ARIZONA DEPARTMENT OF TRANSPORTATION

REPORT NUMBER: FHWA / AZ 86/209

COMPARATIVE ASSESSMENT OF COMPUTER PROGRAMS FOR TRAFFIC SIGNAL PLANNING, DESIGN, AND OPERATIONS

Volume II
Software Description

Prepared by:
A. Essam Radwan
Ahmad Sadegh
Judson S.Matthias
Subramaniam D. Rajan
Center for Advanced Research in Transportation
College of Engineering & Applied Sciences
Arizona State University
Tempe, Arizona

December 1986

Prepared for:
Arizona Department of Transportation
206 South 17th Avenue
Phoenix, Arizona 85007
in cooperation with
U.S. Department of Transportation
Federal Highway Administration

NOTICE

TECHNICAL REPORT DOCUMENTATION PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
FHWA/AZ-86/209		

4. TITLE AND SUBTITLE Comparative Assessment of Computer Programs for Traffic Signal Planning, Design and Operations. Volume 2: Software Descriptions	5. REPORT DATE December 1986 6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) A. E. Radwan, A. Sadegh, J. S. Matthias, S. D. Rajan	8. PERFORMING ORGANIZATION REPORT NO.
9. PERFORMING ORGANIZATION NAME AND ADDRESS Center for Advanced Research in Transportation College of Engineering and Applied Sciences Arizona State University Tempe, AZ 85287	10. WORK UNIT NO. 11. CONTRACT OR GRANT NO. HRP-1-31(209)
12. SPONSORING AGENCY NAME AND ADDRESS Arizona Department of Transportation 206 South 17th Avenue Phoenix, AZ 85007	13. TYPE OF REPORT & PERIOD COVERED Final Report September 1984 - December 1986 14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES Prepared in cooperation with the U.S. Department of Transportation, Federal Highway Administration from a study on Comparative Assessment of Computer Programs for Traffic Signal Planning, Design, and Operations. The opinions and conclusions are those of the authors and not necessarily of the Federal Highway Administration.

16. ABSTRACT

The main goal of this study was to comparatively assess a group of selected computer programs for traffic signal planning, design, and operations. A comprehensive inventory was conducted and a detailed list of the currently available computer software was developed. A short description of individual software was provided, and a comparative processor software, and network software. The final recommendation of software included seven programs for isolated intersections, three preprocessor software, and five network software.

Volume 1 Study Approach, Analysis and Recommendations
Volume 2 Software Descriptions
Volume 3 Recommended Software Output

17. KEY WORDS	18. DISTRIBUTION STATEMENT
Computer Software, Isolated Signals, Arterials, Grid Networks, Software Assessment	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		461	

ACKNOWLEDGEMENTS

This report was prepared as part of Project HPR-PL-1 (27) Item 209 - ''Comparative Assessment of Computer Programs for Traffic Signal Planning, Design, and Operations.'' This research was conducted by the Center for Advanced Research in Transportation of the College of Engineering and Applied Sciences at Arizona State University in conjunction with the Arizona Department of Transportation, and the U.S. Department of Transportation, Federal Highway Administration.

The project staff wishes to acknowledge the assistance provided by Arizona Transportation Research Center of Arizona Department of Transportation. Special thanks are extended to Mr. Harvey Friedson of the City of Tempe for providing the traffic data, and Mr. Mark Schlappi of Arizona Department of Transportation for making the McTran package available to the team and Dr. Robert Wortman, University of Arizona.

Volume 1: Study Approach, Analysis, and Recommendations

TABLE OF CONTENTS

LIST OF TABLES

Volume 1: Study Approach, Analysis, and Recommendations

LIST OF TABLES (continued)

LIST OF FIGURES

Volume 2: Software Description

TABLE OF CONTENTS

Volume 2: Software Description

TABLE OF CONTENTS (continued)

Volume 2: Software Description

TABLE OF CONTENTS (continued)

LIST OF TABLES

Volume 2: Software Description

LIST OF FIGURES

Volume 3: Recommended Software Output

TABLE OF CONTENTS

Volume 3: Recommended Software Output

TABLE OF CONTENTS (continued)

CAPCALC (Circular 212)

PROGRAM DESCRIPTION - In 1982, RCAI (Roger Creighton Associates Incorporated) released CAPCALC, a microcomputer software package designed to enable traffic engineers to calculate the capacities and levels of service of at-grade signalized and unsignalized intersections (1).

CAPCALC (intersection CAPacity CALCulation) calculates the capacities and levels of service (LOS) of at-grade signalized and unsignalized intersections using procedure published in Transportation Research Circular 212. The CAPCALC program can be run by anyone with a good technical background who has taken the time to familiarize himself/herself with Circular 212.

CAPCALC is written for a) the Apple II, II+, or IIe computer, b) the IBM Personal Computer, or c) the TRS-80 model 11, 12, or 16.

CAPCALC does not run under DOS, but under a different operating system called the P-system. Before you can use CAPCALC you must set up a separate partition for this operating system on your hard disk.

INPUT - CAPCALC has a screen editing feature. The input data for the signalized intersection are:

1) Intersection general information
2) Intersection geometry for each approach including the number of lanes, width and movement per lane per approach
3) Volume of traffic turning and going straight including the percentage of trucks and buses per approach
4) Critical gap for turning and through traffic, peak hour factor, gradient and right turn factor per approach
5) Cycle length

The input data for the unsignalized intersection uses the physical and traffic data entered as listed above.

In entering data, it is necessary, if unsignalized capacities are to be calculated, to enter the critical gap acceptance data for the minor traffic.

OUTPUT - The SIGNALIZED intersection calculation is made of two modules. The ''PLANNING'' module follows the ''PLANNING'' method of Circular 212. The output for ''PLANNING'' module is as follows:

1) Volume allocation to lanes
2) Left turn check, and
3) Planning results:
   a) number of phases
   b) movements per phase
   c) critical volume per phase
   d) percent capacity used
   e) intersection critical volume
   f) intersection percent capacity used, and
   g) intersection level of service

The ''OPERATIONS AND DESIGN'' module follows the ''OPERATION AND DESIGN'' method of Circular 212. The procedure is the same except that there is an additional output, adjusted per lane volumes. The final results also include, 1) effective green time per phase, 2) average delay per phase, and 3) intersection effective green time.

The output for UNSIGNALIZED intersection is as follows:

a) volume allocation to lanes, and
b) reserve capacity and level of service per lane and per approach

SPECIAL FEATURES AND SHORTCOMINGS - As an aid to CAPCALC users, several additional MOE have been added to Circular 212 procedures, and they are:

1. <u>Percent Capacity Used</u>

   a) CAPCALC automatically calculates the percent of total intersection capacity used by each phase. In both the Planning and Operations and Design Applications, the total intersection capacity is the maximum sum of critical volumes for level of service E corresponding to the number of phases required at the intersection.

   b) The individual percentages for each phase when summed yield the total percent of intersection used. This statistic is an indication of the degree of saturation of the intersection.

2. <u>Effective Green Time</u>

   CAPCALC automatically calculates the effective green time associated with each phase. The intergreen time (amber) for the intersection (in seconds) is calculated as two times the number of phases involved plus two. The effective green time for the intersection is obtained by subtracting the intergreen time from the desired cycle length. The ratio of the critical volume per phase to the sum of the critical volumes for the intersection is the proportion used in calculating the effective green time per phase.

3. <u>Average Delay</u>

   a) Whereas Circular 212 provides the average stopped delay incurred by all vehicles entering the intersection, CAPCALC provides an estimate of the average stopped delay by vehicles likely to be incurred during each phase of controlled movement.

b) The means for estimating delay is Webster and Cobbe's delay function, adapted for use in conjunction with Circular 212 procedures.

c) No delay values or green times are calculated if the level of service is at E or Failure because of the instability of the operating environment under these conditions.

4. It supplies the user with the directory of existing intersection files.

5. Permits rapid entry of required data.

6. It has a very good editing capability, allowing for correction of each screen.

7. It prints input data ''for the record.''

The program has the following shortcomings:

1) The program is limited to four lanes per approach.

2) Question which require single letter responses (such as ''Y'' or ''N'' accept uppercase letters only.

3) It does not run under MS-DOS, but rather under a different operating system called the P-System. this means that if the user wishes to install the program on a hard disk, a special partition is needed.

CAPCALC (1985 HCM)

PROGRAM DESCRIPTION - CAPCALC (intersection CAPacity CALCulation) calculates the capacities and levels of service (LOS) of at-grade signalized and unsignalized intersections (2).

The new HCM 1985 procedures were so markedly different from the Circular 212 procedures that RCAI (Roger Creighton Associates Incorporated) determined to rebuild CAPCALC totally. A supporting reason was to expand the original CAPCALC so that it could deal explicitly with intersections having up to six lanes per approach.

The resulting program was named ''CAPCALC 85''. Its purpose is to increase the productivity of traffic engineering personnel by replacing the manual forms and calculations used in the new HCM with computer procedures capable of being run on a number of widely-available microcomputers.

CAPCALC is written for a) The Apple II/II+/IIe computers, b) The IBM PC/PC XT computers, and TRS-80 II/12/16.

CAPCALC does not run under MS-DOS, but under a different operating system called the P-system. Before CAPCALC can be used, it is necessary to set up a separate partition for this operating system on a hard disc.

INPUT - CAPCALC has a screen editing feature. The input data for the signalized intersection are:

1) Intersection description
2) If the intersection is signalized, the type of signal (Actuated, Semiactuated, or pretimed)
3) Traffic and roadway conditions:
   - percent grade
   - percent heavy vehicles

- presence of adjacent parking lane
- number of parking maneuvers/hour
- number of buses stopping/hour
- peak hour factor for area type
- number of conflicting pedestrians/hour
- presence of a pedestrian button
- number of seconds allowed for pedestrians
- arrival type for approach

4) Geometrics and Volumes per approach

5) Signal phasing

OUTPUT - The SIGNALIZED intersection calculation is composed of two modules, the ''PLANNING'' and ''OPERATIONS AND DESIGN.'' The output for ''PLANNING'' module is as follows:

1) volume for each lane groups per approach
2) E-W and N-S critical volume, and
3) STATUS of the intersection

The output for ''OPERATIONS AND DESIGN'' module is as follows:

1) VOLUME ADJUSTMENT SCREEN which permits the experienced traffic engineer to edit flow rates, lane utilization factors, adjusted flow rates, and proportion of left and right turns in order to fine-tune the calculation to satisfy special circumstances.
2) SATURATIONS AND DESIGN SCREEN which permits the experienced traffic engineer to edit ideal saturated flow, and adjustment factors for: width, heavy vehicle, grade, parking and bus maneuver, area, and right and left turn. This permits fine-tuning the calculations to satisfy special circumstances.

3) CAPACITY ANALYSIS SCREEN which permits the experienced traffic engineer to edit permissive left turn flows, critical lane groups, and lost time per cycle in order to fine-tune the calculations to satisfy special circumstances.

4) LEVEL OF SERVICE SCREEN is an output screen reflecting the input values on the previous three screens. This screen contains the following MOE per lane group per approach:

a) V/C ratio

b) green ratio

c) 1st delay, 2nd delay, lane group delay, and approach delay

d) lane group capacity and level of service

Also, the values for intersection delay and LOS are included within this screen.

5) Intersection diagram

SPECIAL FEATURES AND SHORTCOMINGS - The following qualities and features were built into CAPCAL 85 to increase productivity.

- It can handle up to six lanes per approach.
- It includes three types of signals (pretimed, actuated, and semi-actuated).
- Full ''bullet-proofing'' (''error-trapping'') to prevent data entry errors
- Program ''defaults'' to typical standard values of lane width, peak hour factor, and many other variables to speed up data entry. (The user can enter any other value if so desired.)
- User control over critical variables including signal phasing, signal timing, and vehicle adjustment factors

- Full screen cursor control to allow rapid editing of input values
- It allows the user to move forward or backward from one screen to another screen.
- ''Lane grouping'' for ease of data entry
- Upon revision of the data in any data cell, all results are re-calculated almost instantaneously.
- Printed output in exactly the same form as input and intermediate-calculation screens
- Output suitable for camera-ready reproduction in engineering reports
- Input data printable ''for the record''
- A cycle length of 0 will cause the program to estimate cycle length based on HCM methodology (only if the intersection is not saturated).
- As each green time is entered, it computes and displays the resulting yellow plus red time.
- The program will allow you to enter data in either upper or lower case.
- It is possible to stop the program when it is in the middle of doing something and get back into the program as though it had never been interrupted.
- It can change the number of lines per page in printout.

There are a few shortcoming within this software. These are:

- At least one type of lane group combination is missing in manual
- The user has to be familiar with the 1985 HCM.
- In order to input movements (LTR) the capital letters have to be used.

- It does not run under MS-DOS, but rather under a different operating system called the P-system. This means that if the user wishes to install the program on a hard disk, a special partition is needed.
- If the user chooses a zero cycle length, the program will not calculate the splits and level of service if the intersection is saturated.

CAPSSI

PROGRAM DESCRIPTION - CAPSSI is an acronym for ''Comprehensive Analysis Program for Single Signalized Intersections.'' It entitles a very powerful traffic engineering programming tool (3).

The purpose of the program is to aid the engineer in optimizing traffic signal settings and/or measuring the performance of a single signalized intersection when a specific data set is given. Preliminary considerations of the evaluation of the impacts at a given signalized intersection indicated the need for an advanced algorithm for traffic impact evaluation that would be highly interactive. The BASIC computer language was chosen to develop such a program.

INPUT - The required input is listed below:

1) The intersection identification number.
2) The intersection name or description.
3) The number of critical phases (computed by the user).
4) The flow, saturation capacity flow and lost time for each critical phase (acquired usually from the field).
5) The minimum green times for the required cycle time solution.
6) The actual green times for the predetermined cycle time solution.
7) A range of cycle lengths that are required to be tested.
8) The minimum time for pedestrian crossing is required only if the user is using the Required Cycle Time option.
9) Data for pollution calculations.

OUTPUT - It can be divided into two parts:

1) A record of the time sharing session with the user. This is an echo of what the user inputted. First, a title is printed and then the name of the intersection as inputted by the user. The data for each phase are columnar with the critical phase numbering across the top and the description of each row of data to the left. The first three rows are, respectively: flow, saturation flow and the lost time for each critical movement as inputted by the user. The minimum times are also repeated. Below the table the values for the ''approach and departure speed'' and ''total intersection volume'' are echoed by the computer if the pollution calculations were requested.

2) The second part of the output is the analysis of the input data as performed by CAPSSI. All the information provided by this section is vital to the user in regard to the performance of the intersection on the basis of the inputted data. The output information provided by CAPSSI is listed and defined below:

- RELATIVE SATURATION: The ratio of the flow to the maximum flow which can be passed through the intersection from a particular approach.
- EFFECTIVE GREEN TIME: The chosen green time for that movement minus the lost time.
- MOVEMENT TIME: The effective green time plus the lost time.
- AVERAGE DELAYS: The theoretical time that a vehicle will have to wait in each respective critical movement to pass through the intersection.
- LEVEL OF SERVICE: An objective rating of the performance of the intersection ranging from A+ to F+.

- AVERAGE QUEUE VEHICLES: The theoretical average queue in each respective phase at the beginning of the green period for that phase.
- AVERAGE STOPS AND STARTS: The theoretical average number of times a vehicle must stop or start during a complete cycle for each respective phase.
- DO VEHICLES CLEAR: Tells if, on the average, the vehicles will theoretically clear during any respective green time.
- WEBSTER'S OPTIMUM CYCLE LENGTHS: The cycle time which gives an equal degree of saturation and minimum delay.
- REQUIRED CYCLE LENGTH: The cycle length chosen by CAPSSI that will satisfy all the minimum green times supplied by the user and results in minimum delay for those conditions.
- PREDETERMINED CYCLE LENGTH: The sum of the actual movement times supplied by the user. This will also be the negative cycle time that the user might input to the program to determine the approximate splits for a particular cycle time.
- WEIGHTED AVERAGE DELAY, AND LEVEL OF SERVICE: The average delay for all critical movement weighted with respect to the flow values, in seconds, and an objective level of service rating for this delay.
- EXCESS FUEL USAGE, GALLONS/HOUR: The amount of fuel usage resulting from vehicles stopping at the intersection.
- EXCESS CO, HC AND NOX, GRAM/HOUR: The amount of emissions in grams (hydrocarbons, carbon monoxides and nitrous oxides) in the critical hour for all vehicles stopping at the intersection.

