0.11	0.38
4.	WOODLEY ROAD	4	0.19	0.57
5.	CATHEDRAL AVENUE	5	0.02	0.59
6.	MASSACHUSETTS AVENUE	6	0.08	0.67
7.	GARFIELD STREET	7	0.19	0.86
8.	EDMUNDS STREET	8	0.14	1.00
9.	CALVERT STREET	9	0.18	1.18
10.	HALL PLACE	10	0.07	1.25
11.	W PLACE	11	0.18	1.43
12.	WHITEHAVEN STREET	12	0.02	1.45
13.	WHITEHAVEN PARKWAY	13	0.06	1.51
14.	34TH STREET	14	0.15	1.66
15.	R STREET	15	0.21	1.87
16.	Q STREET, N.	16	0.03	1.90
17.	Q STREET, S.	17	0.08	1.98
18.	P STREET	18	0.10	2.08
19.	O STREET	19	0.02	2.10
20.	DUMBARTON STREET	20	0.04	2.14
21.	N STREET	21	0.11	2.25
22.	M STREET			

Figure 31. Wisconsin Avenue, southbound (Route 13).

CHECK POINT NO.	INTERSECTION NAME	NUMBER	LINK DISTANCE	CUMULATIVE DISTANCE
1.	17TH STREET	1	0.14	0.14
2.	18TH STREET	2	0.07	0.21
3.	H STREET	3	0.03	0.24
4.	19TH STREET	4	0.09	0.33
5.	20TH STREET	5	0.10	0,43
6.	21ST STREET	6	0.12	0.55
7.	22ND STREET	7	0.035	0.585
8.	K STREET RAMP	8	0.04	0.625
9.	CIRC/N. HAMPSHIRE	9	0.10	0.725
10.	24TH STREET	10	0.10	0.825
11.	25TH STREET	11	0.10	0.925
12.	26TH STREET	12	0.13	1.055
13.	28TH STREET	13	0.06	1.115
14.	29TH STREET	14	0.06	1.175
15.	30TH STREET	15	0.11	1.285
16.	31ST AVENUE	16	0.09	1.375
17.	WISCONSIN AVENUE	17	0.18	1.555
18.	33RD STREET	18	0.10	1.655
19.	34TH STREET	19	0.05	1.705
20.	KEY BRIDGE			

Figure 32. M Street and Pennsylvania Avenue, westbound (Route 24).

CHECK POINT NO.	INTERSECTION NAME	NUMBER	LINK DISTANCE	CUMULATIVE DISTANCE
1.	KEY BRIDGE			
		1	0.05	0.05
2.	34TH STREET			
		2	0.09	0.14
3.	33RD STREET			
		3	0.18	0.32
4.	WISCONSIN AVENUE			
		4	0.08	0.40
5.	31ST STREET			
		5	0.11	0.51
6.	30TH STREET			
		6	0.05	0.56
7.	29TH STREET			
		7	0.06	0.62
8.	28TH STREET			
		8	0.14	0.76
9.	26TH STREET			
		9	0.09	0.85
10.	25TH STREET			
		10	0.11	0.96
11.	24TH STREET			
		11	0.03	0.99
12.	CIRCLE			
		12	0.06	1.05
13.	CIRCLE & 23RD			
		13	0.08	1.13
14.	22ND STREET			
		14	0.13	1.26
15.	21ST STREET			
		15	0.10	1.36
16.	20TH STREET			
		16	0.08	1.44
17.	19TH STREET			
		17	0.10	1.54
18.	18TH STREET			
		18	0.13	1.67
19.	17TH STREET			

Figure 33. M Street and Pennsylvania Avenue, eastbound (Route 22).

CHECK POINT NO.	INTERSECTION NAME	NUMBER	LINK DISTANCE	CUMULATIVE DISTANCE
1.	22ND & F STREET	1	0.06	0.06
2.	22ND & G STREET	2	0.25	0.31
3.	22ND & PA. AVENUE	3	0.035	0.345
4.	22ND & K STREET	4	0.10	0.445
5.	K & 21ST	5	0.10	0.545
6.	K & 20TH	6	0.08	0.625
7.	K & 19TH	7	0.10	0.725
8.	K & 18TH	8	0.12	0.845
9.	K & CONNECTICUT AVE.	9	0.05	0.895
10.	K & 17TH	10	0.10	0.995
11.	K & 16TH	11	0.11	1.105
12.	K & 15TH (VERMONT)	12	0.05	1.155
13.	K & 15TH	13	0.09	1.245
14.	K & 14TH	14	0.07	1.315
15.	14TH & I	15	0.08	1.395
16.	14TH & H	16	0.08	1.475
17.	H & 15TH	17	0.08	1.555
18.	H & MADISON	18	0.08	1.635
19.	H & 16TH	19	0.08	1.715
20.	H & JACKSON	20	0.08	1.795
21.	H & 17TH	21	0.08	1.875
22.	17TH & PA. AVENUE	22	0.07	1.945
23.	17TH & ENT. TO EXEC. OFF.BLDG.	23	0.04	1.985
24.	17TH & F	24	0.07	2.055
25.	17TH & NEW YORK	25	0.07	2.125
26.	17TH & E	26	0.21	2.335
27.	17TH & CONST. AVENUE	27	0.21	2.335
28.	18TH & VIRGINIA AVE.	28	0.19	2.655

Figure 34. Downtown loop (Route 30).

CHECK POINT NO.	INTERSECTION NAME	NUMBER	LINK DISTANCE	CUMULATIVE DISTANCE
29.	18TH & E STREET S.	29	0.04	2.695
30.	18TH & E STREET N.	30	0.10	2.795
31.	18TH & F	31	0.07	2.865
32.	18TH & G	32	0.08	2.945
33.	18TH & PA. AVENUE	33	0.05	2.995
34.	18TH & H	34	0.07	3.065
35.	18TH & I	35	0.07	3.135
36.	18TH & K	36	0.09	3.225
37.	18TH & L	37	0.08	3.305
38.	L & CONNECTICUT AVE.	38	0.07	3.375
39.	CONNECTICUT AVE. & K	39	0.09	3.465
40.	17TH & I	40	0.08	3.545
41.	17TH & H	41	0.08	3.625
42.	17TH & PA. AVENUE	42	0.0	3.695
43.	17TH & EXEC. OFF. BLDG.	43	0.04	3.735
44.	17TH & F	44	0.06	3.795
45.	N.Y., E. & 17TH ST.	45	0.11	3.905
46.	N.Y., E. & 18TH ST.	46	0.10	4.005
47.	E & 19TH STREET	47	0.08	4.085
48.	E & 20TH STREET	48	0.10	4.185
49.	E & 21ST STREET	49	0.07	4.255
50.	E & VIRGINIA AVENUE			

Figure 34. Downtown loop (Route 30). (Continued)

CHECK POINT NO.	INTERSECTION NAME	NUMBER	LINK DISTANCE	CUMULATIVE DISTANCE
1.	23RD & D	1	0.01	0.01
2.	23RD & NAVY HOSPITAL	2	0.03	0.04
3.	23RD & E	3	0.06	0.10
4.	23RD & VIRGINIA AVE.	4	0.02	0.12
5.	23RD & F	5	0.07	0.19
6.	23RD & G	6	0.08	0.27
7.	23RD & H	7	0.07	0.34
8.	23RD & I	8	0.09	0.43
9.	23RD & WASHINGTON CRCL	9	0.09	0.52
10.	WASH. CRCL & N. HAMP.	10	0.06	0.58
11.	N. HAMPSHIRE & L ST.	11	0.13	0.71
12.	L & 21ST STREET	12	0.09	0.80
13.	L & 20TH STREET	13	0.08	0.88
14.	L & 19TH STREET	14	0.10	0.98
15.	L & 18TH STREET	15	0.08	1.06
16.	L & CONN. AVENUE	16	0.08	1.14
17.	L & 17TH STREET	17	0.11	1.25
18.	L & 16TH STREET	18	0.10	1.35
19.	L & 15TH STREET	19	0.09	1.44
20.	L & VERMONT AVENUE	20	0.06	1.50
21.	L & 14TH STREET	21	0.07	1.57
22.	14TH & K	22	0.08	1.65
23.	K & VERMONT AVENUE	23	0.06	1.71
24.	K & 15TH STREET	24	0.10	1.81
25.	K & 16TH STREET	25	0.11	1.92
26.	K & 17TH STREET	26	0.05	1.97
27.	K & CONNECTICUT AVE.	27	0.12	2.09
28.	K & 18TH STREET	28	0.10	2.19

Figure 35. Downtown loop (Route 40).

CHECK POINT NO.	INTERSECTION NAME	NUMBER	LINK DISTANCE	CUMULATIVE DISTANCE
29.	K & 19TH STREET	29	0.08	2.27
30.	K & 20TH STREET	30	0.10	2.37
31.	K & 21ST STREET	31	0.13	2.50
32.	K & 22ND STREET	32	0.10	2.60
33.	K & 21ST STREET	33	0.09	2.69
34.	K & 20TH STREET	34	0.08	2.77
35.	K & 19TH STREET	35	0.07	2.84
36.	19TH & I STREET	36	0.08	2.92
37.	19TH & PA. AVENUE	37	0.13	3.05
38.	19TH & G STREET	38	0.07	3.12
39.	19TH & F STREET	39	0.09	3.21
40.	19TH & E STREET N.	40	0.04	3.25
41.	19TH & E STREET S.	41	0.12	3.37
42.	19TH & C STREET	42	0.10	3.47
43.	19TH & CONST. AVE.	43	0.07	3.54
44.	CONST. AVE. & 20TH ST.	44	0.10	3.64
45.	CONST. AVE. & 21ST ST.	45	0.07	3.71
46.	CONST. AVE. & BACON	46	0.04	3.75
47.	CONST. AVE. & 22ND ST.	47	0.07	3.82
48.	CONST. AVE. & 23RD ST.	48	0.08	3.90
49.	23RD & C STREET			

Figure 35. Downtown loop (Route 40). (Continued)

ENCODING MOVING CAR DATA

These instructions are for transferring machine tape data to the coding sheet for data processing. Sample forms are shown on Figure .