SPECIAL FEATURES AND SHORTCOMINGS: The program has the capability of performing many functions that are of use to the engineer. These functions can be enumerated as follows:

1) CAPSSI will optimize the green time splits at the intersection using the Webster Method. It will then calculate the delays of the critical movements in each phase and objectively evaluate its level of service.

2) CAPSSI can split the cycle time between the critical movements on the basis of equating the degree of saturation for all movements by inputting a negative cycle time into the Actual movement time for one of the critical movements.

3) CAPSSI will evaluate the delays and level of service for existing green splits.

4) CAPSSI will provide the user with (for each critical phase) the average number of stops and starts per vehicle, will calculate the theoretical queue length, and will calculate the fuel consumed and pollution emissions per hour for existing green splits and/or optimized green splits for a given set of data.

5) CAPSSI will consider pedestrian crossing times (or minimum green times) for optimization of splits only. CAPSSI will allow the user to optimize the splits while satisfying all the minimum green times.

6) Because CAPSSI is highly interactive, it allows the user to test a full range of sets of data. These will include saturation flow, a particular movement, test a range of cycle lengths, the approximate split distribution for a particular cycle and test the validity of an existing cycle time and its corresponding split.

7) CAPSSI can be run many times at each intersection to test the impact of different traffic loadings on the intersection. This can be very

useful to the writers of Environmental impacts, as before and after traffic loading conditions may be considered. It can also serve as a useful tool in street design or improvements by evaluating the needed capacity at the intersection.

The major shortcomings of this software are listed below as:

1) CAPSSI requires external hand calculation of Critical Movements.
2) It cannot list the directory of data files.
3) Printer has to be on or the program will automatically terminate the session.

CMA

PROGRAM DESCRIPTION - CMA is a program which uses critical lane movement analysis procedures to examine the capacity and level of service at signalized intersections. The program follows very closely the analysis procedures described in detail in the Transportation Research Board Circular 212. The 212 procedure was developed as the result of research work carried out by JHK Associates and the Traffic Institute of Northwestern University under National Cooperative Highway Research program (NCHRPP) project 3-28 (4).

CMA is a menu driven, interactive program which makes extensive use of prompting to encourage proper operation. The program is designed for simple and effective operation by traffic engineering personnel.

The user should review the TRB 212 procedures to become familiar with its strengths and limitations. It must be noted that some engineering judgement is required to gain the most from the TRB 212 procedure. The procedure is, however, easy to use and will provide insight into the service provided at signalized intersections with a limited amount of effort.

The program is operated from a set of main menu selections. Data entry, data file management and program run operations are each selected from menu choices. Each analysis scenario is managed as an independent disk file. Files can be duplicated and saved under new names and easily modified to test alternative evaluation situations. A concise but complete summary report can be produced as the user wishes.

The program operates on microcomputing systems with the following capacities, features and characteristics:

1. The system must operate under a version of CP/M or MS-DOS.
2. The system must be configured to provide approximately 40K.

3. The system must at least have one disk drive with a minimum of 90K bytes of free space.

4. A printer is not required for analysis but is required to produce summary reports.

INPUT – The required input data are as follow:

1. General Intersection information
2. Intersection volumes
3. Percent Trucks, Percent Buses, Peak Hour Factor, G/C
4. Intersection phasing including phase overlaps
5. Lane Channelization/Widths
6. Intersection Cycle Length
7. Actual Critical Movements

OUTPUT – The first two pages of the output are the user inputted data. The output also includes the following:

1. Left turn capacity check
2. Passenger car equivalent period traffic volumes
3. Calculated critical lane traffic volumes
4. Selected critical volumes (sum of critical volumes)
5. The V/C ratio
6. Indicated intersection Level of Service

SPECIAL FEATURES AND SHORTCOMINGS – CMA has the capability of listing the intersection files. It also has a very good editing feature for both old and new data files.

The user has to choose the critical movements which requires knowledge of Circular 212.

CMA/M

PROGRAM DESCRIPTION

The Critical Movement Analysis of the McTRANS package (CMA/M) is programmed in Applesoft for Apple II computers (5). It carries out a complete analysis for signalized intersections utilizing procedures documented in the TRB Circular 212 publication (Interim Material on Highway Capacity). The program identifies the critical traffic movements at an intersection, performs adjustments for lane width, trucks, pedestrians, left turns, local buses, etc. The program determines degree of saturation and level of service for signalized operation.

INPUT

The input stream to the program includes:

- o Intersection Geometrics: lane configurations and widths
- o Traffic volumes: traffic volumes per movement per direction, truck and local bus volumes, and peak hour factor.
- o Phasing: phasing scheme per direction, cycle length, and pedestrian activities.

OUTPUT

A typical output contains an echo of the input stream and a hard copy of the level of service and critical movements.

SPECIAL FEATURES AND SHORTCOMINGS

The program executes the CMA procedure as described in circular 212 step by step with no special features added to it. It performs analysis of signalized intersections with no capability of performing optimization or design. It runs on Apple II computers only. The screen editing capability is relatively rigid, and some of the editing functions do not perform properly.

EZ-POSIT

PROGRAM DESCRIPTION - EZ-POSIT is an interactive, window-oriented program for traffic engineers (6). With a minimum amount of required input data, the program can produce an optimal signal setting, including cycle time and phasing pattern, that can minimize fuel consumption for a given intersection.

No new techniques were used in developing the technical aspects of the program; however, a great number of concepts and methods described in the Critical Lane Movement Analysis documented in the TRB Interim Materials on Highway Capacity were adopted. This feature makes EZ-POSIT quite applicable to U.S. conditions.

The program is written in UCSD-PASCAL which was developed by the University of California at San Diego. It is the most popular high-level language in microcomputer application today.

INPUT - In the input stage, the program guides the user to specify required data. The user is provided with clear instructions via a menu format, and comprehensive error checks are made. In order to simplify the input procedure, many parameters are defaulted to represent average geometric and traffic conditions. Therefore, volume is the only data the user has to input to the program if the defaulted values can be applicable to his conditions. The input for the optimization process are:

1) Minimum green
2) Lane width
3) Local bus
4) Number of Lanes
5) Pedestrians

6) PHF (%)

7) Trucks (%)

8) Volume (vph)

9) Minimum and Maximum cycle length

The only additional data needed for a design process are:

1) Actual cycle length, and

2) Phasing

The phase setting for isolated intersections is divided into a N-S street pattern and an E-W street pattern, which are independent of each other. The pattern can be grouped into three types:

1) None of the left turn movements are protected

2) One of the left turn movements is protected

   a) North or East bound left turn is protected

   b) South or West bound left turn is protected

3) Both of the left turn movements are protected

   a) Left turn movement first, without overlap

   b) One direction first, without overlap

   c) Left turn movement first, with overlap

   d) One direction first, with overlap

The program can also evaluate the intersection if the user input the Actual green time in addition to above data.

OUTPUT - The optimization process is a procedure to select the optimum signal setting based on the minimum fuel consumption. The Optimization is described by the algorithm in Figure 1.

Three requirements are checked in every iteration before fuel consumption is computed. If any one of the requirements cannot be

```
For each possible N-S phase pattern loop
    For each possible E-W phase pattern loop
        computer PASSENGER_CAR_EQUIVALENT (PCE);
        determine CRITICAL_LANE_VOLUME;
        calculate SUM_OF_CRITICAL_LANE_VOLUME
        set MIN_CYCLE and MAX_CYCLE if necessary;
        from MIN_CYCLE to MAX_CYCLE step 5 sec loop
            calculate SPLIT for each phase;
            check MIN_OVERLAP_TIME (5 sec);
            calculate GREEN_TIME for each movement;
            check MIN_GREEN_TIME;
            check UNPROTECTED_LEFT_TURN;
            compute LEVEL_OF_SERVICE (LOS);
            compute DELAY, NO_OF_STOPS, FUEL_CONSUMPTION;
            record the CYCLE_TIME having MIN_FUEL_CONSUMPTION;
        end_of_cycle_loop;
        record the setting having MIN_FUEL_CONSUMPTION;
    end_of_EW_pattern_loop;
end_of_NS_pattern_loop;
```

Figure 1 EZ.POSIT Optimization Process

fulfilled, computation will not proceed and the program will jump out of the iteration and try the next combination. At the end of each iteration, the computed fuel consumption is compared with the previous optimum setting and the current optimum setting is recorded. The three requirements are as follows:

1) Minimum Overlap Phase - Because the green time defined in this study includes the clearance time, it is not practical to have an overlap phase shorter than the clearance time. Therefore, 5 seconds is used as the minimum overlap phase. This assures that the overlap phase is long enough to release at least two vehicles.

2) Minimum Green Time - It allows pedestrians enough time to cross the street or to avoid an unrealistically short green period.

3) Left Turn Check - This check examines if the exclusive left turn phase should be assigned to accommodate left turn movements. The procedure suggested is as follows:

$$V_L = \frac{7200}{C} + \frac{1200G}{C} - V_o$$

where: V_L = Capacity (vph) for an unprotected left turn

C = Cycle Length (sec)

G = Green time (sec) available for the movement

V_o = Volume (vph) of the opposing through movement

This allows two vehicles per cycle on the clearance and assumes that 1200 vehicles per hour of green (VPHG) will be accommodated as a linear combination of the unprotected left plus the opposing through movements.

If the cycle length is not specified by the user, the program sets a cycle range within the optimum cycle length is determined. The maximum cycle length is defaulted to 120 seconds. As for the minimum cycle length, it is suggested 30 sec. for a 2-phase condition, 45 sec. for a 3-phase condition and 60 sec. for a 4 or more phase condition as the minimum pretimed cycle length.

The OPERATION AND DESIGN APPLICATION of the Critical Movement Analysis is adopted in this program to compute the split and green time. The procedure is as follows:

1) Calculate Passenger Car Equivalence for each lane. The adjustment factors considered by Critical Movement Analysis are as follows:

    a) Lane width

    b) Bus and trucks

    c) Left turns

    d) Right turns and pedestrians activity

    e) Peaking characteristics (Peak Hour Factor)

2) Combining Critical Lane Volume. Critical lane volume for each street is determined.

3) Calculate Splits and Green Time. The allocation of cycle time is based on the proportion of the critical lane volume for each phase to the total critical lane volumes.

The simplified Webster Delay formula is adopted in this program.

$$D = \frac{9}{10}\left[\frac{C(1-L)^2}{2(1-\mu x)} + \frac{X^2}{2Q(1-x)}\right]$$

where: D = average delay (sec/veh.)

C = cycle length (sec)

μ = G/C ratio

X = Degree of Saturation

Q = Volume (pce/sec)

The program uses the following formula to calculate the proportion of vehicles required to stop (stop ratio) at a signal. On an average stop ratio is equal to the vehicles that arrive in red time divided by the vehicles that arrive in whole cycle.

$$S = \frac{R}{C(1-x)} = \frac{(1-L)}{(1-\mu x)}$$

where: R = Red Time (sec)

S = Stop Ratio

This formula assumes that a queue which formed in a red period would release completely during the following green period.

The excess Fuel consumption is calculated using the following formula that is a function of stops and delay.

$$F = Q(AS + BD)$$

where: F = Excess Fuel Consumption (gal/hr)

Q = Volume (pce/hr)

A = Fuel Consumption Rate for Stopped Vehicle (gal/hr/veh)

D = Average Delay per Vehicle (hr/veh)

B = Idle Fuel Consumption Rate (gal/hr/veh)

SOAP suggests 0.6 gal/hr/veh for an idle fuel consumption rate and 0.01 gal/hr/veh for a stopped fuel consumption rate.

As part of the Critical Movement Technique, Table 1 gives the recommended thresholds for the sum of the critical volumes for levels of service A through E.

Table 1 Levels of Service Ranges for CMA

(Source: Critical Movement Analysis)

Level of Service	Two Phases	Three Phases	Four or More Phases
A	1000	950	900
B	1200	1140	1080
C	1400	1340	1270
D	1600	1530	1460
E	1800	1720	1650

SPECIAL FEATURES AND SHORTCOMINGS - EZ-POSIT is an attractive, colorful, and creative window-oriented program. It has a short help file and an easy way of editing the file. It uses every possible combination within the cycle length range (increment of 5 seconds) and every available phase to arrive at the optimum cycle length and phase sequences for an intersection in order to minimize the total delay per intersection.

ICAP

PROGRAM DESCRIPTION - In the late 1960's, the Institute of Transportation and Traffic Engineering of the University of California, with partial support from the Automotive Safety Foundation, developed a series of computer programs for capacity analysis. The ICAP programs are based upon one of these programs. The initial work was carried out by Adolf D. May, Gale Ahlborn and Fredrick L. Colins and is described in articles published in January 1968 Traffic Engineering magazine (7).

ICAP is a set of programs which perform intersection approach capacity analyses in accordance with the definitions and procedures of HRB Special Report No. 87, the 1965 HIGHWAY CAPACITY MANUAL.

The program is configured to operate on a microcomputing system with the following minimum capacities, features and characteristics.

1. Versions of the program are available for CP/M 80 MS-DOS and PC DOS operating systems.
2. CP/M systems require 64K of memory. MS-DOS and PC-DOS systems require a minimum of 128K of memory.
3. The programs are operable on systems with one or more disk drives. One disk drive systems require a minimum of 240K bytes of free space. Two disk drive systems require a minimum of 90K bytes of free space per disk.
4. An 80 column printer is required for the output.

The ICAP programs may be used to calculate combinations of service volume, approach width, load factor, or G/C ratio for signalized intersection approaches. The programs are interactive and menu driven. They are easy to use, require minimum input data, run very quickly and produce essentially identical results to HCM manual capacity analysis efforts.

The ICAP programs consist of two programs, ACAP and TCAP. The ACAP program permits analysis of intersection approaches, the TCAP program permits analysis of intersection turning lanes.

Both ACAP and TCAP include a user accessible '?' Help feature from the main MENU. The '?' Help text files may be customized by the user to include special procedures and instructions.

The ICAP programs operate in two stages, data entry and data processing. The data entry operation is used to prepare, edit and save data. The processing operation analyses prepared input and produces printed reports.

The basic relationship of the intersection capacity program used in ICAP is:

SV = f[(AVw.1f), (G/C), (POP,PHF), (LIM), (RT), (LT), (TF), (BF)]

where:

SV = Service volume in vehicles per hour.

(AVw.1f) = Approach volume in vehicle per hour of green based on approach width and load factor.

(G/C) = Green phase time to cycle time ratio.

(POP,PHF) = Adjustment factor based on metropolitan area size and peak hour factor.

(LIM) = Adjustment factor based on location in metropolitan area.

(RT) = Adjustment factor based on percent right turns.

(LT) = Adjustment factor based on percent left turns.

(TF) = Adjustment factor based on percent trucks through buses.

(BF) = Adjustment factor based on local buses and type bus stop.

After the truck factor is determined, the program proceeds to complete the capacity calculations. Processing is divided into four major parts depending upon the quantity to be determined, service volume, approach width, load factor or G/C ratio.

In calculating an approach width, the Peak Hour Factor, metropolitan area size and location in the metropolitan area are determined. The computer then selects a trial width assuming 10% left turns, 10% right turns and no bus stop. With the initially calculated width, the associated turn factors and bus factors are determined and a revised width is calculated. The iteration continues until:

1. Two consecutive widths are not significantly different;
2. It is established that the calculated width is outside of the Capacity manuals acceptable width range;
3. It is established that an oscillation between two specific width values has occurred; or
4. There are 15 iterations.