Column	First Car I. D.
2	Route identification number
3	Direction identification for Routes 1 and 2 (north 1, east 2, south 3, west 4)
4,5	Month of year (i.e. August = 08)
7,8	Day of the month (01 _ day _ 31)
10,11	Hour of day, based on 24-hour clock
13,14	Minute of hour (00 _ minute _ 59)
17	Driver number, same as route number or other designated number
20	For Comment Card--the number placed in this column designates the number of comment cards to follow

2. Second Card - Comment

Comment cards identify any unusual circumstances that affect traffic flow. Only short statements are needed (i.e. accident on K St. near 18th)

3. Third Card - Data

Each intersection is designated by a negative sign (-) before the intersection number (see each individual route travel time and delay listing)

Each intersection number is followed by the crossing time. This number recorded in accumulative seconds is the time at which the car crosses the stop bar. This point is known as link discharge point. The stop and starts (delay time) are recorded after the intersection number and the Crossing Time. The delay time is recorded in pairs so that every time entry (crossing time and delay time) is in odd groupings. Or in other words, the entries between intersection designation are odd (1, 3, 5..., see example).

When recording stop times, be sure that for every stop time recorded a comparison start is also recorded so that the entry for each link will show a discharge time and a combination of start and stop time.

MANUAL MOVING CAR STUDIES

As noted, certain initial runs and runs which occurred when equipment failed were made manually. The procedures for data collection were similar to those defined in the Manual of Traffic Engineering Studies, Third Edition, Institute of Traffic Engineers, 1964, pp. 41-44. Sample forms as used for one of the downtown loops are shown on Figure 36.

Bus Data Collection

Volume 1 described the bus studies and noted that both intersection and route analyses were conducted. The following instructions were used.

Intersection Analyses

Each person will be responsible for recording data for one approach. The data are to be used in the analysis of bus performance at individual locations. There are three intersections (nine approaches) to be analyzed in this process. They are:

1. 18th Street and Pennsylvania Avenue, N. W.
   Three approaches

2. Wisconsin Avenue and Macomb Street, N. W.
   Two approaches on Wisconsin Avenue

3. 14th Street and K Street, N. W.
   Four approaches

The recorder must record specific time information for as many buses as possible. The types of data desired preclude the ability to record this information for every bus passing his location.

The recorder is requested to get complete data for each bus timed. Selection of buses to be timed is to be made in this priority. All buses with the identifying symbol "B" in their windshield will have first priority. As many of the other buses are to be timed as is possible. Selection of buses shall be random with their position in a group to be variable.

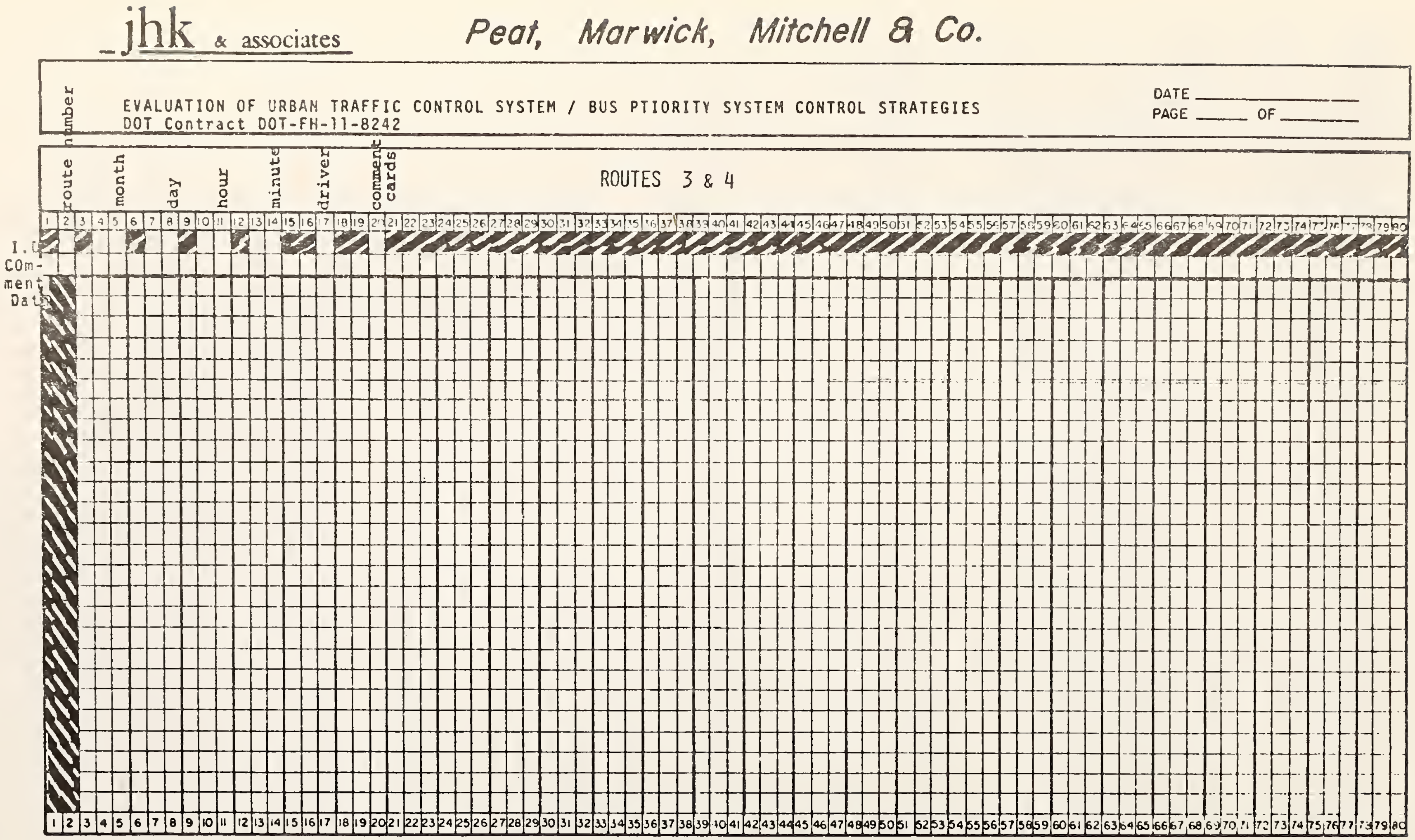

Figure 36. Sample data collection form.

As each bus selected crosses the detector location, the recorder will start the stop watch for recording his through time. He will also record all of the information shown on the attached form.

When the bus stops at the intersection, the second stop watch is to be used to time the dwell time of the bus. The recorder will also indicate reason for dwell using symbol shown on form. Departure time is time for bus to travel from the detector location across intersection. This time is recorded from first stop watch.

After each selected bus has cleared the intersection and the times have been recorded, the recorder will reset both watches and begin on the next bus.

Please be sure each bus timed has the complete information recorded.

The form used to record the intersection data is shown on Figure 37.

Route Analyses

Bus route data is to be collected for two routes in the UTCS/BPS area. The data collection periods are:

Wisconsin Avenue at Macomb Street

Station 1 - one person to be located on the west side of Wisconsin Avenue, north of Macomb at the advance detector point. Record all southbound buses as they cross detector location.

Station 4 - one person to be located on the east side of Wisconsin Avenue at the crosswalk. Record all northbound buses as they pass crosswalk.

Pennsylvania Avenue at 17th Street

Station 3 - one person to be located on the north side of Pennsylvania, east of 17th Street at the advance detector point. Record all buses in series 30 to 39 as they cross detector.

Station 2 - one person to be located on the south side of Pennsylvania Avenue at east of 17th Street at the crosswalk. Record all series 30-39 buses.

TRAVEL TIME AND DELAY

Route 3 - Down Town Loop Direction ______ Run number ______

Date ______ Day ______ Start time ______ Recorder ______

Control Point		Crossing Time	Link Time Delay Time
No.	Location		
30-28	18th & Virginia		
30-29	18th & E St. S.		
30-30	18th & E St. N.		
30-31	18th & F		
30-32	18th & G		
30-33	18th & Pa. Ave.		
30-34	18th & H		
30-35	18th & I		
30-36	18th & K		
30-37	18th & L		
30-38	L & Conn. Ave.		
30-39	Conn. Ave. & K		
30-40	17th & I		
30-41	17th & H		
30-42	17th & Pa. Ave.		
30-43	17th & Exec. Off.		
30-44	17th & F		
30-45	N.Y., E & 17th St.		
30-46	N.Y., E & 18th St.		
30-47	E & 19th Street		
30-48	E & 20th Street		
30-49	E & 21st Street		
30-50	E & Virginia Ave.		

CAUSE OF DELAY

S - Traffic Signals
LT - Left Turns
SS - Stop Sign
PK - Parked Cars
DP - Double Parking
T - General Congestion
PED - Pedestrian
BP - Bus Passengers

Figure 37.

TRAVEL TIME AND DELAY

Route 3 - Down Town Loop Direction ______ Run number ________

Date ____________ Day ________ Start time ____________ Recorder _____

Control Point		Crossing Time	Link Time Delay Time
No.	Location		
30-01	22nd & F Street		
30-02	22nd & G Street		
30-03	22nd & Penn. Ave.		
30-04	22nd & K Street		
30-05	K & 21st Streets		
30-06	K & 20th Streets		
30-07	K & 19th Streets		
30-08	K & 18th Streets		
30-09	K & Connecticut		
30-10	K & 17th Streets		
30-11	K & 16th Streets		
30-12	K & 15th (Vermont)		
30-13	K & 15th		
30-14	K & 14th		
30-15	14th & I		
30-16	14th & H		
30-17	H & 15th		
30-18	H & Madison		
30-19	H & 16th		
30-20	H & Jackson		
30-21	H & 17th		
30-22	17th & Penn. Ave.		
30-23	17th & Ent. to Exec. Off. Bldg.		
30-24	17th & F		
30-25	17th & New York		
30-26	17th & E		
30-27	17th & Const. Ave		

CAUSE OF DELAY

S - Traffic Signals
LT - Left Turns
SS - Stop Sign
PK - Parked Cars
DP - Double Parking
T - General Congestion
PED - Pedestrian
BP - Bus Passengers

Figure 37 (continued).