In calculating service volume or G/C ratio, direct solutions are possible. The Peak Hour Factor (PHF), metropolitan area size, location in metropolitan area, right turn, left turn, and bus factors are determined. The load factor is then calculated. One of three alternative output may result depending upon the calculated load factor value.

1. If the calculated value is between 0.00 and 1.00, then the load factor value constitutes the output.

2. If the load factor is greater than one there is basically no solution. If this situation occurs, the maximum service volume for a load factor of one is calculated, using the initially given width. The computed maximum values for width and service volume are provided in the output.

3. If the load factor is less than zero, additional calculations are made. If this occurs, the maximum service volume for a load factor of zero is calculated, using the initially given width; and the minimum width for a load factor of zero is calculated, using the initially given service volume. The computed service volume and minimum width are provided in the output.

INPUT - A summary tabulation of input values and range limits are provided in Table.

OUTPUT - The program output includes items describing:

1. Input,
2. Termination or modification statements,
3. Results, and
4. An optional summary tabulation.

All input data are repeated in the output. In the event that the program is terminated and no computations are made, the reasons for program termination are given.

SPECIAL FEATURES AND SHORTCOMINGS - ICAP has the following advantages:

1. It can calculate combinations of service volume, approach width, load factor, or G/C ratio for signalized intersection approaches.

2. The programs are interactive, and has a very good and easy editing capability.

3. ICAP has the capability of showing the directory of files.

ICAP has several shortcomings. These are:

1. It requires the printer to be turned on while it is running, or else the program will automatically terminate the session.

2. It utilizes the 1965 HCM, the 1985 HCM makes this program obsolete.

3. It is supposed to run successfully without inputting G/C. This is not possible within TCAP.

4. The logic of creating two separate files (namely ACAP and TCAP) is considered a poor design because the G/C allocation within ACAP does not capture the turning volume. In other words, the user has to combine the left turning G/C value with the through G/C value.

TABLE 2 DATA INPUT SUMMARY OF ICAP

ITEM	DESCRIPTION	VALUE RANGE
1.	TITLE	-----------------
2.	SERVICE VOLUME	0000. TO 9999.
3.	APPROACH WIDTH	10.(1) TO 60.
4.	LOAD FACTOR	0.00 TO 1.00
5.	G/C RATIO	.000 TO .999(2)
6.	PEAK HOUR FACTOR	0.70 TO 1.00
7.	LOCAL ADJUST. FACTOR	0.00 TO 9.99
8.	POPULATION	0001 TO 9999
9.	TYPE OF APPROACH	1 TO 6
10.	TYPE BUS STOP	0 TO 4
11.	LOCATION IN METRO AREA	1 TO 4
12.	NUMBER OF LOCAL BUSES	000. TO 120.
13.	PERCENT RIGHT TURNS	00. TO 99.
14.	PERCENT LEFT TURNS	00. TO 99.
15.	PERCENT TRUCKS	00. TO 50.
16.	CYCLE LENGTH	40. TO 180.
17.	NUMBER OF LANES	0. TO 2.
18.	OPPOSING TRAFFIC	0. TO 999.

(1) Lower boundary depends upon type of approach.
(2) Special Value, can have format of x.xx if coded ''1.00.''

INTERCALC

PROGRAM DESCRIPTIONS - INTERCALC is an interactive program which considers intersection geometry, traffic volumes and traffic signal operation in the determination of isolated intersection performance based upon analytical research performed by Dr. F. V. Webster of the TRRL and upon recent FHWA sponsored research (8).

INTERCALC analyzes the performance at proposed or existing isolated signalized intersections and expresses the results of a given set of traffic and signal conditions in various traffic engineering Measures of Effectiveness terms.

INTERCALC operates on microcomputing systems with the following capacities, features and characteristics:

1. Versions of the program are available for the CP/M 80, MS-DOS and PC-DOS operating systems.
2. CP/M systems require 64K of memory. MS-DOS and PC-DOS systems require a minimum of 128K of memory.
3. The program is operable on systems with one or more disk drives. One disk drive systems require a minimum of 240K bytes of free space. Two disk drive systems require a minimum of 90K bytes of free space per disk.
4. An 80 column printer is required.

INPUT - The following input data are required to run INTERCALC:

1) Intersection name and date.
2) Desired cycle length in seconds.
3) The amount of traffic in vehicles per hour assigned to the particular signal phase movement. Traffic volumes must be adjusted

prior to entry, to reflect conditions such as percent of trucks. Traffic volumes are entered only for traffic movements with separate signal phases.

4) Capacity or Saturation Flow is the number of vehicles which can enter the intersection in one hour under saturated flow conditions for a continuous green display for the sum of all approach lanes.

5) The total minimum time for a phase including Green plus Yellow plus All Red or Walk plus Don't Walk plus all Red as appropriate.

6) The time lost to vehicle movement for each single phase which occurs at the beginning of each green display.

7) The speed at which vehicles approach the intersection under non-congested conditions during periods when the traffic signal is displaying a green indication in the direction of traffic flow.

8) The signal phasing possibilities.

OUTPUT - Intercalc calculates the following items:

1) The interval number in which the signal phase display first occurs.

2) The split time for the intersection is calculated based on the entered traffic parameters and the selected cycle length. The split calculation attempts to balance the degree of saturation on critical approaches to the intersection by adding time to critical movements and subtracting time from non-critical movements, limited by the phase minimum time requirement.

3) The split percent is the split time expressed as a percent of the total cycle.

4) The degree of saturation is a ratio indicating the level of saturation occurring at the approach for the given traffic parameters and cycle length. The ratio is calculated as:

(Volume X Cycle)/(Saturation Flow X (Split time - Lost Time)).

5) The average delay per vehicle in seconds is a calculated measure of intersection performance. The calculation is based upon a modified version for the Webster equation. The delay results are valid for conditions where the degree of saturation is less than 95 percent.
6) The average number of times a vehicle will stop on the intersection approach.
7) The average number of vehicles in a waiting queue for a single lane approach to the intersection when the signal changes from red to green.
8) The probability of clearing the queue of waiting vehicles on each intersection approach. This value is calculated using the Miller equation from the PASSER II-80 program.
9) EX. (EXCESS) FUEL, CO, HC, NOX values in gallons per vehicle and grams per vehicle respectively represent the item quantity in EXCESS of the amount which would be consumed or produced if the control situation did not exist. These values are calculated using the equation and factor values determined by a previous study conducted for the FHWA.
10) The degree of saturation of the most saturated approach at the intersection.
11) The total delay per hour at the intersection, in vehicle hours.

12) The LEVEL OF SERVICE is calculate based on the following Measures of Effectiveness:

a) Total intersection delay,

b) Degree of saturation, and

c) Probability of queue clearance.

The ranges are shown in Table 3.

13) The total stops per hour for all vehicles.

14) TOTAL EXCESS FUEL, CO, HC AND NOX values in gallons and kilograms respectively represent the total item quantity per hour in EXCESS of the amount which would be consumed or produced if the control situation did not exist.

15) Interval diagram illustrating the order of signal displays as they occur in a complete signal cycle. The values are shown in percent of the cycle.

SPECIAL FEATURES AND SHORTCOMINGS - INTERCALC contains the following highlights:

1) It is a menu driven interactive program. It is easy to use, requires minimum input data and produces an effective set of output results. A special feature of the program allows INTERCALC to extract data for an intersection from PASSER II-80 input file.

2) INTERCALC is a computerization of traffic engineering procedures which allows the engineer to perform detailed isolated intersection signal operation analyses considering selection of optimum phasing patterns, geometric changes, left turn phasing, selection of operating strategies and development of detailed traffic signal timing.

Table 3

Delay, Degree of Saturation, and Queue Clearance Logics of INTERCALC

Source (BATHER BELROSE BOJE MICROCOMPUTER PROGRAM USER GUIDE)

DELAY (SEC/VEH)

SOURCE: PASSER II-80

COMMENT: INTERCALC uses the INDIVIDUAL MOVEMENT scale of DELAY values to determine the approach and intersection LOS.

TOTAL INTERSECTION DELAY (Note 1)	INDIVIDUAL MOVEMENT DELAY (Note 2)	LEVEL OF SERVICE
Less than or equal to 16	Less than or equal to 15	A
Less than or equal to 22	Less than or equal to 30	B
Less than or equal to 28	Less than or equal to 45	C
Less than or equal to 35	Less than or equal to 60	D
Less than or equal to 40	Less than or equal to 100	E
Greater than 40	Greater than 100	F

Note 1. From Highway Capacity Manual Interim Guidelines Jan. 1980.
Note 2. From field studies during HPR Study 203 1975.

DEGREE OF SATURATION

SOURCE: PASSER II-80

COMMENT: INTERCALC produces and reports this value and uses the value in the determination of split but does not use the value to report LOS.

DEGREE OF SATURATION	LEVEL OF SERVICE
Less than or equal to 0.60	A
Less than or equal to 0.70	B
Less than or equal to 0.80	C
Less than or equal to 0.85	D
Less than or equal to 1.00	E
Greater than 1.00	

QUEUE CLEARANCE PROBABILITY

SOURCE: PASSER II-80, Miller's Queue Clearance equation.

COMMENT: INTERCALC produces and reports this value as a Measure of Effectiveness but does not use the value to report LOS.

PROBABILITY OF QUEUE CLEARANCE	LEVEL OF SERVICE
Greater than or equal to 0.95	A
Greater than or equal to 0.90	B
Greater than or equal to 0.75	C
Greater than or equal to 0.50	D
Less than 0.49	F

3) Data inputs are similar to PASSER II-80 and it can extract data for an intersection from a PASSER input file.

4) INTERCALC includes a user accessible ''?'' HELP feature from the main MENU. The ''?'' Help text file may be customized by the user to include special procedures and instructions.

5) The phasing possibilities included with INTERCALC are designed to be consistent with PASSER phase designation with several enhancements. Phase references are made using graphical representation rather than either by PASSER phase numbering or NEMA phase numbering. This procedure was selected to minimize the complexity of the phase selection process. Phasing is selected interactively using a set of phase selection screens.

6) It has the capability of showing the list of data files.

The main shortcomings are:

1) INTERCALC requires calculation of saturation flow rate for each approach.

2) It is limited to traffic flow conditions below saturation.

SCA

PROGRAM DESCRIPTION – Signalized Capacity Analysis (SCA) is a spreadsheet program for use with Lotus 123 that incorporates the worksheet calculations contained in Chapter 9 of the 1985 Highway Capacity Manual (9). It is arranged in six sections to look and function much like the worksheets supplied in the capacity manual. To run SCA, the user needs a copy of Lotus 1-2-3 version 1A, an IBM compatible system with at least 256K memory, a double sided disk drive, and an Epson compatible printer to print the output.

INPUT – Once the program is loaded, a help menu is displayed at the top of the screen containing the following functions:

- Input : to input or change data
- GoTo : to move to a specific worksheet
- Header: to input or change header information
- Cycle : to change the signal cycle parameters
- Print : print one or all worksheets
- Save : to save a worksheet
- Recall: to recall a worksheet
- Quit : to Quit SCA

Two tables are provided for ''Input;'' one is related to roadway descriptions and the second is related to traffic.

The input parameters needed for the first table by lane groupings are as follow:

- Grade
- Percentage Heavy vehicles

- Parking data
- PHF
- Pedestrian data

The data inputtsd to the second table are:

- Movement volumes
- Number of lanes
- Green time
- Proportion of left turn
- Proportion of right turn
- Lane width
- Permitted and protected left turn options

OUTPUT - The program produces six worksheets, and they are:

1) General Information Worksheet
2) Volume Adjustment Worksheet
3) Left Turn Adjustment Worksheet
4) Saturation Flow Adjustment Worksheet
5) Capacity Analysis Worksheet
6) Level of service worksheet

These worksheets are identical to the worksheets listed in the capacity manual.

PROGRAM HIGHLIGHTS AND SHORTCOMINGS - SCA provides a capacity analysis of signalized intersections using the 1985 HCM. It produces worksheets identical to the ones documented in the capacity manual, which makes it easy for the user to cross check the procedure and the results. The use of spreadsheet concept permits observation of calculations update on the screen.

SCA has some shortcomings and they are:

1. The user has to have a Lotus 123 program to run the program.
2. The documentation of the program needs major improvements.
3. Changes to the original program may be made unintentionally by a user who is not experienced with spreadsheet.
4. The program asks for proportion of right turns and left turns in percent, yet it has to be inputted in fractional form.

SIAP & FREESIAP

PROGRAM DESCRIPTION - SIAP, the Signalized Intersection Analysis Program, is functionally similar, but more powerful than FHWA's SOAP84 program. SIAP allows a rapid evaluation of an existing or alternative signal timing control plan for an isolated intersection (10). SIAP can select cycle length, green splits, and dial assignments for up to 48 time periods. There are 196 possible phase sequence combinations that can be evaluated. Both permissive and restrictive left turn treatments can be analyzed. As compared to FHWA's SOAP84, SIAP permits partial control over the green split optimization search to favor a particular street and/or specifying a targeted limit on the maximum degree of saturation allowed. In addition, approach speeds varying from 30 to 50 mph can be entered and affect left turn saturation flow rates and excess fuel consumption.

FREESIAP provides a number of powerful commands to ease the task of SIAP input data preparation. The most important of these is the HELP command. The HELP command gives access to FREESIAP's input prompts for any SIAP card type or FREESIAP's other commands. The commands allow listing of files to screen or printer; modify an individual line; perform a replace, delete or insert edit command; show more detailed information about some of the SIAP cards or terminology; save the free and/or fix format files to disk; and stop or restart FREESIAP. Commands may be executed at any time. Some commands permit exiting the command simply by entering a RETURN.

INPUT - The required order of SIAP input data is relatively simple because input data processing is controlled by the SIAP card type name.

BEGIN. . . . .

```
  *

  *

  SIAP data specifications

  *

  *

RUN

  * --------------------------------|
  *                                 |
  *                                 |
  alternative data specifications   }  repeatable option
  *                                 |
  *                                 |
RUN --------------------------------|

COMPARE        (Optional)

END
```

Once a SIAP card has been read, the contents of the card will remain in effect until over-ridden by another SIAP card of the same type, thereby permitting alternative analyses with minimal data input.

When FREESIAP prompts for data or command, simply enter the card type name and data field in free format. The rules for entering SIAP data in free format follow:

1. Enter the SIAP card type name followed by the data field separated by commas.
2. If a data field is skipped, be sure to use a comma in its place.
3. After any given data field, all remaining data fields may be skipped if defaults are acceptable to you.

4. Blanks are not ignored.

5. Do not use commas in comment fields.

FREESIAP detects a wide range of user input errors and responds in a forgiving manner by providing the correct format or specifying what was incorrect and then displays the standard enter data or command prompt. Furthermore, FREESIAP contains an Intraline Editor which is entered with the MODIFY command.

SIAP input data and commands are listed below with a short description of their functions.

- BEGIN - initializes program and global parameters
- CAPACTIY - enters the number of lanes or saturation flow
- CASE - enters title used in conjunction with COMPARE
- CHECK - activates syntex checking only
- COMMENT - documenting or describing input data
- COMPARE - activates comparison for runs with the CASE card
- CONTROL - specify operating parameters for controller
- DENSE - changes internal line counter for 8 lines per inch printers. Affects plots only.
- END - terminate the execution of SIAP
- GROWTH - enters growth factors for traffic volumes
- HEADWAY - specifies saturation headway for each movements
- LEFT - specifies left turns on clearance interval and left turn treatments
- MAXX - targetted maximum degree of saturation used in the green split optimization procedure for pretimed control only
- MIN

  MINNS

MINEW

MINOFF - controls green split optimization in coordination with MAXX for pretimed control only

- MINGREEN - specifies minimum green time by movements
- NOTIMING - cancels preceding TIMING card(s) and allows optimization occur
- NOWARN - suppresses most but not all warning messages
- OPTION - activates or deactivates options. There are currently two options:

  Option1: prints intermediate results of green split optimal search

  Option2: prints intermediate results of cycle length optimization search
- PLOTOFF - requests (or turns off) SIAP printer plots
- RUN - initiates SIAP analysis or optimization for currently defined conditions
- SEQUENCE - specifies phasing sequences
- TABLES - request tables to be printed
- TIMING - specifies timing and phase sequences for an evaluation
- TRUCK - specifies percentage of trucks and or buses
- VOLUME - specifies traffic volumes by movement
- GRNSPLIT - SIAP initially performs a green split based on the critical movement analysis technique. SIAP then begins a green split optimization procedure controlled by user specified options such as MIN, MINNS, and MINEW and the relationship between MAXX and BIGX.