The data collectors at these four locations will be responsible for recording the information listed on the attached form for all buses whose route numbers are 30, 32, 33, 34, 36, 37, 38, and 39.

<u>21st Street at L Street</u>

Station 1 - one person on the west side of 21st Street, north of L Street at the advance detector points. Record all D2, D4, L4, and L6 buses on 21st Street.

<u>14th Street at K Street</u>

Station 2 - one person on the south side of K Street, east of 14th Street at the crosswalk. Record all eastboudn D2, D4, L4 and L6 buses as they pass crosswalk.

Station 3 - one person on the north side of K Street, east of 14th Street at the advance detector point. Record all D2, D4, L4 and L6 buses as they cross detector location.

<u>20th Street at L Street</u>

Station 4 - one person on the east side of 20th Street at the crosswalk. Record all northbound D2, D4, L4 and L6 buses.

Each data collector is to record all of the information shown on the sheets for each bus on the designated routes passing the intersection. The information is listed in order of its importance. See the attached sample form. Note: When recording arrival time, please be sure to record the time to the nearest second.

The station locations and routes are shown on Figures 38 and 39. The recording form is shown on Figure 40 and 41.

Figure 40. BPS Data collection points, Pennsylvania Avenue routes.

Figure 41. BPS Data collection points, Glover Park and Connecticut Avenue routes.

jhk & associates

BPS/INTERSECTION DATA

DATE ______________ DAY ______________ TIME PERIOD______________ RECORDER __________

LOCATION ____ DIRECTION _________________________

ARRIVAL TIME	ROUTE NO.	BLOCK NO.	B BUS ?	BUS NO.	STOP CAUSE	DWELL TIME	DEPARTURE TIME	EST. NO. PASS.
						: SEC.	: SEC.	

STOP CAUSE

O = NO STOPPING
P = PASSENGER LOADING ONLY
R = RED ONLY
PR = PASSENGER LOADING PLUS RED

Figure 40.

jhk & associates

UTCS/BPS # 2911

ROUTE ANALYSIS

DATE ________ DAY ________ TIME PERIOD ________ RECORDER ___

LOCATION

DIRECTION

ARRIVAL TIME	ROUTE NO.	BLOCK NO	B BUS ?	BUS NO.	EST. NO. PASS.	SCHEDULE TIME

Figure 41.

APPENDIX C

USER'S MANUAL

MOVING CAR ANALYSIS ROUTINES

USER'S MANUAL

MOVING CAR ANALYSIS ROUTINES

INTRODUCTION

Moving car (MC) data reduction and analysis programs were developed to handle data collected as part of the UTCS/BPS evaluation project. The software is not directly transferable to other situations, although it could be modified for usage elsewhere.

The underlying concept of the analysis is to develop link-specific measures of travel time, delay, stops, and speed from the moving car runs performed as part of the study. Following a series of edit checks, a network volume is added to each observation for purposes of ensuring that data, for comparative purposes, was collected under similar circumstances.

These network volumes are extracted from the UTCS 15-minute detector summary data and are the same used for the detector analysis carried out as part of the evaluation study. This step ties the procedure to the UTCS software, although an alternative mechanism for assigning network volumes or other global traffic performance measure could be made.

Once the volumes have been assigned, the observations are ranked in ascending volume sequence and the two sets of observations under comparison are matched, following the same procedure as used for the detector analysis. Briefly, this procedure is to first eliminate any "outliers" so that the two distributions span similar volume ranges, then eliminate selected points from the larger distribution to achieve equal sample sizes.

Following the matching process, a series of statistical post processor inputs are prepared and the UTCS post processor is run for the selected links. Subsequently, the links are aggregated into a series of subnetworks and the post processor is run again for these statistical comparisons. The post processor is the same as developed for the detector analysis and requires only the particular type of inputs derived from the MC runs for successful operation.

COMPONENTS

The MC chain is composed of seven special-purpose programs, plus the post processor and the system sort routines. As the procedure is currently applied, two of the programs are executed once for each MC route/time-of-day data set. Three of the programs and the sort are run on each of the data sets for a specific statistical comparison. The remaining two programs plus two executions of the post processor are made once for each comparison.

The procedure is illustrated in Figure 42. For simplicity, the set-up indicates a single application from raw data through the post processor runs. In actual practice, the MC2 and MCPLUG2 programs are run only once for a route/time-of-day data set. The remaining programs are executed together for each comparison. Briefly, the individual program performs the following functions:

<u>MC2</u>

1. Performs the basic data reduction and computation of measures of effectiveness (MOE);

2. Performs data checks for editing the raw data;

3. Flags partially-usable observations.

<u>PLUG2</u>

1. Substitutes mean values for unusable items in an observation.

<u>STRIP</u>

1. Pre-arranges the MOE data set for post processor input;

2. Extracts a header record for each observation.

<u>MCVOL2</u>

1. Assigns a network volume from the detector data to each observation (if possible).

<u>SORT</u>

1. Sorts the header records by ascending volume.

<u>MCMTCH2</u>

1. Matches the two header records.

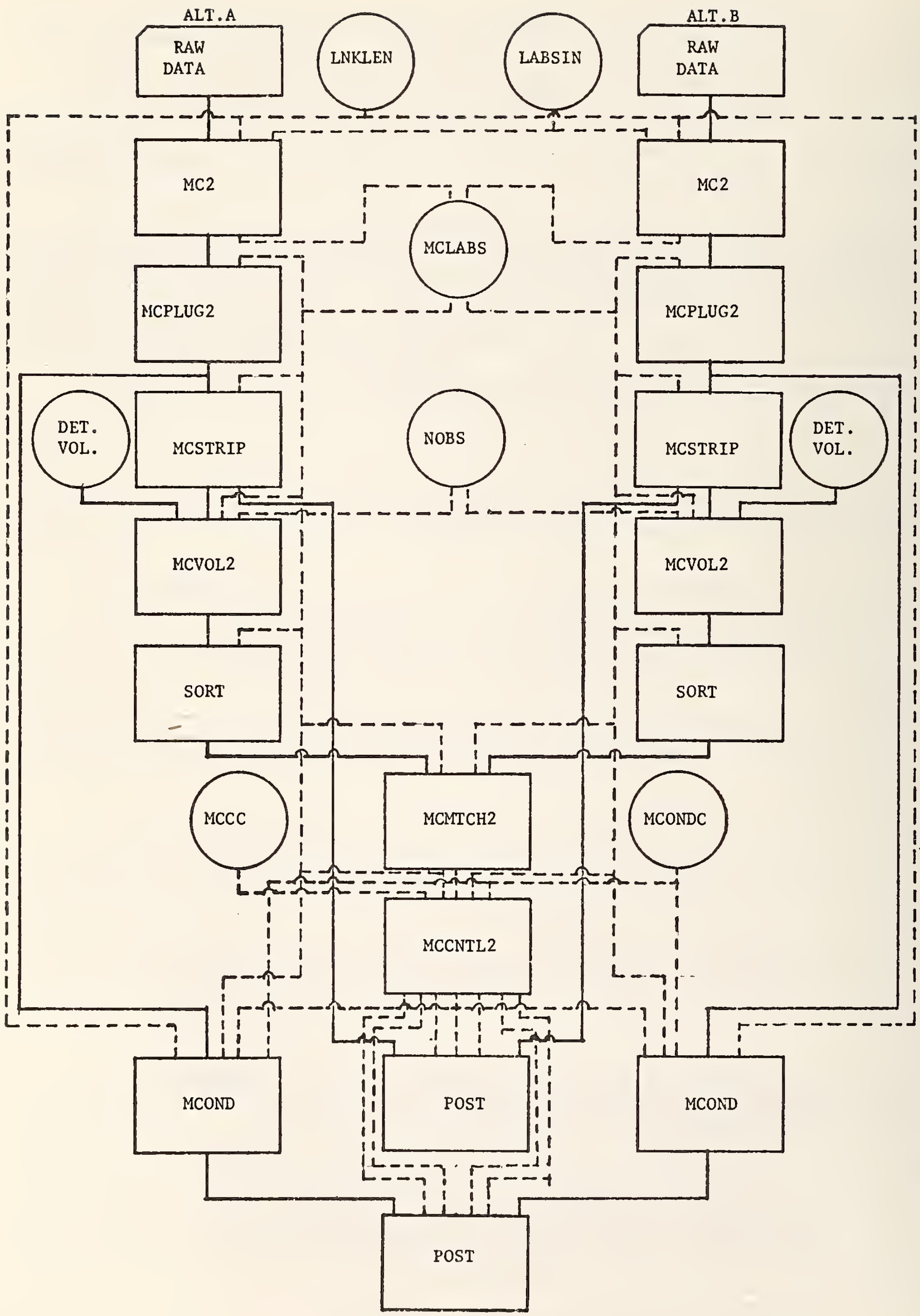

Figure 42. Flow chart of moving car analysis.

MCCNTL2

1. Produces control inputs for post processor runs.

MCOND

1. Computes aggregate MOE's for subnetworks.

POST

1. Performs statistical analyses on link and subnetwork data sets.

OPERATIONAL PROCEDURE

For ease of application, the MC programs have been written so that they may be executed using catalogued procedures. All control is exercised through special-purpose data sets with unique names (or member names) that can be altered at execution time by the passing of parameters through the PROC. The MC2 and MCPLUG2 programs are incorporated in a simple PROC and the rest of the procedure (14 separate program executions) are included as a second large PROC. Figures 43 and 44 indicate the programs for the two procedures.