  MAXX = targetted maximum degree of saturation

BIGX = the largest degree of saturation for any movement

- WARN - reverses the effect of NOWARN card

OUTPUT - Included on the floppy diskette are two batch files and their corresponding data files for easily controlling compressed (17 characters per inch) and normal (10 characters per inch) printing.

SIAP output consists of input echoes; various tables if requested; a left turn capacity check; an overall measures of effectiveness (MOE's) summary; time period specific MOE's summary; plots of saturation and delay versus time; and a case comparison if requested. MOE's printed include delay, stops, excess fuel consumption due to stops and accelerations, excess lefts greater than capacity, and maximum queue.

SPECIAL FEATURES AND SHORTCOMINGS - SIAP FREESIAP has the advantage of doing multiple runs. It has the graphic capability of showing the phasing patterns, minimum degree of saturation, average delay, and cycle length. It also is capable of comparing different runs within a simple table form.

SIAP evaluates actuated signal control, however, maximum green time and unit extension are not considered. Cycle lengths and green splits are estimated for average values. By the nature of their design, actuated signals, in general, result in high degree of saturation for the critical movements at both low and high volumes. Most delay equations contain a term for estimating random (and possibly over-saturation) delay. At high degrees of saturation and low volumes, these random delay terms greatly over-estimate delay. A very rough correction for this problem has been added to SIAP for actuated control. This over-estimate may also occur for pretimed control, but only under unusual conditions. Again by their design, actuated signal control results in left turn capacities slightly greater than demand with restrictive left turn treatments unless constrained by maximum green time settings and general intersection over-capacity. For a rough estimate of approach capacity, the user is advised to set minimum cycle time equal to maximum cycle. ''Warning,'' SIAP's and SOAP's signal timing and delay estimation for phasing sequence T (single phase) under actuated control may produce unexpected values due to the complexity of modeling left turn and opposing through traffic conflicts with an actuated signal design. The user is advised to use judgment in evaluating the SIAP output for phase sequence T under actuated signal control.

SICA

PROGRAM DESCRIPTION - SICA is an acronym for Signalized Intersection Capacity Analysis. This computer program was developed for the IBM PC and ''IBM compatible'' microcomputers. The program is written in Microsoft Advanced Basic (11). SICA is intended for use by traffic engineers who are analyzing signalized intersections to determine average stopped delay per vehicle and level of service.

This program is a computerized reproduction of the methodology adopted as Chapter 9 of the 1985 edition of the Highway Capacity Manual (HCM). The program neither changes nor modifies in any way the procedure or parameters used in the HCM methods. While it is expected that an experienced traffic engineer could use this program and interpret the results without knowing the details of the HCM procedure, it is strongly recommended that program users thoroughly familiarize themselves with the HCM.

INPUT -

1) Descriptive Information
2) Turning Movement Volumes per approach
3) Pedestrian Volumes per approach
4) Arrival Type per approach
5) Peak Hour Factor per approach
6) Percent Heavy Vehicles per approach
7) Number of Buses per approach
8) Percent Grade per approach
9) Number of Lanes per approach
10) Lane Group Configuration and Geometrics
11) Parking Information

12) Signal Information including Signal Type and Cycle Length, and

13) Signal Timings for each existent lane group

OUTPUT - The printed output consists of three pages.

The first page provides a copy of all the input data for reference purposes, especially useful for analyzing the same intersection under a variety of conditions.

The second page gives the results of the analysis in terms of the v/c ratio, average stopped delay, and level of service for each lane group; delay and level of service for each approach; and delay and level of service for the intersection as a whole.

The third page provides an additional output information per lane group per approach. These are:

- Adjusted Flow (V)
- Proportion Left Turns (PLT)
- Proportion Right Turns (PRT)
- Adj. Saturation Flow Rate (S)
- Flow Ratio (V/S)
- Green/Cycle Ratio (G/C)
- Lane Group Capacity (C)
- Volume/Capacity Ratio (X)

SPECIAL FEATURES AND SHORTCOMINGS - The following special features are built within this program:

1) The default values provided by the HCM for Pedestrian flow, percent heavy vehicles, peak hour factor, grade, number of buses, number of

parking maneuvers, and arrival type can be used by the program without inputting the data.

2) The program uses the Lane Group Configuration Type shown in Table 4.

3) When the V/C ratio for a given lane group is greater than 1.2, the output form will not show a delay; rather, the lane group, approach, and intersection will be labelled OVERSATURATED. The V/C ratio is printed out so that the analyst can see the degree of oversaturation.

4) It can handle pre-timed, fully actuated, and semi-actuated signals.

SICA has the following shortcomings:

1) It has poor diagnostic messages.

2) It requires the percent trucks in decimal, for example 5 percent trucks should be inputted as 0.05. If the user inputs 5 it will treat it as 500 percent and it will not detect this error.

3) It is not clear whether green time per lane group includes yellow or not.

Table 4 LANE GROUP CONFIGURATION TYPES OF SICA

LEFT TURNS		RIGHT TURNS
TYPE #	LANE GROUP DESCRIPTION	TYPE #
1	EXCLUSIVE TURN LANE PROTECTED PHASING	9
2	EXCLUSIVE TURN LANE PERMITTED PHASING	10
3	EXCLUSIVE TURN LANE PROTECTED + PERMITTED PHASING	11
4	SHARED TURN + THROUGH LANE PROTECTED PHASING	12
5	SHARED TURN + THROUGH LANE PERMITTED PHASING	13
6	SHARED TURN + THROUGH LANE PROTECTED + PERMITTED PHASING	14
7	SINGLE LANE GROUP APPROACH	7
8	DOUBLE EXCLUSIVE TURN LANE PROTECTED PHASING	16

SIGNAL

PROGRAM DESCRIPTIONS - Barton-Aschman Associates, Inc. has developed a package of traffic engineering and related programs which provide quick, accurate and consistent analysis results of traffic and transportation-related engineering problems. These programs are known collectively as TEAPAC which is an acronym for Traffic Engineering Application PACKage. One of the programs in the package is called SIGNAL (12). SIGNAL analyzes individual intersection control needs based upon approach capacity, lane usage, pedestrian and clearance constraints, and multi-phase signal control. The methodology uses the capacity analysis procedures documented in the 1965 HRB Highway Capacity Manual. The user of the program can analyze existing conditions and timings as well as generate optimum signal timings and phasings. The optimizer in the program seeks to establish the least number of phases of control at the best attainable level of service for a given cycle length.

This interactive analysis tool is particularly useful in developing individual intersection control strategies, controller requirements and required timings since an entire day's variation in volumes can be analyzed quickly and accurately. Two-phase, three-phase, multi-phase, and split-phase control strategies are considered, with the ''best'' phasing being identified for each set of input data and operational constraints, such as the acceptable range of cycles. The user can easily scan the output and quickly determine which phasing will handle an adverse situation most efficiently.

Further, this analysis quickly identifies capacity problems and points directly to the cause of the problem. At this point, solutions can be generated to solve the problem in terms of more sophisticated control strategies, revised lane usage, improved roadway geometrics, or any combination of the above. The results of this analysis will provide valuable timing data related to local control which otherwise would be very time consuming to obtain by conventional hand analysis.

The output is presented to allow the engineer to quickly determine the problem movements at an intersection. Data concerning the level of service of each approach traffic lane are given. The amount of available green time used, and the maximum queue length is also given. Alternative solutions can be quickly and easily tried helping to optimize the solution in a relatively short time.

It is suggested that a companion document, the TEAPAC TUTORIAL, be read first, before reading the user manual. The TEAPAC TUTORIAL introduces the user to the basic commands of the TEAPAC control language and gives instruction on its use. Familiarity with the TEAPAC control language will greatly aid the user in learning the commands unique to SIGNAL, since they are used in the same way and share the same basic format.

INPUT - Almost all of the input parameters have defaults which eliminate the need to enter data for every parameter. These defaults are logical and may be seen for any or all commands with the use of the HELP command. However, there remains two commands for which it is necessary to input data. These are the VOLUMES and WIDTHS commands. The remaining input data commands will be grouped by their subject areas which include movement, approach, intersection and analysis. Through the use of these commands, almost any intersection may be specified quickly in a simple, straightforward manner.

Intersection/Area-wide parameters - There are five commands which refer to factors of the intersection as a whole and the area-characteristics around the intersection. These commands are METROAREA, POPULATION, TYPE, PEDTIME and NETWORK.

Approach-Specific Parameters - The next set of commands refer to the three or four approaches to the intersection. These commands are PEAKHOURFACTORS, PARKINGSIDES, BUSVOLUMES, BUSSTOPS, PEDLEVELS, and PERMISSIVES.

Movement-Specific Parameters - VOLUMES and WIDTHS are two of the movement-specific parameters. Five other commands are also movement specific. These commands are LANES, CHANNELIZATIONS, TRUCKPERCENTS, REQCLEARANCES, and MINIMUMS.

Analysis factors - All the analysis factor commands relate to the timings and phase settings for the intersection. Often these commands are used to override settings of parameters which other operations have set. These commands are GREENTIMES, YELLOWTIMES, CRITICAL, OFFSET, LEADLAGS, and PHASEMOVEMENTS.

OUTPUT - The major output reports provided by the SIGNAL program are listed below along with their calculated Measures Of Effectiveness:

1) Summary of Parameter Values,
2) Capacity Analysis,
    a) Level of Service, and
    b) Maximum Queue.
3) Evaluation of Intersection Performance,

a) Level of Service,

b) Degree of Saturation,

c) Average Delay,

d) Total Delay,

e) Number Stopped,

f) Average Queue, and

g) Maximum Queue.

4) Service volumes,

5) Required G/C's,

6) Successful Sequences,

   a) Level of Service.

7) Sorted Sequences,

   a) Level of Service.

8) Optimum Timings,

9) Sequence Phase Diagram, and

10) Map of Design Hour Volumes.

SPECIAL FEATURES AND SHORTCOMINGS - The program is able to handle the following:

1) SIGNAL is able to calculate the optimum splits and phasing for a given cycle length.

2) It is capable of drawing Sequence Phase Diagram and Map of Design Hour Volumes.

3) It can generate a clear and useful summary of the data inputted by the user and the default values.

4) It can provide a capacity analysis summary for the intersection and all individual approaches of intersection.

5) It is able to report on a number of performance measures for each traffic stream, as well as the roadway conditions and capacities of each stream.

6) It can calculate the service volumes which are available in each traffic stream at each level of service of operation.

7) It can test every available phasing at a specified cycle length in an attempt to make phasing work at the highest level of service.

8) It enables the user to identify the needed parameters using the ''HELP'' command.

The following shortcomings were observed within this software:

1) In spite of the ability of the software to do so many different types of analysis and evaluation, the program is not very user friendly.

2) The documentation is not clear.

3) It takes time to master the software.

4) It requires capital letters.

5) According to the manual, the program is supposed to be able to choose optimum cycles, however it doesn't have that capability.

6) It utilizes 1965 HCM, the 1985 HCM makes this program obsolete.

SIGNAL85

PROGRAM DESCRIPTION - SIGNAL85, the 1985 Highway Capacity Manual software for signalized intersections is one of the programs available in TEAPAC from Barton-Aschman Associates, Inc. (13).

The purpose of SIGNAL85 is to analyze individual signalized intersections in terms of capacity adequacy for a given set of approach volumes. Factors such as approach width, signal phasing, signal timing, cycle length, pedestrian walk time constraints, controller requirements and many other constraints are all considered in the analysis.

SIGNAL85 can be used to analyze a pre-determined condition or calculate new and optimum timings given a set of constraints. The program can also automatically select an optimum timing and produce an analysis of it.

It is suggested that a companion document, the TEAPAC TUTORIAL, be read first, before reading the signal document. Familiarity with the TEAPAC control language will greatly aid the user in learning the commands unique to SIGNAL85, since they are used in the same way and share the same basic format.

INPUT - The required data to analyze the intersection capacity by approach are divided to the following groups of parameters.

Intersection Parameters

Includes the location within the metropolitan area, population of the metropolitan area in thousands, intersection type, and time allocated to any exclusive pedestrian phases.

Approach Parameters

These parameters include peak-hour factors, parking conditions on each leg of the intersection, pedestrian interference levels for right turns on each approach, and control status of left turns on a green ball following an exclusive phase (permissive left condition). Local bus frequency and stop information is also included.

Movement Parameters

Traffic volumes and truck percentages for the 12 possible movements in the intersection, lane width and usage data for each movement; also, channelizations for turning lanes, minimum timings for green indications and clearance intervals are included within this category.

Analysis Parameters

Timings for each phase (green times and clearance interval times), and critical movements for each phase are inputted within this category.

Design Parameters

Design level of service identifies the desirable operation level, the allowable range of cycle checking, the phasing sequence, and excess data (specifies how any excess green time is assigned to the phasing during the design procedure) are inputted in this section.

OUTPUT - The major output reports provided by the SIGNAL85 program are listed below:

1. **Summary of Parameter Values** - The intersection capacity parameter summary which is a compilation of data pertinent to analyzing intersection capacity by approach are summarized here using the SUMMARISE command.

2. **Capacity Analysis** - The intersection Capacity Analysis report, generated primarily by the ANALYZE command, provides a capacity analysis summary for the intersection and all individual approaches of the intersection.

3. **Evaluation of Intersection Performance** - The evaluation of intersection performance generated by the EVALUATE command is a report designed to summarize a number of performance measures for each traffic stream. The emphasis is more on the estimated level of each stream's performance, rather than on the parameters which were used to calculate the level, as in the case of the capacity analysis worksheet.

4. **Service Volumes** - The Service Volumes report calculates the service volumes which are available in each traffic stream at each level of service of operation. The service volumes are given in vehicles per hour of green, making the assumption that the G/C allocated to the movement is 100 percent. This procedure identifies the maximum service volumes which can be obtained from each traffic stream.

5. **Required G/C's** - The Required G/C's report generated by the GOVERCS command, identifies the green to cycle time ratios which are required by the demand volumes in each traffic stream to maintain each level of service of operation. Thus, if a given movement were to receive exactly the G/C shown for a level of service, that movement would operate at exactly that level of service.

6. **Successful Sequences** - While the program is proceeding with the operational design of each of the specified phasings, the Successful Sequences report is produced by the DESIGN command to inform the user of the progress of the design. Each phasing is tested at all specified cycle lengths in an attempt to make the phasing work at the highest level of service. The first time a phasing is successfully designed at a given level of service, the phasing code is printed along with the level of service, the G/C + Y/C represents the proportion of the cycle which is required to make the critical movements operate at exactly the specified level of service; therefore, the lower the G/C + Y/C required, the more probable is a successful operation of the sequence.

7. **Sorted Sequences** - The Sorted Sequences report generated by the SORT command, provides a valuable tool which can be used in making phasing selections and cycle selection. This report is also an optional output of the DESIGN command. The sorted order reflects a very basic way of identifying which phasings have the highest probability of being successful--those with the lowest overall requirement of cycle time at the best level of service. This is the order in which the sequences are sorted.

8. **Optimum Timings** - The Optimum Timings report generated by the TIMINGS command provides a list of the G/C + Y/C phase times which are required by the critical movements to maintain operation at the designed level of service. These phase times are shown in seconds per second for the cycle length shown and are totalled in the last line. The required phase times are then adjusted so that the total G/C + Y/C equals 1.000 by allocating excess time to the defined priority movements.

9. Sequence Phase Diagram - This diagram, generated by the DIAGRAM command, provides a graphical interpretation of a specified phase sequence code number. This diagram is also an integral part of the output for the ANALYZE and EVALUATE commands, and is an optional output of the TIMINGS commands. Each phase is designated by a square box with arrows inside indicating the movements which are allowed during that phase.