The use of PROCs requires the creation of several special-purpose control data sets for passing parameters and other control values to the various programs. Some of these data sets, particularly those that vary only by route, exist as unique data sets with names containing substitutable parameters. Most of the other data sets, however, are of the partitioned data set (PDS) form, having separate members for each route, time of day, and alternative (RTA). These various data sets include the following:

LNKLEN

Content: length of each link on a route, in hundredths of a mile

Organization: sequential data set, one per route.

Form: BCD (card image).

Format: (10I6).

LABSIN

Content: major parameters for each RTA.

```
//MCES PROC DSN=XXX,RTNC=YYY
//OUT EXEC PGM=IEFBR14
//BYE DD DSN=PMMCO.UTCS.MCAR.MO&DSN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(OLD,DELETE,DELETE)
//INITMC EXEC PGM=MC2,TIME=1
//STEPLIB DD DSN=PMMCO.UTCS.LOAD,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT03F001 DD SYSOUT=A
//FT06F001 DD SYSOUT=A
//FT05F001 DD DSN=PMMCO.UTCS.LABSIN(&DSN),LABEL=(,,,IN),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT15F001 DD DSN=PMMCO.UTCS.MCAR.LNKLEN.R&RTNO,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT08F001 DD DSN=PMMCO.UTCS.MCAR.RD&DSN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT09F001 DD DSN=&&TEMP,UNIT=SYSDA,
// DCB=(RECFM=VBS,LRECL=988,BLKSIZE=6447),SPACE=(6447,(6,1),RLSE),
// DISP=(NEW,PASS)
//FT10F001 DD DSN=PMMCO.UTCS.MCLABS(&DSN),LABEL=(,,,OUT),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=OLD,
// DCB=(RECFM=VBS,LRECL=28,BLKSIZE=32),SPACE=(TRK,(2,1,20))
//PLUGMC EXEC PGM=PLUG,TIME=1
//STEPLIB DD DSN=PMMCO.UTCS.LOAD,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT03F001 DD SYSOUT=A
//FT06F001 DD SYSOUT=A
//FT08F001 DD DSN=&&TEMP,UNIT=SYSDA,DISP=(OLD,DELETE)
//FT09F001 DD DSN=PMMCO.UTCS.MCAR.MO&DSN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=(NEW,KEEP,DELETE),
// DCB=(RECFM=VBS,LRECL=1228,BLKSIZE=6447),SPACE=(6447,(6,1),RLSE)
//FT05F001 DD DSN=PMMCO.UTCS.MCLABS(&DSN),LABEL=(,,,IN),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
```

Figure 43. MC PROC 1.

```
//MCSTRIP PROC DSN=X,DSN2=Y,DAB=XX,DBA=YY,DETA=XXX,DETB=YYY,RTE=Z
//UN1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD DSN=PMMCO.UTCS.MCLABS(&DSN),LABEL=(,,,IN),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//SYSUT2 DD DSN=&&MCLAB1,UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=28,BLKSIZE=32),SPACE=(TRK,(2,1))
//UN2 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD DSN=PMMCO.UTCS.MCLABS(&DSN2),LABEL=(,,,IN),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//SYSUT2 DD DSN=&&MCLAB2,UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=28,BLKSIZE=32),SPACE=(TRK,(2,1))
//STRIP EXEC PGM=STRIP
//STEPLIB DD DSN=PMMCO.UTCS.LOAD,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT03F001 DD SYSOUT=A
//FT08F001 DD DSN=PMMCO.UTCS.MCAR.MO&DSN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT09F001 DD UNIT=SYSDA,
// DCB=(RECFM=VBS,LRECL=252,BLKSIZE=1500),SPACE=(TRK,(5,1))
//FT10F001 DD UNIT=SYSDA,
// DCB=(RECFM=VBS,LRECL=252,BLKSIZE=1500),SPACE=(TRK,(5,1))
//FT11F001 DD UNIT=SYSDA,
// DCB=(RECFM=VBS,LRECL=252,BLKSIZE=1500),SPACE=(TRK,(5,1))
//FT12F001 DD UNIT=SYSDA,
// DCB=(RECFM=VBS,LRECL=252,BLKSIZE=1500),SPACE=(TRK,(5,1))
//FT14F001 DD DSN=&&MCTH1,
// UNIT=SYSDA,DISP=(NEW,PASS),
// DCB=(RECFM=VBS,LRECL=36,BLKSIZE=1000),SPACE=(TRK,(1,1))
//FT16F001 DD DSN=&&MCTF1,
// UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=252,BLKSIZE=3520),SPACE=(TRK,(4,1),RLSE)
//FT18F001 DD DSN=&&MCTM1,
// UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=252,BLKSIZE=3520),SPACE=(TRK,(20,1),RLSE)
//FT05F001 DD DSN=&&MCLAB1,UNIT=SYSDA,DISP=(OLD,PASS)
```

Figure 44. MC PROC 2.

```
//STRIP2 EXEC PGM=STRIP
//STEPLIB DD DSN=PMMCO.UTCS.LOAD,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT03F001 DD SYSOUT=A
//FT08F001 DD DSN=PMMCO.UTCS.MCAR.MO&DSN2,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT09F001 DD UNIT=SYSDA,
// DCB=(RECFM=VBS,LRECL=252,BLKSIZE=1500),SPACE=(TRK,(5,1))
//FT10F001 DD UNIT=SYSDA,
// DCB=(RECFM=VBS,LRECL=252,BLKSIZE=1500),SPACE=(TRK,(5,1))
//FT11F001 DD UNIT=SYSDA,
// DCB=(RECFM=VBS,LRECL=252,BLKSIZE=1500),SPACE=(TRK,(5,1))
//FT12F001 DD UNIT=SYSDA,
// DCB=(RECFM=VBS,LRECL=252,BLKSIZE=1500),SPACE=(TRK,(5,1))
//FT14F001 DD DSN=&&MCTH2,
// UNIT=SYSDA,DISP=(NEW,PASS),
// DCB=(RECFM=VBS,LRECL=36,BLKSIZE=1000),SPACE=(TRK,(1,1))
//FT16F001 DD DSN=&&MCTF2,
// UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=252,BLKSIZE=3520),SPACE=(TRK,(4,1),RLSE)
//FT18F001 DD DSN=&&MCTM2,
// UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=252,BLKSIZE=3520),SPACE=(TRK,(20,1),RLSE)
//FT05F001 DD DSN=&&MCLAB2,UNIT=SYSDA,DISP=(OLD,PASS)
//MCVOL EXEC PGM=MCVOL2,COND=(4,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT03F001 DD SYSOUT=A
//FT06F001 DD SYSOUT=A
//FT08F001 DD DSN=PMMCO.UTCS.SRT3&DAB,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT10F001 DD DSN=&&MCTH1,
// UNIT=SYSDA,DISP=(OLD,PASS)
//FT09F001 DD DSN=&&MCTV1,
// UNIT=SYSDA,DCB=(RECFM=VBS,LRECL=40,BLKSIZE=1000),
// DISP=(NEW,PASS),SPACE=(TRK,(4,1),RLSE)
//FT05F001 DD DSN=&&MCLAB1,UNIT=SYSDA,DISP=(OLD,PASS)
//FT15F001 DD DSN=PMMCO.UTCS.NOBS(&DETA),LABEL=(,,,IN),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//MCVOL2 EXEC PGM=MCVOL2,COND=(4,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT03F001 DD SYSOUT=A
//FT06F001 DD SYSOUT=A
//FT08F001 DD DSN=PMMCO.UTCS.SRT3&DBA,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT10F001 DD DSN=&&MCTH2,
// UNIT=SYSDA,DISP=(OLD,PASS)
//FT09F001 DD DSN=&&MCTV2,
// UNIT=SYSDA,DCB=(RECFM=VBS,LRECL=40,BLKSIZE=1000),
// DISP=(NEW,PASS),SPACE=(TRK,(4,1),RLSE)
//FT05F001 DD DSN=&&MCLAB2,UNIT=SYSDA,DISP=(OLD,PASS)
//FT15F001 DD DSN=PMMCO.UTCS.NOBS(&DETB),LABEL=(,,,IN),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
```

Figure 44. MC PROC 2 (continued).

```
//SORT EXEC PGM=SORT,PARM='MSG=AP,CORE=80000',COND=EVEN
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(1),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(1),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(1),,CONTIG)
//SORTIN DD DSN=&&MCTV1,
// UNIT=SYSDA,DISP=(OLD,PASS)
//SORTOUT DD DSN=&&MCTS1,
// UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=40,BLKSIZE=1000),SPACE=(TRK,(1,1))
//SYSIN DD DSN=PMMCO.UTCS.CC(SORT5),UNIT=SYSDA,VOL=SER=OTHERS,
// DISP=SHR,LABEL=(,,,IN)
//SORT2 EXEC PGM=SORT,PARM='MSG=AP,CORE=80000',COND=EVEN
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(1),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(1),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(1),,CONTIG)
//SORTIN DD DSN=&&MCTV2,
// UNIT=SYSDA,DISP=(OLD,PASS)
//SORTOUT DD DSN=&&MCTS2,
// UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=40,BLKSIZE=1000),SPACE=(TRK,(1,1))
//SYSIN DD DSN=PMMCO.UTCS.CC(SORT5),UNIT=SYSDA,VOL=SER=OTHERS,
// DISP=SHR,LABEL=(,,,IN)
//MTCH2 EXEC PGM=MCMTCH2,COND=(0,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT03F001 DD SYSOUT=A
//FT08F001 DD DSN=&&MCTS1,UNIT=SYSDA,DISP=(OLD,PASS)
//FT10F001 DD DSN=&&MCTS2,UNIT=SYSDA,DISP=(OLD,PASS)
//FT09F001 DD DSN=&&MCMTCH,UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=404,BLKSIZE=408),SPACE=(TRK,(1,1))
//FT05F001 DD DSN=&&MCLAB1,UNIT=SYSDA,DISP=(OLD,PASS)
// DD DSN=&&MCLAB2,UNIT=SYSDA,DISP=(OLD,PASS)
//CNTL2 EXEC PGM=MCCNTL2,COND=(0,LT)
//STEPLIB DD DSN=PMMCO.UTCS.LOAD,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT03F001 DD SYSOUT=A
//FT08F001 DD DSN=&&MCMTCH,UNIT=SYSDA,DISP=(OLD,PASS)
//FT11F001 DD DSN=&&MCL12,UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=404,BLKSIZE=408),SPACE=(TRK,(1,1))
//FT10F001 DD DSN=PMMCO.UTCS.MCOND.NUMOUT&RTE,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT09F001 DD DSN=&&MCO12,UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=404,BLKSIZE=408),SPACE=(TRK,(1,1))
//FT05F001 DD DSN=&&MCLAB1,UNIT=SYSDA,DISP=(OLD,PASS)
// DD DSN=&&MCLAB2,UNIT=SYSDA,DISP=(OLD,PASS)
//FT15F001 DD DSN=PMMCO.UTCS.MCCC(R&RTE),LABEL=(,,,IN),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT13F001 DD DSN=&&PCARDS,UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),SPACE=(800,(5,1))
//FT17F001 DD DSN=&&ACARDS,UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),SPACE=(800,(5,1))
//FT19F001 DD DSN=&&LNKLAB,UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80),SPACE=(80,(1,1))
//FT21F001 DD DSN=&&LAG,UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=124,BLKSIZE=128),SPACE=(128,(1,1))
```

Figure 44. MC PROC 2 (continued).

```
//POST EXEC PGM=POSTPROC
//STEPLIB DD DSN=PMMCO.UTCS.LOAD2,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT03F001 DD SYSOUT=A
//FT12F001 DD DSN=&&MCL12,UNIT=SYSDA,DISP=(OLD,PASS)
//FT13F001 DD DSN=&&MCO12,UNIT=SYSDA,DISP=(OLD,PASS)
//FT15F001 DD DSN=&&MCTM1,UNIT=SYSDA,DISP=(OLD,PASS)
//FT16F001 DD DSN=&&MCTM2,UNIT=SYSDA,DISP=(OLD,PASS)
//FT05F001 DD DSN=&&PCARDS,UNIT=SYSDA,DISP=(OLD,PASS)
//COND EXEC PGM=MCOND,TIME=1
//STEPLIB DD DSN=PMMCO.UTCS.LOAD,UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT03F001 DD SYSOUT=A
//FT08F001 DD DSN=PMMCO.UTCS.MCAR.MO&DSN,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT09F001 DD DSN=&&MCMAG1,UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=252,BLKSIZE=3520),SPACE=(TRK,(20,1))
//FT10F001 DD DSN=&&MCL12,UNIT=SYSDA,DISP=(OLD,PASS)
//FT12F001 DD DSN=PMMCO.UTCS.MCOND.NUMOUT&RTE,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT15F001 DD DSN=PMMCO.UTCS.MCAR.LNKLEN.R&RTE,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT05F001 DD DSN=PMMCO.UTCS.MCLABS(&DSN),LABEL=(,,,IN),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//COND2 EXEC PGM=MCOND,TIME=1
//STEPLIB DD DSN=PMMCO.UTCS.LOAD,UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT03F001 DD SYSOUT=A
//FT08F001 DD DSN=PMMCO.UTCS.MCAR.MO&DSN2,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT09F001 DD DSN=&&MCMAG2,UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=252,BLKSIZE=3520),SPACE=(TRK,(20,1))
//FT10F001 DD DSN=&&MCL12,UNIT=SYSDA,DISP=(OLD,PASS)
//FT12F001 DD DSN=PMMCO.UTCS.MCOND.NUMOUT&RTE,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT15F001 DD DSN=PMMCO.UTCS.MCAR.LNKLEN.R&RTE,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT05F001 DD DSN=PMMCO.UTCS.MCLABS(&DSN2),LABEL=(,,,IN),
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//POSTAG EXEC PGM=POSTPROC
//STEPLIB DD DSN=PMMCO.UTCS.LOAD2,
// UNIT=SYSDA,VOL=SER=OTHERS,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT03F001 DD SYSOUT=A
//FT11F001 DD DSN=&&LNKLAB,UNIT=SYSDA,DISP=(OLD,PASS)
//FT12F001 DD DSN=&&LAG,UNIT=SYSDA,DISP=(OLD,PASS)
//FT13F001 DD DSN=&&MCO12,UNIT=SYSDA,DISP=(OLD,PASS)
//FT15F001 DD DSN=&&MCMAG1,UNIT=SYSDA,DISP=(OLD,PASS)
//FT16F001 DD DSN=&&MCMAG2,UNIT=SYSDA,DISP=(OLD,PASS)
//FT05F001 DD DSN=&&ACARDS,UNIT=SYSDA,DISP=(OLD,PASS)
```

Figure 44. MC PROC 2 (continued).

. words 1-3: alphanumeric label

. word 4: blank (for number of observations)

. word 5: number of links

. word 6: printing switch

- 0: not print
- 2: print requested.

Organization: PDS, one member per RTA.

Form: binary integer.

Note: members of this data set are created in advance by a special-purpose routine that simply converts card input values of the parameters.

MCLABS

Content: major parameters for each RTA; as LABSIN with word 4 (number of observations filled in).

Organization: PDS, one member per RTA.

Form: binary integer.

NOBS

Content: number of observations in each detector data set.

Organization: PDS, one member per time of day/alternative (TA).

Form: BCD.

Note: same data set as used extensively in detector processing.

MCCC

Content: control cards for MCCNTL2; passed through as control cards to POST and MCOND.

Organization: PDS, one member per route.

Form: BCD.

Format: see MCCNTL2 program description.

MCONDCC

Content: control cards for MCOND.

Organization: sequential, one data set per route.

Form: BCD

Format: see MCOND program description

In addition to the above control card data sets, raw data as card images are supplied to MC2 and detector volume are supplied to MCVOL2. Detailed descriptions of these files are included in the respective program descriptions.

PROGRAM DESCRIPTIONS

The following program descriptions outline the basic operation of each of the MC analysis programs. Each program description includes a brief outline of the program operating procedure; inputs required; outputs produced; and size limitations. References are made to each of the control data sets discussed above.

MOE Computation Program - MC2

Operation

MC2 begins by transferring the raw card image data into a single variable length table. The recorded times are incremented whenever the cumulative values pass an additional 1000. The program then performs certain basic error checking computations, including determination of appropriate sequencing of link identifiers. It then ensures that an odd number of values has been recorded between each link identifier.

If all error checks are passed, the program then computes the following MOEs for all links not flagged as unusable:

- total travel time (seconds);
- stopped time delay (seconds);
- number of stops;
- average speed (mph).

The last measure is computed from total travel time and link length. An approximate route total is computed and the results printed and an output data set produced.

The values recorded on the raw data cards indicate the time at which one of the following occurred:

. vehicle entered a link (left the previous link);

. vehicle stopped in traffic or for a signal;

. vehicle resumed action motion after a stop.

Thus, if a vehicle passes through a link without stopping, only one time is recorded between two link identifiers. For each stop within a link, a pair of times must be recorded. Hence, the error check noted above is performed to ensure that an odd number of values is recorded.

Inputs

1. Identification: LABSIN control card (C2)

   FORTRAN unit: 5

2. Identification: LNKLEN control cards (C1)

   FORTRAN unit: 15

3. Identification: raw data

   FORTRAN unit: 8

   Content: for each observation, the following cards are supplied:

   a. type: header

      number: 1

      format: (I3, I2, 5I3)

      fields: 1 - route identifier (11, 13, 22, 24, 30, or 40)
      2 - month
      3 - day
      4 - hour (24-hour clock)
      5 - minute
      6 - driver (arbitrary code)
      7 - number of comment cards

b. type: comments

number: as field 7 of header card (may be 0)

format: (2X,26I3)

fields: link identifier coded as a negative integer, followed by an odd number of cumulative times. Bad data flagged with a -99.

Outputs

1. Identification: MCLABS control card (C3)

FORTRAN unit: 10

Content: as LABSIN with number of observations supplied by counting for the current run

2. Identification: MOE output data set

FORTRAN unit: 9

Structure: binary integer full words

Number: 1 record per observation

Content: first 6 words are the header (comment cards omitted) succeeding sets of 4 words are MOEs for each link

Limitations

1. Maximum number of links: 60

2. Maximum number of entries on raw data cards per observation: 250

Bad Value Substitution Program - MCPLUG2

Operation

MCPLUG2 computes mean values for all usable observations on each MOE and for each link. These values are then substituted for any values flagged previously by a -99.

Inputs

1. Identification: MCLABS control card (C3)

FORTRAN unit: 5

2. Identification: MOE data from MC2

FORTRAN unit: 8

Number: 1 record per observation

Content: first 6 words are the header, succeeding sets of 5 words are the MOEs and a flag for each link. The flag has a value of 1 if the data is original and a value of 3 if mean values have been substituted.

Limitations

1. Maximum number of links: 60

Data Rearrangement Program - MCSTRIP

Operation

MCSTRIP extracts the header from the basic data set and creates a separate header record file for subsequent processing. The MOEs are rearranged so that the output data set contains observations across all links for each MOE. This is the data set organization required by the post processor.