10. Map of Design Hour Volumes - The Map of Design Hour Volumes generated by the MAP command, provides a worksheet document which shows the relative positions of the 12 turning movement volumes on a schematic diagram of the intersection. In addition to the 12 turning movement volumes, the diagram displays the total volumes of traffic entering and exiting the intersection on each leg.

SPECIAL FEATURES AND SHORTCOMINGS - The output is presented in a way that allows the engineer to quickly determine the problem movements at an intersection. Data concerning the level of service of each approach lane is given. The amount of available green time used, and the maximum queue length is also given. Alternative solutions can be quickly and easily tried, helping to optimize the solution in a short time.

SIGNAL85 has adapted the new Highway Capacity Manual 1985 methodology and is able to do different types of analyses.

According to the user manual the next version of the program will be able to calculate the Optimum Cycle length from a given range using the Webster Delay Model.

One of the few shortcomings with this software is the documentation. It is the old SIGNAL documentation, and it takes a long time to master the software. Furthermore, the program is so structured that a new user would feel that the program is unfriendly, and that a relatively long training period is needed to utilize its fullest potential.

SOAP 84

The Florida Department of Transportation and Federal Highway Administration have recently developed a computer model that provides a valuable tool for examining a wide range of intersection signal design alternatives and selecting the best alternative.

SOAP, which is an acronym for Signal Operations Analysis Package, is a traffic signal controller optimizing tool which enables the user to design the signal timing for any three or four legged intersection. SOAP will determine the optimal cycle length, phasing pattern and left-turn configuration for isolated intersections. The user may preselect any of the design parameters if he chooses, or allow SOAP to determine them by an optimization algorithm. SOAP can analyze present timing as well. Since the model has this dual capability--design and analysis--it can be used as an evaluation tool to compare the relative effectiveness of alternative control strategies (14).

MODEL DESCRIPTION

The Signal Operations Analysis Package (SOAP) was designed and written by the University of Florida Transportation Research center. The program was written in Fortran IV on an IBM 370/165 computer system. The program consists of over 11,000 card images. Almost one half of these are actual Fortran code with the remaining lines used for Program documentation.

This program requires 202K bytes of computer memory. During the development phase the program has been run using IBM FORTRAN G, H-extended and WATFIV compilers. A version is also available for Burroughs computers. The current program is a stable and reliable version and should be free of errors. The program should be ready to run on most IBM systems with some changes required for other systems.

Execution time will vary considerably depending upon the time periods, type of control and use of progression analysis features. Typically, on the IBM 370/165, an execution time of two or three seconds may be required. More detailed information of the model program is found in the Programmer's Manual.

INPUT REQUIREMENTS

The developers of the model have provided a program which can be run with only the normal information gathered by typical traffic engineering agencies. Provisions have been made for the user to modify the default values built into the program to reflect local conditions.

A standardized format for all input data is used to simplify the coding as much as possible.

There are three types of inputs which are required. These are:

Type 1 - Instruction cards which tell SOAP what to do,

Type 2 - Parameter cards which tell SOAP how to do it; and

Type 3 - Data cards which supply the input variables for the intersection under study.

Data may be coded and submitted to the computer as a single run or for multiple runs.

SOAP input data may consist of an original data deck for a given intersection with multiple runs for evaluating alternatives. In addition, multiple intersections, or problems, may be included at the user's discretion.

Instruction Cards

It was noted earlier that multiple runs can be accommodated by SOAP. This does not mean that data requirements become overly burdensome. There are three levels of a complete execution:

1. A ''job'' which is the complete execution
2. Problems, which are completely separate and independent analyses, but stacked for convenience to avoid multiple job executions; separated by BEGIN cards; and
3. Runs within a problem separated by RUN cards

The key instruction cards are thus the BEGIN, RUN and END cards. The BEGIN card clears all data arrays and commences a completely new problem. When a RUN card is encountered, SOAP begins execution and outputs all reports requested prior to the RUN card. It then looks for either another BEGIN card (to start a new problem), a COMPARE card (to insure that the previous run is included in the comparison) or an END card to terminate execution. If none of these is encountered (including the card following a COMPARE card) SOAP will begin to accept changes to the current data in preparation for the next run.

Parameter Cards

The parameter cards follow a BEGIN card. These four cards (PATTERN, LEFTURN, CONTROL, and LINK) establish the signal patterns, left turn sequence, the controller dial settings, cycle lengths and coordination data. All are optional and SOAP either has default values or will produce the parameters internally. Additionally, the EXISTING data card has parameters similar to the PATTERN card.

With multiple phasing and sequencing, there can be up to eight phases and these may be sequenced in many combinations, or patterns. To understand how to use the PATTERN, LEFTURN and EXISTING cards, it is necessary to know precisely how SOAP interprets several traffic engineering terms, specifically ''phase,'' ''pattern,'' and ''sequence.''

1. Phase is a unique green display which authorizes only certain movements to occur. Typical phases are shown in Figure 2.
2. Pattern is the combination of phases for the north-south (N-S) and east-west (E-W) directions. For example, in Figure 2, the N-S pattern consists of phases 1, 2, and 3, as indicated at the top of the figure.
3. Sequence is the complete phasing for the cycle, or phases 1-5 as shown at the top of Figure 2.

To simplify coding of the input cards, a standard terminology for describing phases was developed. The permitted movements are simply named according to their direction, as illustrated in Figure 2.

Data Cards

Eight data cards exist, but only two (VOLUME and CAPACITY) are required.

The VOLUME card is necessary to establish the traffic volumes for each of the eight movements. A separate card is necessary for each time interval where a volume change occurs. If data are missing for some intervals, the user has the option to allow the program to estimate the volumes by interpolation of values on each side of the vacant interval(s) or to omit the interval(s).

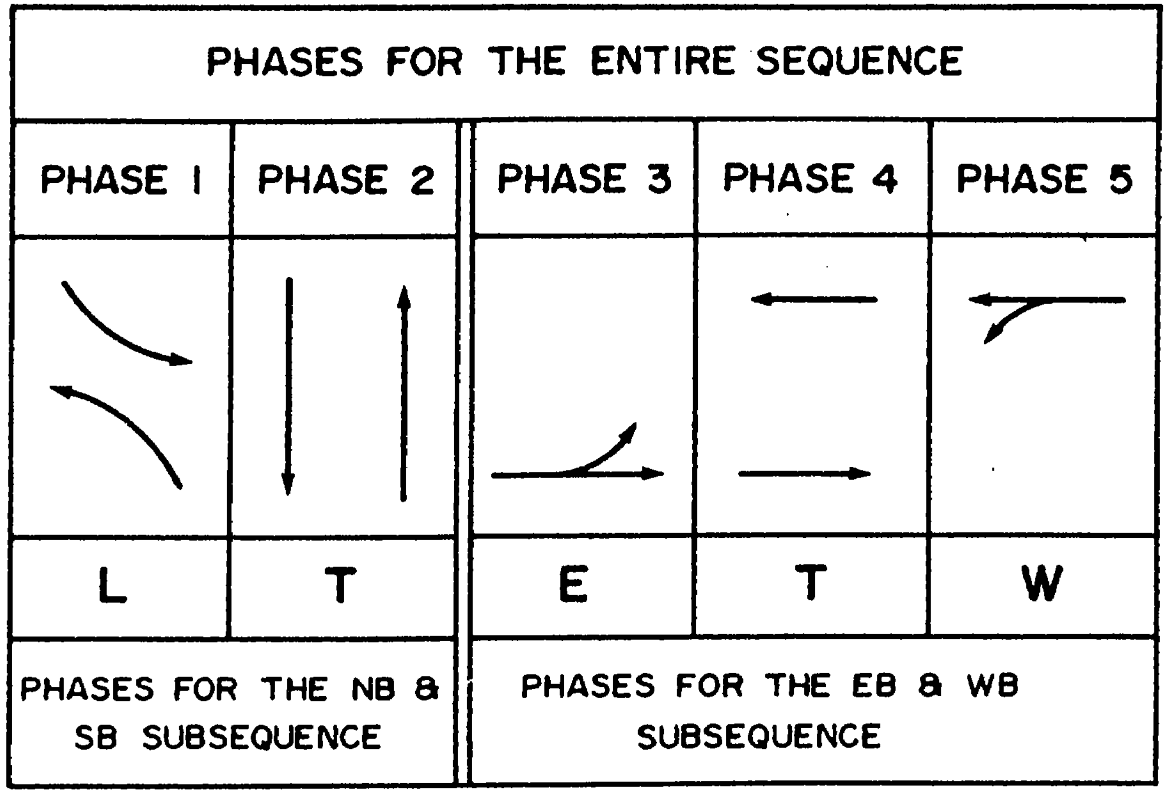

Figure 2 SOAP Phasing Code.

The CAPACITY Card establishes the (maximum) capacity, or saturation flow per hour of green time, given to each movement. However, the user has the option of coding the number of lanes and the saturation flows will be estimated using the departure headways provided in the HEADWAY Card. The number of lanes should be coded as a decimal number (e.g. 2.1) to permit the user to adjust saturation flows for narrow pavements and other restrictions.

Although the HEADWAY Card is optional, frequent users of SOAP will find it desirable to conduct headway studies for their area.

The EXISTING Card is optional but can be used to input existing signal timing.

The MINGREEN Card is also optional but should be used when minimum green times for pedestrian crossing are different than the default values. The default values are 10 seconds for protected left turns and 15 seconds for through movement for pretimed signals and zero seconds for actuated signals.

The traffic volumes can be adjusted to reflect trucks and buses by use of the TRUCKS Card. The program converts the percent of trucks and buses to equivalent passenger vehicles by multiplying by a factor 1.6.

The GROWTH Card can be used to update old data or to reflect projected changes in traffic volumes. The user can apply factors to each movement to reflect these changes.

OPERATIONAL SUMMARY

SOAP has three inherent functions, namely, design, analysis, and operations. To design signal timing it is necessary to configure the intersection and input the appropriate data. SOAP then produces all legitimate phasing patterns. It internally analyzes each pattern and selects the ones which can be executed using the minimum amount of green time.

The next step is dial assignment and timing. A typical controller provides three dials which allow up to three timing patterns to be implemented. SOAP can handle up to six such patterns. The user must decide how many patterns are to be used at a given intersection and assign them to the appropriate dial (control period). If any pattern is unassigned, SOAP will do so, based on the traffic demands. If actuated control is desired, no pattern assignments are made and SOAP makes its computations accordingly.

Cycle length is the most difficult design element to determine. This is a particularly complex problem when several control periods are to be designed. However, SOAP produces these quickly, based on the volumes, capacities and several other parameters. A trial and error optimization procedure is used to find the cycle length which produces the minimum total delay, subject to constraints which govern the amount of queueing which can be tolerated.

Analysis is accomplished by computing the various measures of effectiveness, MOE, which are:

- delay,
- stops,
- excess fuel consumption,
- degree of saturation, and
- left-turn conflicts.

This allows the user to quantify the effects of either the designed control strategy, or if desired, any explicit scheme he wishes to analyze.

Evaluation comes in the comparison of several alternative schemes. Comparisons can be produced by SOAP automatically or the user may make them off-line, manually.

COMPUTATIONAL ALGORITHMS

The MOE's produced by SOAP were identified above. The computational algorithms to compute these measures are discussed in the following paragraphs.

Delay is calculated using the well accepted Webster's method for unsaturated flow under fixed-timed operations. The Webster model has three components. The delay due to uniform arrivals is expressed as:

$$D_1 = \frac{C(1-\lambda)^2}{2(1-\lambda X)}$$

D_1 = delay due to uniform arrivals (sec/veh),

C = cycle length (sec),

λ = the proportion of green time given to the movement (effective green time / C), and

X = the degree of saturation of the movement (V/C).

The delay due to random arrivals, D_2, is,

$$D_2 = \frac{X^2}{2v(1-X)}$$

where v = volume (veh/sec) and the rest as shown above.

An adjustment factor, D_3, is,

$$D_3 = -0.65 \frac{C}{V^2}^{1/3[X(2+5\lambda)]}$$

which was developed empirically to provide a better mathematical fit to field studies. Webster's delay increases infinitely as the v/c ratio approaches 1.0; therefore Webster's formula is only practical to use up to v/c = 0.975. For high degrees of saturation, the modified Webster's delay model, used in the TRANSYT signal network design and analysis program, is also used in SOAP. In this delay model, the first term of Webster's model was retained, and the second and third terms were replaced by a single term of the form:

$$D_2 + D_3 = \left[\left(\frac{B_n}{B_d}\right)^2 + \frac{X^2}{B_d} \right]^{1/2} - \frac{B_n}{B_d}$$

where: $B_n = 2(1-X) + XZ$

$B_d = 4Z - Z^2$

$Z = (2X/v)*(60/T)$

v = approach volume in vehicle per hour

T = period length in minutes (usually 60 minutes)

For actuated control, no reliable delay model existed and this problem is extremely complex. The approach used in SOAP was to modify Webster's model. The actuated control strategy is assumed to:

a) Distribute the available green time in proportion to the demand on the critical approaches, and

b) To minimize ''wasted'' time by terminating each green interval as soon as the queue has been served.

This approximation simulates a ''well timed'' actuated controller. To achieve the results calculated by SOAP, it is therefore necessary to avoid excessively long initial and extension intervals.

The cycle length calculated by SOAP uses the Webster's method also. For fixed time operation the optimal cycle length, C_o, is,

$$C_o = \frac{1.5L + 5}{1-Y}$$

where: L = sum of all lost time due to starting and stopping critical movements, and

Y = overall degree of saturation (i.e. the proportion of green time required for the movement of traffic)

For actuated control the ''cycle length'' is the average cycle length which ensures all excess time is dissipated in the starting and stopping process, or 1 - Y. Therefore, the average cycle length, C_a, is simply $1.1L/(1-Y)$. In the low to moderate demand range, will always be lower than C_o and the difference is slack time necessary to provide for the stochastic variation in demand.

The proportion of vehicles required to stop, P_s, is equal to the number of vehicles joining the queue while it is still discharging, all divided by the number of arrivals per cycle, or:

$$P_s = \frac{rs}{c(s-v)}$$

where: r = length of red (sec.)

s = saturation flow during green (veh/sec) and the rest as before

Excess fuel consumption is computed from the percentage of stops as follows:

$$E_s = \alpha v P_s$$

where: E_s = gallons of fuel consumed due to stops (gal/hr)

α = fuel consumption rate (gal/stop)

v = volume (veh/hr), and

P_s = percent of stops

The excess fuel consumption due to delay, E_d, is:

$$E_d = \beta v d/3600$$

where: β = fuel consumption rate per veh-hr of idling

d = average vehicle delay (sec/veh)

and of course total consumption, E, is the sum of E_s and E_d. The Claffey data indicate that a reasonable value for idling fuel consumption is approximately 0.6 gallons per hour.

The v/c ratio is a reflection of the degree of saturation of the intersection. For an individual approach the degree of saturation, X, is found by:

$$X = \frac{v}{\lambda S} = \frac{v}{s}$$

as previously defined.

Left-turn conflicts occur when left turns are permissive, or not exclusively protected. The measure of effectiveness is the number of left turns which cannot be accommodated safely. Since protected left turns have no conflicts, none are computed. When the turning vehicles may cross traffic there must be sufficient gaps in the oncoming traffic. Based on data derived from simulation using the NETSIM model, curves that relate opposing volume to left turn saturation rate were developed and they are used by SOAP 84. The left turn saturation flow rate is described by the following two conditions:

1. Single lane opposing flow:

$$S_L = 1404 - 1.632\ V_o + 0.0008347\ V_o^2 - 0.0000002138\ V_o^3$$

2. Multiple lane opposing flow:

$$S_L = 1393 - 1.734\ V_o + 0.0009173\ V_o^2 - 0.0000001955\ V_o^3$$

where: S_L = saturation flow for unprotected left turns (vph)

V_o = opposing through volume (vph)

Given the opposing flow, the left turn saturation flow is taken from the curve and compared to the left turn demand. Any ''excess'' demand is the number of left turn conflicts. It is recognized that many left turns are made at the beginning or end of the red; thus the left turn conflicts are not necessarily denied their turn, but it is felt that this MOE would indicate when (and where) enough excess left-turn maneuvers may occur that remedial action might be warranted.