Inputs

1. Identification: MCLABS control card (C3)

FORTRAN unit: 5

2. Identification: MOE updated values

FORTRAN unit: 8

Scratch Data Sets

1. Identification: travel time scratch data set

FORTRAN unit: 9

2. Identification: delay scratch data set

FORTRAN unit: 10

3. Identification: stops scratch data set

FORTRAN unit: 11

4. Identification: speed scratch data set

FORTRAN unit: 12

Outputs

1. Identification: header card

FORTRAN unit: 14

Number: 1 record for each observation

Structure: binary integer full words

Content: word 1 - sequence number

word 2 - route identifier

word 3 - month

word 4 - day

word 5 - hour

word 6 - minute

word 7 - driver

word 8 - duration of run (seconds)

2. Identification: MOE flagged values

FORTRAN unit: 16

Number: 1 record for each observation

Structure: binary integer full words

Content: sequence number, followed by value of flag for each link

Note: not currently used in subsequent processing

3. Identification: MOE values for post processor

FORTRAN unit: 18

Number: 1 record for each observation for each of 4 MOEs

Structure: binary integer full words

Content: sequence number, followed by the value of an MOE for each link. All observations for the travel time MOE are written first, followed by delay, stops, and speeds, in that order.

Limitations

1. Maximum number of links: 60

Volume Assignment Program - MCVOL2

Operation

MCVOL2 assigns a network volume to each MC observation based upon the starting time and duration of the MC run. The program first converts the detector header information into a date and time configuration suitable for matching. It next builds a table of network volumes for each date and 15-minute time period. Missing values are estimated by taking averages across all dates for a specific 15-minute period. Values are then estimated for each date for the 15-minute periods immediately before and after the time covered. These values are estimated in the following way:

$$t_{-1} = t_0 - (t_1 - t_0) = 2t_0 - t_1$$

$$t_{n+1} = t_n + (t_n - t_{n-1}) = 2t_n - t_{n-1}$$

The network volume for the 15-minute summary nearest the starting time of the run is selected. If the duration of the run is greater than 15 minutes, an average volume over successive 15-minute intervals is used. The network volume is added to the header record and updated records printed and written out. If a match is not found, an appropriate error message is generated and the volume field is set to zero.

Input

1. Identification: MCLABS control card (C3)

   FORTRAN unit: 5

2. Identification: detector observations - NOBS control

card (C4)

FORTRAN unit: 15

3. Identification: header card from MCSTRIP

FORTRAN unit: 10

4. Identification: detector volume

FORTRAN unit: 8

Content: word 1 - identification number

word 2 - month (e.g. bMAR)

word 3 - day (e.g. b04b)

word 4 - year (e.g. 74bb)

word 5 - time (e.g. 0745)

word 6 - volume

Structure: word 1 and word 6 - binary integer full words;
word 2 - word 5 - binary character data as above.

<u>Output</u>

1. Identification: updated header record

FORTRAN unit: 9

Number: 1 record for each observation

Structure: binary integer full words

Content: similar to output of MCSTRIP with volume in word 9

<u>Limitations</u>

1. Maximum number of 15-minute detector volume summaries: 120

2. Maximum number of dates: 20

3. Maximum number of 15-minute periods per day: 15 (including estimated periods)

System Sort/Merge Routine - SORT

Operation

The system sort/merge routine is invoked to sort the header records by ascending order on volume.

Inputs

1. Identification: updated header records from MCVOL2

   Data set: SORTIN

2. Identification: sort/merge control card

   Data set: SYSIN

   Content: 1 control card, identical for all runs, as:

```
SORT FIELDS=(37,4,CH,A),SIZE=E100
where 37 = location in bytes of the beginning
           of the sort field (computed as
           word 8 x 4 bytes/word + 4 bytes
           for VBS records + 1 byte for be-
           ginning of word 9)
where  4 = length of word 9 in bytes
where CH = type of data (BI would produce
           identical results)
where  A = ascending sort desired
 "  E100 = estimated number of records (upper
           limit)
```

Scratch Data Sets

1. Identifications: 3 system sort/work areas

   Data sets: SORTWK01, SORTWK02, SORTWK03

Output

1. Identification: sorted header records

   Data set: SORTOUT

   Content: identical to SORTIN

Volume Matching Program - MCMTCH2

Operation

MCMTCH2 reads in two sorted header record data sets. An observation usage (OBUSE) vector is constructed, containing initially values of 1 for all observations except those with a zero volume (unmatched) for which the OBUSE value is set to 0, indicating an observation that is not to be used. The same matching subroutine used in the detector data processing is then called.

The matching routine determines if the difference between the outliers of the two distributions exceeds the following value:

$$v = \frac{(\max(MAXVOL1, MAXVOL2) - \min(MINVOL1, MINVOL2))}{\min(NOBS1, NOBS2)}$$

If the point is eliminated, the criteria is applied to the resulting outlier pair recursively until no further points are removed. Values closest to the mean are then written as a control data set to subsequent programs and a summary of the matched observations is printed.

Inputs

1. Identification: MCLABS control cards (CS)

   FORTRAN unit: 5

   Number: 2 records, one for each alternative, concatenated

2. Identification: sorted header records for alternative 1

   FORTRAN unit: 8

3. Identification: sorted header records for alternative 2

   FORTRAN unit: 10

Output

1. Identification: OBUSE vector

   FORTRAN unit: 9

   Number: 1 record

Structure: binary integer half words

Content: values of 0 or 1 for each observation in alternative 1, followed by values for each observation in alternative 2

Limitations

1. Maximum number of observations in either alternative: 100

Control Card Processing Program - MCCNTL2

Operation

MCCNTL2 is designed to produce control card inputs for the post processor runs for both link statistical analyses and aggregate subnetwork analyses. Some control cards are also produced for the subnetwork aggregation program MCOND. The program does not perform any significant analytical tasks.

Inputs

1. Identification: MCLABS control cards

   FORTRAN unit: 5

   Number: 2 records, one for each alternative, concatenated

2. Identification: MCONDC control cards

   FORTRAN unit: 10

   Number: 2 records used, plus other skipped

   Content: see MCOND writeup for description; number of aggregate subnetworks and subnetwork identifier labels are read

3. Identification: MCCNTL parameters - MCCC Control cards #1 (C5)

   FORTRAN unit: 15

   Number: 1 card

   Content: FORTRAN NAMELIST containing the following parameters:

NMOE - number of MOEs (normally set to 4)
NX01 - number of observations in alternative 1 to be eliminated (OBUSE to be set to 0 exogenously); assumed value 0
NX02 - number of observations in alternative 2 to be eliminated; assumed value 0.

NXL - number of links to be eliminated (value of link use vector LUSE to be set to 0 - normally identical for all runs for a given route); assumed value 0.

Note: this card is read as FORTRAN NAMELIST named PARAM and should be coded as:

&PARAM NMOE=4,NXL=3 &END

4. Identification: post processor control card for link statistical analysis - MCCC control cards #2 (C5)

FORTRAN unit: 15 - concatenated to (c)

Number: 1 card

Content: FORTRAN NAMELIST for the post processor, coded as:

&PARAM LNKNT=1,TESTBS=2,TWIND=2,NMOE=4 &END

The values of these parameters are described in the post processor program description.

Note: this card is not read as a NAMELIST but is passed to the post processor as a card image

5. Identification: post processor control card for sub-network statistical analysis - MCCC control cards #3 (C5)

FORTRAN unit: 15 - concatenated to (d)

Number: 1 card

Content: FORTRAN NAMELIST for the post processor, coded as:

&PARAM LNKNT=2,TESTBS=2,TWIND=2,NMOE=4 &END

Note: see (d)

6. Identification: MOE labels - MCCC control cards #4 (C5)

FORTRAN unit: 15 - concatenated to (e)

Number: NMOE/4 (1 for the current application)

Content: 20 columns of alphanumeric data identifying such MOE; up to 4 MOEs may be described on each card

Format: (4(5A4)

7. Identification: observation deletions for alternative 1 - MCCC control cards #5 (C5)

FORTRAN unit: 15 - concatenated to (f)

Number: NX01/20 (not supplied if NX01=0)

Content: observation numbers for alternative 1 to be deleted; i.e., OBUSE value to be set to 0

Format: (20I4)

8. Identification: observation deletions for alternative 2 - MCCC control cards #6 (C5)

FORTRAN unit: 15 - concatenated to (g)

Number: NX02/20 (not supplied if NX02=0)

Content: observation numbers for alternative 2 to be deleted

Format: (20I4)

9. Identification: link deletions

FORTRAN unit: 15 - concatenated to (h)

Number: NXL/20 (not supplied if NXL=0)

Content: link numbers to be deleted; i.e., LUSE values set to 0

Format: (20I4)

10. Identification: OBUSE table from MCMTCH2

FORTRAN unit: 8

Note: In the current operation, the MCCC control cards have been created as a PDS with one member for each route.

Outputs

1. Identification: updated OBUSE vector

   FORTRAN unit: 9

   Content: same form as input OBUSE vector; identical if NX01=NX02=0

2. Identification: link use (LUSE) vector

   FORTRAN unit: 1

   Number: 1 record

   Content: values of 0 or 1 for each link, with 1 indicating that a link will be used in the link level statistical analysis and may be aggregated in MCOND

   Structure: binary integer half words

3. Identification: subnetwork use vector

   Fortran unit number: 21

   Number: 1 record

   Content: all aggregate subnetworks are presumed to be usable in the subnetwork post processor statistical analysis and thus all elements of this vector have a value of 1

   Structure: binary integer half words

4. Identification: aggregate subnetwork identifier labels

   FORTRAN unit: 19

   Number: as required, with 18 values per record

   Content: 4 character alphanumeric labels for each aggregate subnetwork

   Structure: BCD card images

   Format: (18A4)

5. Identification: link level post processor control cards

FORTRAN unit: 13

Number: as required (7 in current application)

Structure: BCD card images

Content: the following card images are written:

card1: post processor control card - input data set (d)

card2: label for alternative 1 - from MCLABS

card3: label for alternative 2 - from MCLABS

card4: number of observations for alternative 1 - from MCLABS

card5: number of observations for alternative 2 - from MCLABS

card6: number of links - from MCLABS

card(s)7(+): MOE labels - from input data set (f)

Format: (20A4/3A4/3A4/I10/I10/I10/(5A4))

6. Identification: subnetwork post processor control cards

FORTRAN unit: 17

Number: as required (7 in current application)

Structure: BCD card images

Content: the following card images are written:

card1: post processor control card - input data set (e)

card2: label for alternative 1 - from MCLABS with prefix AGGREGATE

card3: label for alternative 2 - from MCLABS with prefix AGGREGATE

card4: number of observations for alternative

1 - from MCLABS

card5: number of observations for alternative 2 - from MCLABS

card6: number of subnetworks - from input data set (b), plus 1 for route total

card(s)7(+): MOE labels - from input data set (f)

Limitations

1. Maximum number of links: 100

2. Maximum number of observations in either alternative: 100

3. Maximum number of aggregate subnetworks: 60

4. Maximum number of MOE's: 10

Subnetwork Aggregation Program - MCOND

Operation

MCOND reads in a series of control parameters indicating which links are to be aggregated into subnetworks. As the program is currently written, these links must be consecutive entries on the route. Other inputs include the raw data from MCSTRIP, a LUSE table from MCCNTL2, and link lengths for computing subnetwork speeds from LNKLEN.

Travel time, stopped time delay, and number of stops are accumulated for all consecutive links comprising a subnetwork. The subnetwork length is computed in a similar manner and speed is computed from this value and total travel time. Not all links need be specified in a subnetwork. The program will also compute a route total for all links with an LUSE value equal to 1. This may or may not equal the sum of the subnetwork values. An output data set is produced in a form suitable for input to the post processor.

Inputs

1. Identification: MCLABS control card (C3)

FORTRAN unit: 5

2. Identification: link lengths (C1)

FORTRAN unit: 15

3. Identification: LUSE vector from MCCNTL2

FORTRAN unit: 10

4. Identification: MCOND control cards

FORTRAN unit: 12

Number: as required

Content: three types of cards are supplied:

card1: number of subnetworks

number: 1

format: (I2)

card2: beginning and ending link numbers

number: from card1

format: (2I2)

card3: subnetwork label identifiers

number: as required - 18 per card

format: (18A4)

Note: records of type card 3 are not read in MCOND but are read in MCCNTL2

Organization: PDS, one member per route

5. Identification: raw data from MCPLUG2

FORTRAN unit: 8

Output

1. Identification: subnetwork MOE post processor inputs

FORTRAN unit: 9

Number: 1 record for each observation for each of the 4 aggregate MOE's

Structure: binary integer full words

Content: sequence number, followed by the value of an aggregate MOE for each subnetwork. All observations for travel time are written first, followed by delay, stops, and speed, in that order.

Limitations

1. Maximum number of links: 60

2. Maximum number of aggregated subnetworks: 12

3. Maximum number of observations: 40

Statistical Post Processor Program - POST (moving car analysis write-up)

Operation

The statistical post processor is designed to provide a comparison between two sets of data. For each MOE selected, the program first computes means and standard deviations, then the percentage difference between the means. The distributions are then examined with three different statistical tests: a simple t-test, the Mann-Whitney U-test, and the Kolmogorov-Smirnov test. Statistical significance at the 1%, 2%, and 5% levels is indicated.

The statistics are normally generated for each link or aggregate subnetwork across all observations. The comparisons can also be made for a pair of observations across all links. The program will accept multiple MOE's for which independent statistical tests will be performed. Statistical significance is indicated by 0, 1, 2, or 3 stars next to the test parameter, indicating insignificance differences or differences at the 5%, 2%, and 1% levels for the latter values, respectively.

The program will accept data as binary integer full words, binary integer half words, or binary floating point full words. All data must be supplied with one record per observation, containing values of one MOE for all links. All observations for one MOE are supplied consecutively, followed by observations for subsequent MOE's. Either link or subnetwork analyses may be specified. For subnetwork analyses, subnetwork identifier lables must be input to the program.

Inputs

1. Identification: program control card

FORTRAN unit: 5

Structure: BCD card image

Number: as required, normally 1

Content: FORTRAN NAMELIST of the following type:

&PARAM	NAMELIST identifier	
NMOE = n	number of MOE's for testing	
TESTBS = n	type of tests to be performed	
	n = 1:	link or subnetwork statistics only
	n = 2:	link or subnetwork and observation statistics
	n = 3:	observation statistics only
LNKNT = n	type of analysis	
	n = 1:	link analysis
	n = 2:	subnetwork analysis
TWIND = n	type of data	
	n = 1:	binary floating point full words
	n = 2:	binary integer full words
	n = 3:	binary integer half words
&END	NAMELIST terminator	

Note: all values must be supplied

2. Identification: label for alternatives

FORTRAN unit: 5 - concatenated to (a)

Structure: BCD card image

Number: 2, one for each alternative

Format: (20A4)

Vehicle Minutes/Vehicle Miles Computation - VEHMIN

Operation

VEHMIN begins by reading a series of control parameters, including trap lengths and LUSE and OBUSE vectors, for two sets of detector outputs being compared. The program then begins a loop over all observations for each alternative. If the observation is usable (OBUSE=1), vehicle minutes of travel, vehicle minutes of delay, and vehicle miles of travel are computed. These values are then accumulated by section and for the network as a whole for all usable links (LUSE=1). Average speed as vehicle miles/vehicle minutes is also computed within subroutine SPEEDY.

After the values have been accumulated for all usable observations and for both alternatives, a vehicle minutes summary is produced. Average values for vehicle minutes of travel and vehicle minutes of delay are computed by dividing the cumulative values by the number of usable observations. Similarly, an average speed across all observations is computed.

The vehicle minutes summary is then printed. Values are recorded for each link for both alternatives with the differences recorded separately for positive and negative results. Section totals are also computed, with the positive and negative sums computed separately as well as a net figure. Finally, the average speeds for each section and the network as a whole for the two alternatives are printed.

VEHMIN then passes control to routine PLOT which prints the values of vehicle minutes and vehicle miles for each observation for each section and the network as a whole. Printer plots are then generated with vehicle minutes against vehicle miles for each section, with data for the two alternatives shown with distinctive characters.

Finally, VEHMIN passes control to subroutine REG which computes the linear regression of vehicle minutes on vehicle miles. A simple summary is printed showing the intercept, slope, multiple correlation coefficient (R^2), and standard error of the two equations generated. Separate equations are summarized for each section and the network as a whole.

The program requires outputs of the NETCON and PREPOST routines developed as part of the general detector analysis package. Therefore, for the current application, VEHMIN is run behind these two programs as a special procedure, passing the necessary information, to avoid maintaining a large number of additional data sets.

Inputs

1. Identification: label cards

   FORTRAN unit: 5

   Number: 2

   Format: (20A4)

   Source: 2 members of general purpose label data set LABS, concatenated

2. Identification: trap lengths

   FORTRAN unit: 8

   Number: as required

   Format: (20I4)

   Content: detector identification followed by trap length, in pairs

   Source: special data set

3. Identification: link use vector

   FORTRAN unit: 10

   Number: 1 record

   Structure: binary integer half words

   Source: LUSE vector for particular combination; output of NETCON

4. Identification: number of observations

   FORTRAN unit: 12

   Format: (I10)

   Number: 2

   Source: 2 members of observation vector NOBS, concatenated

5. Identification: observation use vector

   FORTRAN unit: 14

Number: 1 record

Structure: binary integer half words

Source: OBUSE vector from PREPOST for a particular combination

6. Identification: link volumes

FORTRAN unit: 16

Number: as appropriate

Structure: binary integer half words

Source: 2 unique data sets for MOE2, concatenated

7. Identification: average delay

FORTRAN unit: 18

Number: as appropriate

Structure: binary integer half words

Source: 2 unique data sets for MOE8, concatenated

8. Identification: average travel time

FORTRAN unit: 20

Number: as appropriate

Structure: binary integer half words

Source: 2 unique data sets for MOE7, concatenated

Outputs

VEHMIN produces no outputs other than the printed records noted above.

Limitations

1. Maximum number of usable links: 150

2. Maximum number of observations: 100

Note: Breakpoints for sections are compiled into the program at detector numbers 19, 50, and 195. Alternate identifications could be accommodated, but a different number of sections requires major program alterations.