OUTPUT REPORTS

There are six types of outputs available from SOAP. Each of these provide useful information to the user.

Input Summary

The input data is echoed prior to execution. Where appropriate, messages are included so the user can verify that the action taken by SOAP was as intended. The liberal use of the comment card will assist the user in recalling the basis for the input data.

MOE Report

For each run a table of the numerical results of the current run is output. General and control strategy information is found above the table. Within the table are the current values of the MOE, namely:

1. Delay in vehicle-hours
2. Percent saturation (v/c)
3. Maximum queue length in vehicles
4. Percentage of stops
5. Excess fuel consumed (due to stops and delays) in gallons, and
6. Left-turn conflicts

All but the last are given separately for the through and left-turn movements for the four directions.

Below this table is a summary of items 1 (also in average seconds/vehicle), 2, 5 and 6 for the entire intersection. To the right of the summary is the phasing diagram.

Design Recommendations

SOAP develops recommended designs based on optimal flow as constrained by input parameters. There are two types of output for recommended designs.

1. Phasing Patterns. When protected left turns are specified for one or more approaches, it is necessary to choose the optimal phase patterns from several alternatives. SOAP determines the best two and three phase patterns for both the N-S and E-W directions. Each of the four possible phase combinations which may result from these choices is analyzed as a separate design configuration so the user may compare the MOE. The phase sequence in each pattern is indicated as either:

a. User specified

b. Determined by analysis of progression characteristics, or

c. Unimportant (i.e., opposite phase sequence equally acceptable)

2. Timing Design. Each design configuration must be optimized in terms of cycle length, splits and patterns before the MOE can be calculated. For each analysis period, the table includes dial number, cycle length and splits. Above the table is general information and control strategy specifications. The ''PATTERN'' entries indicate the possible sequences resulting from the choices available and are interpreted exactly as discussed in the previous section.

When the control is actuated, an asterisk (*) will appear in the DIAL column and the cycle length and splits are average for each period.

Intermediate Calculations Reports

The TABLE and PLOT commands are instruction cards which enable the user to call for output of many tables (or plots) which are maintained by SOAP.

Table options include printouts of 42 different types of tables which indicate either basic parameters (trucks and bus factors, minimum green time, capacities, etc.) or operational measures (v/c ratios, degree of saturation, average delay by period, etc.)

Plot options graphically portray a comparison of two different statistics. Presently, eight plots are available and show such comparisons as cycle length versus period, delay or volumes per period and excess fuel consumption by period.

Comparison Summaries

SOAP may be used to examine several different control strategies at an intersection. Each alternative may generate up to four MOE tables depending on the choice of phasing patterns to handle left turns. To facilitate the comparison of these alternatives, the user may request a separate summary of MOE's following a series of runs.

A second table gives the comparison of the ''best'' case designs. The output is obtained by including a COMPARE card in the input deck (after the last RUN card which is to be included). Cases can be labeled by including CASE card(s) in the deck.

Diagnostic Messages

SOAP contains an extensive library of messages to inform the user of fatal errors in the inputs; to alert the user to potential, but non-fatal, errors; and to advise the user of actions taken by SOAP, such as the use of default values in lieu of data which were not input. There are (4) levels of messages, as follow:

1. 100 level - fatal messages which must be corrected before SOAP can execute. There are a total of 34 errors at this level.

2. 200 level - warning that the user may wish to reconsider some aspect of his inputs. There are 17 of these messages.

3. 300 level - simply informing the user that SOAP took some action as a result, usually, of omitted data cards. There are 19 of these messages.

4. 400 level - these are high level messages that will not generally occur except when the user is highly proficient with SOAP and is getting into the program itself.

The placement of messages generally occurs in the input report at the location where SOAP had to make a decision.

ADDITIONAL FEATURES

The SOAP options are extensive in terms of the design, configuration and control strategies which can be analyzed or optimized. To summarize, the following options are available in SOAP:

1. Analysis vs. design
2. Existing preset timing vs. optimization
3. Pretimed vs. actuated
4. Protected vs. unprotected left-turn
5. Isolated run vs. multiple runs with comparison
6. Preset vs. optimal phase sequencing
7. Preset vs. optimal dial assignments
8. Numerous input data vs. default options
9. Isolated vs. coordinated control
10. Data check without execution

APPLICATIONS AND LIMITATIONS

As stated earlier, SOAP can be used to design and/or analyze any standard traffic control strategy for either pretimed or actuated operations. As such, it is limited primarily in the same areas which the controller itself is limited. The analysis and optimization is clearly based on mathematical approximations of the real world and therefore necessarily cannot take into account any extraordinary or erratic human behavior.

SOAP cannot duplicate fully the logic of intelligent controllers with microprocessor ''brains'' which can be programmed to be extremely responsive to traffic in real time. For instance, the combining of right turns with through traffic in SOAP presents some problems with accurate estimation of capacity. This is not a severe limitation, however, since the very function of these sophisticated controllers is to optimize on a real time basis, but SOAP is a very powerful and realistic off-line design tool for the practicing signal design engineer.

SOAP 84 (MICROCOMPUTER VERSION)

The Signal Operation Analysis Package (SOAP) was developed as a tool for design and evaluation of the operation of a signalized intersection.

The data input scheme is based on 80 column records which are read from a text file. The scheme was originally designed to accommodate standard punched card data entry. The purpose of the data input manager (DIM) is to produce the text file in the required format from data entered interactively using a microcomputer with screen displays for input and editing. This allows one to concentrate on the problem one is trying to solve, instead of worrying about getting the right numbers in the right columns.

In this sense, the DIM is an excellent aid to productivity and creativity. It is important to note, however, that it does not eliminate the need for a thorough knowledge of SOAP's data coding requirements. If one is not familiar with the use of SOAP as a traffic engineering tool, it will be necessary to review the SOAP84 Users Manual (14).

SOAP/M

PROGRAM DESCRIPTION -- SOAP/M is a microcomputer version of the Signal Operation Analysis Package. It is part of the McTRAN system developed by the University of Florida (15). SOAP/M carries out a complete design for signalized intersection timing, including calculation of cycle length and splits. The computational methodology is similar, but not identical, to the methodology used in the original mainframe version of SOAP. SOAP/M permits inputting only one set of design volumes, capacities, etc., whereas the original mainframe version can run up to 48 contiguous time periods, each with its own characteristics.

The approach used in timing signals is based on an iterative process to minimize a weighted sum of delay and stops. Computation of delay and stops is based on the same model using in the TRANSYT program.

INPUT

The program requires the basic parameters, such as:

- Volume
- Capacity
- Percent Trucks
- Peak Hour Factor
- Lost Time per phase
- Phasings
- Minimum green times

OUTPUT

The output of the program includes:

- Optimal cycle length and splits

- Estimates of delay, stops, fuel consumption and annual operating cost.
- Two-page summary report showing the input stream.

SPECIAL FEATURES AND SHORTCOMINGS

SOAP/M has the following special features:

- Phasing optimization is attained by a trial and error procedure.
- Graphical display of the degree of saturation on all movements is provided.
- Annual operating costs can be estimated.
- Multiple runs can be carried out by the user.

The shortcomings of the program are:

- Peak hour factor is given for the intersection and not by the approach.
- The last three features of the traffic data input (Growth Factor, Approach Distances, and Speed) are not operational.
- It does not provide a comparison of results for multiple runs.
- The program is only available for Apple Computers.

TEXAS

The TEXAS (Traffic EXperimental and Analytical Simulation) model was developed for Texas State Highways and Public Transportation (SDHPT) by the University of Texas' Center for Highway Research (16). It is a microscopic model which simulates isolated intersections from two uncontrolled one-way streets to complex intersections with multiphase control, and/or multiple lane movements.

MODEL DESCRIPTION

The model was originally developed using FORTRAN IV. With close to 15,000 lines of executable statements, and it was developed for both IBM and CDC computers. It requires a maximum of 110K octal words on the CDC computer and 210K bytes on the IBM computer.

The model is composed of three major subprograms:

1. The Geometry Processor: it reads geometric data, ''constructs'' the physical intersection, and plots the intersection and prints detailed output.
2. The Driver-Vehicle Processor: it reads input data and ''creates'' the driver-vehicle traffic stream to be used in the traffic simulation.
3. The Simulation Processor: it is a microscopic stochastic simulation model with time scan updating.

INPUT REQUIREMENTS

There are two basic formats for the three processors in the model. The Geometry Processor and Driver-Vehicle processor have the same input format while the simulation processor has its own separate input format.

Four basic types of data must be inputted for the simulation processor:

1. Geometric information related to the intersection including number of approaches, number of lanes, lane width, number of sight distance restrictions, etc.
2. Traffic data such as volumes, speeds, types of headway distributions, etc.
3. Types of vehicles to be included in the simulation, vehicle length, deceleration and acceleration rate, maximum velocity for each vehicle class, minimum turning radius, etc.
4. Lane control parameter such as type of traffic control, number of intervals in signal cycle, interval length, initial interval, vehicle interval, detector locations, etc.
5. Types of drivers, and perception-reaction time for each type.

OPERATIONAL SUMMARY

As defined earlier, TEXAS is a microscopic, deterministic and stochastic time scan simulation model. The detailed descriptions of the three major subprograms are provided individually below.

Geometry Processor (GEOPRO)

The purpose of the GEOPRO is to describe the physical system to be simulated. The attributes of the system remain constant for any simulation of the physical configuration input. The geometric configuration of the intersection is usually based on the engineering data available from a scaled engineering drawing of the intersection. The only significant restrictions on the geometric layout is that all approaches must be linear, but may approach at any reasonable angle and may have no vertical curves.

Curb radii, vehicle paths and lanes are all realistically flexible and bays (or parking in portions of lanes) can be described as lanes which are available only for specified sections.

After ''constructing'' the geometric layout, GEOPRO determines all allowable vehicle paths through the intersection and identifies all points of conflict. Lane changing within the intersection may be permitted as an option. Maximum speeds, sight distance restrictions and conflicts (including non-crossing conflicts, such as merges or close passing of opposing left-turns) are generated by GEOLPRO. Plots of the intersection and vehicle paths are output by GEOPRO as are printed details and coded data output to tape to be used by the Simulation Processor.

Driver-Vehicle Processor (DVPRO)

The DVPRO reads the same data as GEOPRO, this subprogram is concerned with the preprocessing of driver-vehicle units. The data are generally available from routine traffic studies, and were described earlier. It is primarily in DVPRO where the random, or stochastic variation in the traffic stream is applied. The user may specify the number of driver and vehicle classes (defaults are three and ten, respectively). Driver classes are nonaggressive, normal or aggressive. Vehicle characteristics are length, vehicle operational factor (e.g., sluggish, normal or responsive), maximum acceleration and deceleration rates, maximum speed, and turning radius. Based on the percentage of drivers and vehicles assigned to each of the several classes a driver-vehicle class matrix is generated. The traffic streams (per approach) are generated by randomly assigning the above classes to each individual vehicle to be simulated. Thus, an input ''queue'' is built into arrays and each driver-vehicle unit is fully described in terms of the (mostly) randomly assigned attributes which are:

- o queue-in time (sum of previous headways, or arrival time)
- o driver class number
- o vehicle class number
- o desired speed
- o desired outbound approach number
- o inbound lane number (inbound approach numbers are not randomly assigned)

Table 5 shows the default values used for the various characteristics. A variety of probability distributions are used to assign the above attributes. Output are printed summaries of the input streams and coded data written to tape for use in the simulation model.

Traffic Simulation Processor (SIMPRO)

This subprogram (SIMPRO) is the actual simulation model. Using previously generated data stored on magnetic tape and further card inputs to establish parameters to be used, SIMPRO performs the dynamic activity computations required for the simulation.

SIMPRO handles the physical case of any single, multi-leg, multi-lane, mixed traffic intersection (including split intersections) either without control or with any conventional type of traffic sign or signal control. The model attempts to minimize preparatory calculations and is thus highly user oriented.

The model operates on a time scan basis, where at every time increment (1/2 to one second) the simulated position and operational status of every driver-vehicle unit and (any existing) control status are updated, as

Table 5 - TEXAS Default Driver and Vehicle Characteristics

	1	2	3	4	5	6	7	8	9	10
	Small Car	Medium Car	Large Car	Vans, Mini-bus	Single unit	Semi-trailer	Full trailer	Recrea-tional	Bus	Sports Car
Length	15	17	19	25	30	50	55	25	35	14
Operating Characteristics Factor	100	110	110	100	85	80	75	90	85	115
Maximum Deceleration	8	11	11	8	11	11	11	8	11	12
Maximum Acceleration	8	9	11	8	8	7	6	6	5	14
Maximum Velocity	150	192	200	150	160	160	150	150	125	205
Minimum Turning Radius	20	22	24	28	42	40	45	28	28	20
Percentage Aggressive Drivers	30	35	20	25	40	50	50	20	25	50
Percentage Average Drivers	40	35	40	50	30	40	40	30	50	40
Percentage Slow Drivers	30	30	40	25	30	10	10	50	25	10
Percentage in Traffic Stream	20	32	30	15	.5	.2	.1	.2	.5	1.5

	1	2	3
Driver Class and Type	Aggressive	Average	Slow
Driver Characteristics Factor	110	100	85
Perception-Reaction Time	0.5	1.0	1.5

needed. The degree of updating depends on the likelihood of change. For example, the relative actions of driver-vehicle units are interdependent, thus must be updated at every time increment.

Some events (e.g., interval changes of traffic signal displays) are predictable and times are flagged for updating at the appropriate time increment. With regard to the simulation time increment, the shorter the time, the more accurate the results.

There are two control times of importance to the simulation process. The first is startup time, where the system is started empty and the simulation model proceeds to load the system. No statistics are recorded during this step. The user must input this time since no algorithm has yet been offered to reliably determine when equilibrium has been achieved. The developers have suggested using at least two minutes (simulated real time) for this step.

The second step is the actual simulation time, which is also user specified. Due to the high cost of simulation (despite significant compression from real time), simulation times will normally be short, compared to say field or macroscopic studies. The developers recommend at least 10 minutes to obtain sufficient results for analysis.

The simulation process operates within the above time constraints in a manner very closely approximating the real world. Arrivals are random (due to the stochastically derived headways), decisions are dynamic (e.g., gap acceptance and lane changes are responsive to the immediate traffic environment) and the car following submodel is among the most complex, and realistic, of any existing model. At each instant, the model makes available to the simulated driver his desired speed, destination, present position, speed, acceleration, deceleration (as well as the rate of change of these, referred to as jerk) and the relative positions and velocities of adjacent vehicles. The ''driver'' may decide to maintain speed, accelerate, decelerate or maneuver to turn or change lanes. The decision is dependent on the driver-vehicle characteristics, roadway geometry, traffic control status and the actions of other driver-vehicle units on the system.

COMPUTATIONAL ALGORITHMS

The computational capabilities of TEXAS are complex, particularly in the SIMPRO subprogram. In the interest of brevity, only the more significant algorithms are included in the subsections below.

Geometry Processor (GEOPRO)

Construction of the physical layout of the intersection is based simply on the appropriate connection of required arcs and lines. Of more interest is the technique by which vehicles are tracked through the system. Coordinates are not used. GEOPRO establishes all possible paths through the system and the vehicle positions are stored (in the simulation) on the basis of position in the path. When the end of a path is reached, the vehicle is ''transferred'' to another path (or processed out of the system). these are all based on simple geometric or trigonometric computations.

The most significant computational technique of interest in this subprogram is that for maximum speed on curves. The relationship for maximum speed (V) is as follows:

$$V = \frac{-B + \sqrt{B^2 - 4AC}}{2A}$$

For radii greater than 300m (1000 ft.) the values of A, B and C are as follows:

A = one (1)

B = -15 x radius x (-0.001)

C = -15 x radius x 0.190

For radii less than 300m (1000 ft.) the values of A, B and C are as follows:

A = 1 - (15 x radius x 0.00013951)

B = -15 x radius x (-0.01404)

C = -15 x radius x 0.49671

These are based on AASHTO standards.