APPENDIX D
RAW DATA TABLES

AM	Route No.	Route Mileage	Travel Time $\bar{X}$	Travel Time SD	Travel Time $\bar{Y}$	Travel Time SD	Delay $\bar{X}$	Delay SD	Delay $\bar{Y}$	Delay SD	Stops $\bar{X}$	Stops SD	Stops $\bar{Y}$	Stops SD	Speed $\bar{X}$	Speed SD	Speed $\bar{Y}$	Speed SD
3-Dial	11	2.285	478	40	519	49	84	29	87	40	6.3	1.6	7.1	2.8	17.4	1.5	16.0	1.4
	13	2.25	465	27	472	100	85	47	80	51	4.2	1.9	3.9	3.2	18.0	3.4	17.8	3.3
	22	1.50	374	68	328	89	104	40	64	55	5.7	1.8	3.3	2.4	14.8	2.5	17.6	4.3
	24	1.53	481	162	506	123	179	118	201	84	8.3	4.2	7.5	3.2	12.3	2.8	11.5	2.4
	30	3.105	984	280	1043	332	245	146	363	203	12.2	5.8	14.6	8.2	12.4	3.7	11.9	3.8
	40	2.66	818	186	785	150	295	119	251	85	12.4	4.3	9.8	2.9	12.1	2.4	12.6	2.2
TOD	11	2.285	478	41	479	73	84	30	91	54	6.2	1.5	6.3	2.4	17.4	1.6	17.6	2.5
	13	2.25	466	86	451	73	86	47	87	41	4.2	1.9	4.1	1.7	18.0	3.3	18.3	2.6
	22	1.50	376	68	359	57	105	40	86	30	5.7	1.8	5.1	1.4	14.7	2.5	15.4	2.4
	24	1.53	488	168	466	131	184	122	160	89	8.5	4.3	7.8	3.1	12.2	2.8	12.4	2.5
	30	3.105	987	284	1086	349	264	149	373	240	12.3	5.9	12.0	5.4	12.4	3.8	11.4	3.6
	40	2.66	823	191	911	128	299	122	287	127	12.3	3.8	14.0	3.8	12.1	2.4	10.8	1.6
CIC	11	1.675	393	40	377	49	52	28	67	34	4.5	1.6	4.6	1.8	15.6	1.8	16.3	2.3
	13	1.68	405	58	377	75	76	30	80	39	3.6	1.5	3.9	1.8	15.1	2.1	16.7	3.0
	22	1.5	442	131	460	79	130	82	125	48	6.1	3.5	5.7	2.3	13.1	2.9	12.2	1.9
	24	1.53	512	159	565	164	185	111	207	112	8.3	3.9	8.7	3.7	11.5	2.8	10.3	2.4
	30	3.105	883	264	988	308	254	163	304	175	12.5	6.1	14.6	7.0	13.6	3.6	12.3	3.2
	40	2.66	999	203	1004	260	354	122	365	173	15.4	5.0	17.2	7.8	9.9	1.9	10.1	2.3
BPS	11	2.285	477	75	488	39	68	42	87	27	5.4	2.7	6.5	1.8	17.6	2.0	17.0	1.5
	13	2.25	455	55	464	52	94	38	94	38	4.1	3.5	5.0	1.6	18.1	2.1	17.7	2.2
	22	1.50	424	85	444	90	104	43	120	50	4.6	1.8	5.8	2.2	13.1	2.3	12.7	2.9
	24	1.53	472	90	512	136	139	58	161	82	6.1	1.7	7.7	3.3	12.0	2.2	11.4	2.5
	30	3.105	890	269	1028	321	278	143	345	196	12.8	6.6	18.7	10.3	13.8	4.2	11.9	3.8
	40	2.66	886	119	897	171	296	73	302	124	14.2	3.7	14.2	4.9	10.9	1.4	11.0	2.1

Figure 45. Numeric base data, a.m. period.

MIDDAY	Route		Travel Time				Delay				Stops				Speed			
	No.	Mileage	$\bar{X}$	SD	$\bar{Y}$	SD	$\bar{X}$	SD	$\bar{Y}$	SD	$\bar{X}$	SD	$\bar{Y}$	SD	$\bar{X}$	SD	$\bar{Y}$	SD
3-Dial	11	2.285	561	69	573	73	95	43	73	57	6.7	2.3	5.0	4.2	14.9	1.9	14.5	1.9
	13	2.25	631	105	679	97	155	60	146	108	8.9	3.4	6.7	4.8	13.1	212	12.2	1.6
	22	1.50	637	131	448	50	273	87	142	39	12.2	4.3	5.9	1.6	8.7	1.8	12.2	1.3
	24	1.53	474	64	444	55	151	47	132	35	7.0	2.4	4.7	1.8	11.8	1.7	12.6	1.6
	30	3.105	1311	218	1047	95	453	116	319	68	22.5	7.4	13.2	4.4	8.8	1.4	10.8	1.1
	40	2.660	899	127	1071	188	306	83	447	145	12.8	3.1	15.6	3.3	10.8	1.5	9.3	1.8
TOD	11	2.285	564	67	570	75	97	41	106	48	6.9	2.4	7.3	2.3	14.8	1.9	14.7	1.8
	13	2.25	633	104	621	88	157	63	143	59	9.0	3.1	8.4	2.8	13.1	2.1	13.3	1.7
	22	1.50	635	126	571	90	273	83	222	64	12.1	4.2	9.7	2.5	8.7	1.7	9.6	1.6
	24	1.53	475	64	497	73	151	47	164	48	6.9	2.3	7.5	2.2	11.8	1.6	11.3	1.7
	30	3.105	1313	216	1191	205	456	116	426	180	21.8	6.8	12.3	2.4	8.7	1.4	9.7	1.4
	40	2.66	909	92	982	107	303	62	331	66	13.2	2.8	13.8	2.5	10.6	1.1	9.8	1.1
CIC	11	1.675	501	92	500	73	94	61	103	43	6.1	2.4	6.0	2.1	12.5	2.1	12.3	1.7
	13	1.68	513	71	492	52	112	47	99	48	5.4	2.1	5.9	2.7	12.2	1.7	12.4	1.3
	22	1.50	622	90	631	117	236	70	229	65	10.7	2.7	10.0	2.8	8.9	1.3	8.9	1.7
	24	1.53	552	94	576	89	192	71	185	51	8.4	2.6	7.8	2.0	10.2	1.6	9.8	1.5
	30	3.105	1277	163	1189	91	507	108	434	67	22.9	4.4	19.9	3.4	8.8	1.3	9.3	.9
	40	2.66	1213	169	1093	167	498	122	408	116	19.6	4.3	17.5	4.4	8.0	1.2	8.9	1.4
BPS	11	2.285	537	60	562	68	78	39	106	40	5.3	2.4	7.9	2.8	15.5	1.5	14.8	2.0
	13	2.25	593	85	629	117	121	45	169	63	6.6	2.2	10.5	3.7	14.0	2.0	13.3	2.3
	22	1.50	687	96	607	94	237	46	188	58	10.7	2.8	9.7	2.3	8.0	1.2	9.1	1.2
	24	1.53	565	64	582	88	171	51	172	61	7.0	2.2	8.8	2.2	9.8	1.3	9.7	1.4
	30	3.105	1121	129	1376	139	372	87	590	113	16.3	3.0	26.0	3.8	10.2	1.2	8.3	.9
	40	2.66	1049	119	1198	147	385	90	485	104	17.2	4.0	20.6	3.0	9.1	1.0	8.1	.9

Figure 46. Numeric base data, midday period.

PM	Route		Travel Time				Delay				Stop				Speed			
	No.	Mileage	$\bar{X}$	SD	$\bar{Y}$	SD	$\bar{X}$	SD	$\bar{Y}$	SD	$\bar{X}$	SD	$\bar{Y}$	SD	$\bar{X}$	SD	$\bar{Y}$	SD
3-Dial	11	2.285	464	42	542	80	48	34	66	56	4.2	2.7	3.5	3.2	17.8	1.5	15.5	2.1
	13	2.25	552	88	629	93	125	50	130	79	6.1	2.2	7.3	3.5	15.0	2.0	13.1	1.8
	22	1.5	485	78	537	101	176	62	218	85	8.2	2.1	7.2	1.4	11.4	1.7	10.3	1.8
	24	1.53	533	178	501	186	197	135	183	136	8.4	4.1	5.2	2.5	11.4	3.6	12.2	3.7
	30	3.105	1366	389	1243	258	503	256	483	166	21.9	8.3	18.8	6.2	8.9	2.2	9.3	1.8
	40	2.66	977	252	984	203	374	175	356	144	14.5	5.6	14.5	6.5	10.4	2.7	10.2	2.5
TOD	11	2.285	480	51	485	42	59	45	70	27	4.6	3.0	5.0	1.4	17.3	1.8	17.1	1.4
	13	2.25	553	85	527	61	123	49	124	44	5.9	2.0	5.6	1.7	14.9	1.9	15.5	1.7
	22	1.5	488	70	477	88	174	54	158	65	8.1	2.1	7.6	2.1	11.3	1.5	11.6	1.8
	24	1.53	533	180	522	236	196	135	177	161	8.3	4.3	8.0	6.6	11.5	3.9	12.2	4.0
	30	3.105	1370	366	1355	283	502	252	279	237	20.8	7.7	15.7	2.9	8.8	2.1	8.6	1.9
	40	2.66	950	244	995	272	356	161	327	158	14.2	5.8	13.6	5.4	10.7	2.8	10.5	3.1
CIC	11	1.675	429	55	411	73	61	28	65	35	4.1	1.7	4.4	2.1	14.3	1.9	15.1	2.6
	13	1.68	466	67	471	92	103	41	117	55	5.0	1.9	5.0	1.9	13.2	2.0	13.3	2.4
	22	1.5	555	90	575	195	205	73	193	154	9.5	3.0	7.9	4.2	9.9	1.5	10.1	2.6
	24	1.53	591	131	796	498	218	86	340	371	10.2	4.5	12.7	11.6	9.6	2.1	8.5	3.1
	30	3.105	1317	418	1408	362	553	321	596	254	22.9	8.0	25.0	9.9	9.2	2.2	8.3	1.9
	40	2.66	1382	317	1444	303	599	217	639	201	23.4	7.4	27.4	8.6	7.3	1.5	6.8	1.5
BPS	11	2.285	500	38	540	66	61	30	103	51	3.9	1.7	6.8	2.4	16.8	1.3	15.4	1.7
	13	2.25	534	31	599	119	110	21	163	49	4.8	1.0	8.3	2.6	15.1	.9	13.9	2.7
	22	1.50	521	96	527	77	148	63	150	59	7.0	2.3	7.6	2.3	10.6	1.8	10.4	1.7
	24	1.53	625	222	618	150	214	131	216	108	8.6	4.0	9.3	3.7	9.7	2.9	9.4	1.8
	30	3.105	1424	331	1524	416	591	241	699	311	24.8	7.0	31.7	12.0	8.3	2.0	7.8	1.9
	40	2.66	1259	391	1326	345	522	268	577	231	22.1	9.4	23.2	8.0	8.3	2.5	7.7	1.9

Figure 47. Numeric base data, p.m. period.

☆U.S. GOVERNMENT PRINTING OFFICE: 1975 629-045/628 1-3

inistration.

E	DATE
	12/15/86
	9-21-92
	~~20/03/94~~

FHWA
R&D