Driver-Vehicle Processor (DVPRO)

The major computational function of DVPRO is to randomly assign the various driver and vehicle characteristics. Probability density functions are provided for assigning headways (or arrival times) are the Erlang, gamma, log normal, negative exponential (shifted or unshifted) and uniform. The driver and vehicle classes, inbound lane and outbound approach are generated based on an empirical discrete distribution (e.g., percentages of occurrence for each class). Desired speeds are derived from a normal distribution.

As an example of a stochastic process arrivals, the Erlang probability distribution can be used to represent the waiting time T until the Kth arrival. This distribution is thus the sum of K negative exponential variates. The probability density function is expressed as follows:

$$f(t) = \frac{\frac{K}{\alpha} \times T^{(K-1)}}{(K-1)!\, e^{-T'}} \text{ for } T>0, \text{ and } K>0$$
$$0 \text{ elsewhere}$$

Without developing the entire process, α is equal to the mean divided by the variance of the headways and the Erlang variate, T, is found by

$$T = -\frac{1}{\alpha} \log \left(\pi \frac{K}{1} RN\right)$$

where π = the product of K random numbers (RN).

The empirical discrete probability function is, as the name implies, based on field studies. For example, inbound lane assignments would be based on actual measures of lane distribution. For a simple example, assume a two-lane approach on which P% of the traffic is in lane 1 and (1-P%) in lane 2. The lane assignment, L, is determined for each vehicle on this approach simply by

$$L = \begin{cases} 1, \text{ if } RN \leq P/100 \\ 2, \text{ if } RN > P/100 \end{cases}$$

and P is within the range 0-100.

All characteristic assignments are made similarly, although by a somewhat more sophisticated algorithm to account for greater numbers of characteristics.

Traffic Simulation Processor (SIMPRO)

This is the most important subprogram in TEXAS, as noted earlier. The multitude of algorithms is too complicated to include all of them in this report; thus, only qualitative comments are offered about most of the computations. Only the more salient submodels are defined mathematically.

Acceleration and deceleration are based on empirically validated linear models.

Car following is based on a noninteger, microscopic, generalized car following equation as follows:

$$A_i = \frac{V_i^{\mu}}{(X_{i-1} - X_i)^{\lambda}} (V_{i-T} - V)_i$$

where: A_i = acceleration or deceleration of the i^{th} vehicle

V = velocity (of the i^{th} and (i-1)th, or lead, vehicles

X = location of the i^{th} and (i-1)th vehicles

α, μ, λ = empirically derived constants

The values of the parameters , μ and λ may be set by the user, but suggested values are available. For detailed information, the reader is referred to the TEXAS manual ( ).

Initial speed is based either on desired speed or a speed dictated by the traffic already in the lane ahead, subject to a complicated logical algorithm to determine whether a vehicle should accelerate, decelerate or remain at the initial speed.

Lane control strategies are based on a logical decision process which is dependent on the type of control. Driver responses are determined by traffic control. Driver responses are determined by traffic control, right-of-way and gap acceptance (depending on control type), right turn-on-red and other possible maneuvers. A complex set of algorithms is used for this function.

Lane changes may be optional (e.g., to achieve higher speed) or forced (e.g., a path does not exist from the present lane to the desired outbound leg). All optional lane changes are based on expected savings in delay, but penalties are based on empirical data. Lane changing geometry is also based on empirically validated trajectories.

Operational factors such as driver classification (e.g., degree of aggressiveness) and vehicle classification (e.g., responsiveness) affect the slopes of the speed change submodel and other similar parameters. Perception-reaction times affect the times at which decisions are implemented.

OUTPUT REPORTS

As in the previous sections, the output reports are described separately for the three processors, plus error messages.

Geometry Processor (GEOPRO)

GEOPRO produces printed summaries and plots for inspection and a tape with data for use in SIMPRO. Printed output contain an echo of the input data with convenient column headings and listings of slight distance restrictions, intersection paths and intersection conflicts.

Plots include an overall layout of the system, the intersection detail and vehicle paths for each approach. The composite vehicle paths, showing all potential conflicts can also be plotted. The plots may be interactively displayed on a CRT screen as well, if appropriate hardware exists.

The tape for SIMPRO contains extensive details needed for the simulation. These data may also be written to disk storage if desired.

Driver-Vehicle Processor (DVPRO)

Printed and tape outputs are issued by DVPRO. Some of the same input data discussed above are printed since both GEOPRO and DVPRO use the same input card deck. However, error checks are peculiar to the separate processors.

Traffic Simulation Processor (SIMPRO)

A similar input data echo is issued by SIMPRO, but for the cards input exclusively for this subprogram. Other input data reports also provide more readable formats of the data for the system and the traffic control.

The summary statistics for each approach, as well as the whole intersection, are reported. Traffic control statistics are also output, as appropriate. Finally, the printed output contains summary statistics of the simulation run itself.

Diagnostic Messages

Each processor has its own set of error messages, which are too numerous to list here. The three processors have 59, 62 and 81 input data error messages, respectively. Once an error is detected, it is reported and execution stops. This could result in several runs to ''debug'' the input data.

ADDITIONAL FEATURES

The TEXAS simulation model produces a realistic simulation of intersection operations. The variety of inputs and outputs have been discussed previously and those discussions covered most of the available options. A summary review of these would appear to be warranted, however.

1. Geometry - any feasible design of a single intersection including divided highways which operate under a single signal controller, parking lanes, turn bays and channels.
2. Driver-vehicle units - extremely flexible classifications, all randomly assigned.
3. Turning - lane changes; right and left-on-red; U-turns, protected, permissive and unprotected.
4. Traffic Control - no control; stop or yield sign control; and/or fixed time, semi-actuated or full-actuated signal control. The latter may be based on detector calls set in the pulse or presence modes.
5. Output - printed input data, intermediate results and summary statistics of traffic MOE; line plots of geometrics, turning movements and sight- distance restrictions; and interactive graphics displays.

APPLICATIONS AND LIMITATIONS

The TEXAS Model analyzes a variety of conditions. Alternative geometric strategies, vehicle mixes and traffic control strategies can all be investigated. While separate runs are required for the three main processors, many runs could be made, say, with the Traffic Simulation Processor, using the same outputs of the two preprocessors.

While the TEXAS Model is extremely versatile and powerful, several limitations warrant notice. First is the absence of any effect by pedestrians. All-red signal phases can be modeled for pedestrian intervals at signalized intersections, but the interference to traffic by pedestrians moving simultaneously cannot be simulated.

Approaches must be straight and (essentially) at zero grade. In reality, many intersections have approaches on grades, which affect acceleration and deceleration. This can be compensated for somewhat by using different headway distributions or parameters for the effected approaches, but automatic adjustments would be more convenient.

External preemption of traffic signals cannot be modeled (e.g., bridge, RR or tire preemption).

Finally, there is no provision for coordination, or even the effect of adjacent signals. Nearby signals will clearly affect the arrival patterns, tending to establish platoons. Despite the impressive variety of available arrival distributions, this type of effect cannot be simulated except by direct user input (special vehicles) of driver-vehicle units to DVPRO.

EZ-PASSER

PROGRAM DESCRIPTION - EZ-PASSER is a user-friendly, optimized implementation of the popular PASSERII-80 program.

PASSER is an acronym for Progression Analysis and Signal System Evaluation Routine (17). It is a general purpose mainframe computer program developed by Texas Transportation Institute to assist traffic engineers in determining signal patterns for an arterial street where progression is desired through signals having more than one arterial signal phases. The theory of the program is well documented in HRB Record 445.

EZ-PASSER has the following enhancements over PASSERII-80:

1. User-Friendly Input. The program is window-oriented. The input data are displayed in various windows on the screen all the time. The data can be changed and the update is immediately displayed on the screen. No more batch job input is required. The data can be stored on the diskette or hard disk and retrieved later.
2. Instead of designating numbers to each turning movement, EZ-PASSER uses POSIT-like notations that are easy to understand. It is not necessary to memorize the numbers assigned to each movement.
3. Optimized Program Structure. The program is written in PASCAL and is restructured in a more efficient way that it runs much faster and requires less memory than other similar products. On an IBM PC without 8087 mathematical processor, it takes less than one minute to optimize a seven-intersection arterial.
4. Better Output. Different algorithm is used to generate the time-space diagram so that the progression bands are displayed more clearly. The program also decides the optimum horizontal and

vertical scales for the diagram. The Optimum Solution and Measures of Effectiveness (MOE) tables are more readable.

EZ-PASSER is written in PASCAL language and has been run successfully on the Apple II+ and IBM PC/XT microcomputers. No knowledge of the PASCAL language is required.

The following equipment is needed:

1. An IBM PC/XT or IBM PC compatible microcomputer with MS-DOS (PC-DOS) operating system.
2. At least one disk drive
3. A 132-column printer

INPUT - Much of the input data required by the program is usually available to traffic engineers as intersection distances, progression speeds, allowable cycle lengths, turning movement volumes and saturation flow rates are used. In addition, the traffic engineer can specify a particular arterial phase sequence at an intersection or a set of allowable arterial phase sequences to be analyzed. In case of the latter, each allowable phase sequence is analyzed for the intersection to determine which one gives the best progression.

Since there is only one intersection displayed on the window at any time, FORWARD, BACKWARD, and JUMP commands are designed to allow examination of the data for other intersections. The intersection currently displayed is called Current Intersection. The screen is a window to a string of intersections along the arterial. The intersection has to be current, i.e. on the screen window in order to be modified.

The input data are similar to the one explained in PASSERII-84. For more information refer to description of PASSERII-84 Main-frame version.

OUTPUT - The output is printed in three parts.

1. BEST SOLUTION SUMMARY - The best solution specifies the optimum cycle length, speed, and phase setting for each intersection. It also generates the following:

   - Bandwidth - Indicating the time available for the traffic to flow within the band.
   - Efficiency - The average fraction of cycle length used for progression.
   - Attainability - The average fraction of the arterial minimum through movement greens used for progression.

2. MEASURE OF EFFECTIVENESS REPORT -

   - X Ratio. The ratio of phase time required to phase time available.
   - Delay
   - Probability of Clearing Queue (PCQ)
   - Level of Service (LOS). The accepted criteria for describing the level of service as related to the above MOE are as follows:

LOS	X RATIO	DELAY	PCQ
A	0.60 <	15 <	0.95 >
B	0.70 <	30 <	0.90 >
C	0.80 <	45 <	0.75 >
D	0.85 <	50 <	0.50 >
E	1.00 <	60 <	
F	1.00 >	60 >	0.50 <

PROGRAM HIGHLIGHTS AND SHORTCOMINGS - EZ-PASSER is an attractive, colorful, and creative window-oriented program. It has a useful help file and an easy way of editing the file.

It uses cardinal directions to indicate the vehicle movements entered for each intersection to prevent confusion.

The program has the following shortcomings:

1. EZ-PASSER cannot be used as a preprocessor for PASSERII-80. It only can be used with the accompanying PASSER program in the same disk.
2. It is not capable of adding or deleting intersections in the middle of two already existing intersections.
3. The output does not show the data using the NEMA phasing sequence.
4. The printer has to be on in order to run the program.

NOSTOP

PROGRAM DESCRIPTION - Barton-Aschman Associates, Inc. has redeveloped the original NOSTOP program to provide accurate and useful analyses presented in a more meaningful form (18). This program develops the most productive set of timing plans from the point of view of linear bandwidth progression in an arterial traffic signal system. The program provides a fast and effective means of presenting a graph of the variation of progression efficiency over a complete range of cycle lengths and progression speeds. After the cycle/speed analysis is tabulated, the cycle with the best efficiency is determined. Further analysis provides the optimum system control parameters, which include cycle length, speeds, and offsets. Additional refinements which are provided are:

1) The amount of lead and lag left-turn time available at each intersection without interfering with the through bands,
2) The amounts of through green time unused by the progression bands, and
3) Widening the band in the preferential flow direction.

The analysis provides an accurate, computerized version of band-width analysis which has previously been done manually. Because of the speed of the computer, the analyses can be conducted hundreds of times for each system in order to find the optimum cycle length and progression speed combination for the system. The program thus produces the optimum cycle/speed/offset combination for each signal, which in turn produces the best possible performance of an arterial street system. Once the optimum timings are determined, the program will plot a time-space diagram showing all of the offsets and the progressive bands.

Using these results, combined with the optimum signal splits produced by SIGNAL, leads to a highly efficient operation of all linear systems and isolated intersections within the total road network in a community.

It is suggested that a companion document, the TEAPAC TUTORIAL, be used before reading this document. Familiarity with the TEAPAC control language will greatly aid the user in learning the commands unique to NOSTOP, since they are used in the same way and share the same basic format.

INPUT - Input can be entered either by free format or interactively. To enter the data interactively the command ASK will be used. The function of this command is to request parameter values from the user for each of the listed commands below:

SIZE - Set the number of signals on the route to be progressed.

DIRECTIONS - Set the number of traffic flow directions to be progressed on the route.

CLEARANCES - Set the single clearance interval to be used at the end of all through phases in the system.

BASE - Set the base location to which all offsets will be related, as well as the absolute offset which the base location must have.

UNITS - Set the system of measurement to be used for the units of the input parameters.

LINKNODEDATA - Set all parameter values for an intersection and its adjacent link. There are 12 parameters which describe the characteristics of an intersection and its adjacent link.

- cross street name,
- distance to next intersection to the right,

- directional link speeds to the next intersection,
- through split, in percent,
- all red time,
- available lead/lag inclusion,
- specified lead/lag times,
- offsets, and
- non-concurrent mainline green.

RATIO - Set the desired ratio between the sizes of the directional progressive bands.

ADJUST - Set the factor by which the link speeds established on the SPEEDS command will be adjusted.

FINETUNE - Set the indicator which instructs the design procedure to perform a second iteration of optimization.

TOLERANCE - Set the permitted variation of progression speeds around the desired speeds.

CYCLES - Set the range of cycles to be considered in the design of the progressive systems.

OUTPUT - The major output reports provided by the NOSTOP program are as follow:

1) Table of Efficiency vs. Cycle and Speed - While the program is proceeding with the operational design of the combinations of speeds and cycle lengths for the linear system, the Table of Efficiency vs. Cycle length is produced by the DESIGN command to inform the user of the progress of the design. Each cycle length specified is tested at all speeds specified. The resulting efficiency matrix is the percent of the smallest split in the system that is being used by the through band progression.

Along with the numerical matrix (in percent), a character scale is presented which allows the user to graphically see the results of the progression analysis. The report is useful in identifying what the impact will be of varying the progression speed slightly given a set cycle length.

2) **Graph of Efficiency vs. Cycle** – Upon completion of a design with multiple cycle lengths (and multiple speeds if desired), the Graph of Efficiency vs. Cycle will be produced. This is generated by the DESIGN command if multiple cycles are to be tested. The efficiency plotted on the graph is the bandwidth as a percentage of the smallest split in the system. If there are varying speeds used in the analysis, this report will be preceded by the Table of Efficiency vs. Cycle and speed. Also, the efficiency that is plotted in this graph is for the progression speed that yields the greates efficiency.

3) **Optimum Progression Data** – The Optimum Progression Data generated by the DESIGN command provides all the information pertaining to the optimum operation of a linear signal system. This report is only produced when there is only one speed and cycle combination being analyzed.

4) **Time Space Diagram** – The Time-Space Diagram is generated by the PLOT command. A time-space diagram can be plotted from the offset values obtained directly from the DESIGN command or from offset input by the user on the OFFSETS command. The Time-Space Diagram is useful in visualizing the progression in the system.

PROGRAM HIGHLIGHTS AND SHORTCOMINGS – The major advantages of NOSTOP is that it provides progression in both directions. NOSTOP recognizes double cycle whenever the summation of phases adds up to one half of the cycle length. It has an on-line help file for every command.

In spite of the ability of the software to do so many different types of analysis the program is not very user friendly. Also, the documentation is not very clear and it takes time to master the software. The program is written in the CP/M operating system which utilizes the Baby Blue Card.

PASSER II-80 & PASSLOAD

PROGRAM DESCRIPTION - The PASSER program was prepared by the Texas Transportation Institute, in conjunction with the Dallas Urban Corridor Project, for the Federal Highway Administration in cooperation with the Texas Highway Department and the City of Dallas. The program's theory is presented in the HIGHWAY RESEARCH RECORD NO. 445. PASSER II-84 is a later version of the program which includes enhancements by the Texas Department of Highways and Public Transportation (19).

The PASSER II-80 program analyzes signal performance at proposed or existing intersections, produces signal timing values and a timespace diagram and expresses the results of a given set of traffic and signal conditions in various traffic engineering Measures of Effectiveness terms. For more information reader is referred to the Mainframe version of PASSER II-84 writeup.

PASSLOAD, the data loader program, provides easy entry, storage and management of data files for the PASSER II-80 program. In addition, PASSER II-80 can be run directly from the data loader program thus, the user needs PASSLOAD to enter, edit data sets and to operate the PASSER II-80 program.

PASSLOAD displays a series of data input screens on the computer monitor. Each screen serves two purposes: to display each data item, and to allow the user to input new data values.

On the data input screen, the cursor can be advanced to the next input field without changing the values in the current field by pressing the ''RETURN'' or ''ENTER'' key. Similarly, the cursor can be returned to an earlier input field by pressing the ''BACKSPACE'' key. The cursor also automatically advances to the next input field when the current input field is filled by the user (for example, the user enters three digits in a three-character field).

Pressing the ESC key when a screen is displayed will skip all the entry fields and move to the bottom of the screen. The user will be asked if all the input values are correct. An ''N'' response returns the cursor to the data input screen and to the field from which the cursor left. A ''Y'' response indicates that the user has approved all values for that particular screen. Next the program checks the entered values for obvious errors. If an error is found, an error message is displayed either alone or at the bottom of the data input screen, and the user is required to correct the item. When no input errors are detected on a screen, the program advances to the next data input screen or returns to the main menu.

The PASSER program set operates on microcomputing systems with the following capacities, features and characteristics.

1. Versions of the program are available for the CP/M 80, MS-DOS and PC-DOS operating systems.
2. CP/M systems require 64K of memory. MS-DOS and PC-DOS systems require a minimum of 128K of memory.
3. The program is operable on the systems with one or more disk drives. One disk drive system requires a minimum of 140K bytes of free space. Two disk drive systems require a minimum of 90K bytes of free space per disk.
4. A 132 column printer is required (80 column printer can be used if it is possible to compress the printer).

INPUT - The menu options are listed below, followed by a short description of each:

OPTION 1. ADD PASSER DATA FILE: This option allows the user to input a new data set. This option uses four different data input screens. The first screen allows input of general information pertaining to the arterial. The cross street name and phasing options are entered from the second data input screen. Distance, speed, and queue clearance data between adjacent intersections are entered from the third data input screen. Volumes, capacities, and minimum green times for each of eight movements at each intersection are entered on the fourth screen.

OPTION 2. RECALL PASSER DATA FILE: This option is used to retrieve a data set previously saved. The program allows the user to examine which files are available for retrieval.

OPTION 3. EDIT PASSER DATA FILE: The edit option is used to review or modify data that has been added or recalled. Rather than going through all the steps involved with Option 1, the Edit option lets the user change data on an intersection by intersection basis. In addition, Option 3 allows the user to add or delete intersections from the data set.

OPTION 4. RUN PASSER: This option executes the PASSER II-80 arterial signal timing program using the data set currently in memory. The printer should be available and ready before selecting this option.

OPTION 5. SAVE PASSER DATA FILE: This option allows the user to save the data set currently in memory under an assigned file name.

OPTION 6. DUPLICATE PASSER DATA FILE: This option copies the data from an existing data file to a new data file assigned by the user.

OPTION 7. DELETE PASSER DATA FILE: This option allows the user to erase an existing disk data file.

OPTION 8. LIST PASSER DATA FILES: This option displays files currently available on the disk.

OPTION 9. END PASSER ANALYSIS: This option completes any analysis being done using the data loader program and returns to the computer's operating system.

OUTPUT - PASSER II-80 produces three types of printed report output:

1) A summary and restatement of inputs,
2) The analytical results, and
3) Timespace diagrams (if requested).

The first part of the output presents a summary of the input values arranged into PASSER form. The arterial header is printed before the first intersection. Data entry errors that are encountered by the program, if any, are then printed. Minimum and maximum advisable cycle lengths for each intersection are then printed.

During execution the program will pause for several minutes to actually calculate the signal timing and evaluation results. When complete the results will be printed.

For more information, the reader is referred to the description of PASSER II-80 mainframe version.

PROGRAM HIGHLIGHTS AND SHORTCOMINGS - PASSER II-80 is an arterial signal timing analysis and development program well suited to multi-phase traffic signal situations. Program inputs include link speeds, queue clearances, distances, street names, traffic volumes, saturation capacities and minimum phase times. Speeds may be varied for all links by plus or minus two mph. The program results are a single best, through band solution and an optional timespace diagram.

Four possible phase sequence options are analyzed by the program. For the arterial movements, the program selects the most efficient alternative based on through band efficiencies from the permitted option(s). Cross streets phasing is set in the input data set of data management utilities.

The PASSLOAD program is a menu driven interactive program used to prepare, edit and manage data sets and to run the PASSER II-80 program. It is easy to use, provides extensive menu prompting, and a full set of data management utilities.

PASSLOAD uses a cardinal directions to indicate the vehicle movements entered for each intersection but the outputs will be presented using either NEMA or PASSER numbered phase movements.

PASSLOAD can insert or delete an intersection anywhere within the arterial. The printer has to be on to run the program.

PASSER II (84)

The original model, called PASSER I, was developed by Texas Transportation Institute (of Texas A M University) for use in the Dallas Corridor Project sponsored by the Federal Highway Administration (FHWA) and the Texas State Department of Highways and Public Transportation (SDHPT) in cooperation with the City of Dallas. It was later adapted and expanded as PASSER II for off-line processing and analysis purposes in HPR Project 165, sponsored jointly by the Texas SDHPT and FHWA (20).

A research project conducted by the Texas Transportation Institute entitled ''Reduced-Delay Optimization and Other Enhancements to PASSER II-80.'' The research was sponsored by the Texas Department of Highways and Public Transportation in cooperation with the U.S. Department of Transportation, Federal Highway Administration. The brief six-month research effort was directed toward several topic areas which included:

- Development of a practical procedure which could be used to fine-tune the offsets of traffic signals to minimize total delay and maximize progression of traffic in a progression system.
- Development of a method that can better estimate delay to travel in a nearly saturated traffic system, and
- Development of methods to estimate fuel consumption associated with arterial travel movements in an urban network.

An enhanced version of popularly used PASSER II program - PASSER II-84, which deals with the design and operation of signalized intersections, was programmed on SDHPT's computer system. Program documentation and revised data coding instructions were also prepared.

The computer program is written in FORTRAN IV. The model has been set up on numerous computers with relatively little difficulty. It is estimated that machines with core storage of 92K bytes can handle most problems.

MODEL DESCRIPTION

PASSER II-84 (hereafter referred to as PASSER) is an acronym for Progression Analysis and Signal System Evaluation Routine, version 1984. The basic purpose of the model is to assist in determining optimal traffic signal timings for progression along an arterial considering various multiphase sequences.

The model was designed to calculate all of the signal timing information needed for plan development and field implementation. The program calculates degree of saturation, delay and probability of queue clearance for all movements.

The optimization algorithm of PASSER identifies (from those permitted) the best cycle length, phasing sequence and offsets--best being defined as that combination which results in the greatest bandwidths in both directions of travel. Phase splits are calculated to minimize delay at each intersection.

INPUT REQUIREMENTS

Much of the input data required by PASSER is similar to those needed by the other signal timing programs. They are the same data required for the existing PASSER II-80 program. In order to perform the arterial progression analysis, information is required from each intersection and between intersections which include traffic turning movements, intersection approach saturation flow rate, minimum green times, preferred phase movement, intersection separation distances, progression speeds, and allowable cycle lengths.

Three types of input cards are used for PASSER - 1) arterial header data, 2) Intersection header data, and 3) Intersection detail data.

Arterial Header - This single card is used to describe the arterial signal system under study and defines the general analysis parameters and options.

Intersection Header - One card is required for each signalized intersection to describe the location, connecting link description and signal phasing information.

Intersection Details - Three cards are required for each signalized intersection. Card one is for traffic volumes for each of the movements, card two is for the saturation flow rates for the respective traffic movements and the third card is to establish the minimum phase length for each movement.

OPERATIONAL SUMMARY

PASSER is a macroscopic, deterministic optimization model. The user inputs the minimum cycle length, the maximum cycle length, and the number of seconds the program should increment between the lower and upper cycle length limits. With these data, the program seeks the optimal design by iteratively varying the splits and offsets for each design cycle length and determining the ''bandwidth efficiency.'' The variation of splits is naturally constrained by the minimum green times input. The variation of offsets is about the desired progression speeds input.

It is suggested that for best engineering, the range of cycle lengths be limited to ten (10) seconds between minimum and maximum. This limitation only means that it is necessary to make multiple runs with varied minimum and maximum cycle lengths and minimum movement green times to study a broader range of possibilities.

The model can analyze up to four (4) arterial phase sequences (with or without overlap) per intersection and will select, from those available for consideration, the phase sequence at each intersection that provides the best overall arterial progression.

The user must select one or more of the four basic phase sequences for the arterial permitting the program to select the optimal solution for arterial progression and only one sequence for the cross-street approach. The user has the option to either delete a phase, to specify only one of the specific sequences and/or to permit overlap between phases.

COMPUTATIONAL ALGORITHMS

The developers of the model have combined Brooks Interference Algorithm with Little's Optimized Unequal Bandwidth Equation, and extended them to multi-phase signal operations.

The program first determines the optimal demand/capacity relationships and from these green splits are determined. Trial cycle lengths, phase, patterns and offsets are varied to determine the ''best'' set of timings, i.e., that which maximizes the bandwidths.

The salient computational expressions include the following:

1. Determine Maximum Bandwidth (Bmax) by Direction

$$B_{max} = G_{o_{min}} + G_{i_{min}} - I_{i_{min}}$$

where: $G_{o_{min}}$ = minimum outbound progressive green

$G_{i_{min}}$ = minimum inbound progressive green

$I_{i_{min}}$ = minimum possible inbound band interference optimized subject to upper and lower limits

2. Determine Maximum Band Efficiency (Ec)

$$E_c = \frac{B_a + B_b}{2C}$$

where: B_a = bandwidth in ''A'' direction

B_b = bandwidth in ''B'' direction

C = cycle length

3. Determine Green Time (g)

Green times (including clearances) are determined by a gradient search technique which minimizes delay at the intersection (subject to specified minimum greens). The algorithm shifts the phase change times in small increments until the least calculated delay is obtained. The calculation of delay is discussed later.

The last relationship (the objective function) is the basis of the most significant algorithm used. Some earlier models required that the bandwidths be equal. This is not the case for PASSER, in fact neither direction is automatically favored.

It it is desired to favor one direction, this can be done by use of the minimum percent of progressive bandwidth (Option 1) on the Arterial Header Card or by appropriate adjustments to the desired progressive speeds on the

Intersection Header Card or by adjustments to minimum green times on the Intersection Detail Card. This is subject to the availability of sufficient green time to be absorbed by the ''B'' direction.

4. Degree of Saturation (X)

$$X = \frac{VC}{gS}$$

where: V = traffic volume

C = cycle length

g = effective green time

S = saturation flow rate

5. Estimate of Delay (D)

a) Webster Delay Equation

Analytical estimates of delay are commonly used in many computer models. The most widely used analytical model is Webster's model, expressed as follows:

$$d = \frac{C(1-\lambda)^2}{2(1-\lambda X)} + \frac{X^2}{2q(1-X)} - .65 \left(\frac{C}{q^2}\right)^{1/3} X^{(2+5\lambda)}$$

where: d = average delay on a particular approach, sec/veh

C = cycle length, sec

λ = proportion of the cycle effectively green for this approach (i.e., G/C)

q = traffic volume, veh/sec

X = degree of saturation (i.e., phase volume to saturation flow ratio)

The first component in Webster's model, d1, is the delay due to recurring cyclic demands and stops, called the uniform delay component. The second component, d2, is the random delay component, which adjusts for the random arrival of traffic. The last component, d3, is an empirically derived adjustment, which adjusts the sum of the uniform and random elements to conform more closely with measured delay. Thus, total delay is expressed by the equation $D = d1 + d2 - d3$.

b) <u>NCHRP Delay Equation</u>

In the update of the 1965 Highway Capacity Manual, National Cooperative Highway Research Program (NCHRP) Project 3-28(2) developed a capacity and level of service method for urban signalized intersections. Specifically, a new delay estimation equation was developed to calculate delay and level of service for each lane combination (RT, THRU or LT lane), approach, and the overall intersection under normal, saturated, and over-saturated conditions. The delay equations for the basic delay conditions are summarized in Table 6.

A two-term equation is used in the NCHRP delay equation; namely, terms for Uniform Delay and the Overflow Delay. Uniform Delay (UD) occurs when all queued vehicles clear the approach on each cycle. The UD formula estimates the average stopped delay per approach vehicle for lane groups with a v/c ratio less than or equal to the overflow condition. It is based on uniform arrivals and can be utilized for various analysis period lengths. The Overflow Delay (OD) occurs when on some cycles all queued vehicles clear

TABLE 6 NCHRP DELAY ESTIMATION EQUATION

NEW NCHRP DELAY EQUATION

NCHRP DELAY = UD + OD

$$\text{WHERE UD} = 0.5*\left[\frac{(C)*(1-G/C)^2}{(1-V/S)}\right]$$

$$\text{OD} = 225*F*(X)^2*\left[(X-1)+\sqrt{(X-1)^2+\frac{16X}{V*F}}\right]$$

Where UD - UNIFORM DELAY
OD - OVERFLOW DELAY
C - CYCLE LENGTH
G - EFFECTIVE GREEN TIME
V - DEMAND VOLUME
S - SATURATION FLOW RATE
X - SATURATION FLOW RATIO
F - STUDY PERIOD ADJUSTMENT FACTOR
F = 1.0 WHEN STUDY PERIOD IS 15 MIN.
F = 4.0 WHEN STUDY PERIOD IS 1 HOUR

the approach while on other cycles some of the queued vehicles do not clear the approach due to variation of the traffic volume. Overflow delay is estimated only when v/c is greater than the oversaturated conditions. The Uniform delay component is not affected by the length of the analysis period. The overflow delay, because it is an estimate of arrival variations, is highly dependent on the analysis period. For the convenience of study, a 15 minute analysis period was assumed in the application of PASSER II-84.

The NCHRP delay estimation equation is valid for a v/c ratio above 1.00.

6. Probability of Queue Clearance (P)

The probability of the queue clearing in the available time is calculated by Miller's method ( ):

$$P = 1 - e^{-1.58\phi}$$

where: e = the natural base of logarithms

$$\phi = [(1-x)/x] - (sg/3600)^{1/2}$$

where: s = saturation flow rate

g = green time duration

7. Fuel Consumption Model

The fuel consumption routine used in TRANSYT-7F was modified and applied to PASSER II-84. The basic form of the fuel consumption model is as follows:

$$F = (A11 + A12*V + A13*V**2) * TT$$
$$+ (A21 + A22*V + A23*V**2) * D$$
$$+ (A31 + A32*V + A33*v**2) * S$$

where: F = estimated total arterial system fuel consumption gal/hr

TT = total travel, veh-mile/hr

D = total delay, veh-hr/hr

S = total stops, veh/hr

V = cruise (free) speed, mph

A_{ij} = model coefficients

where:

$$A_{ij} = \begin{matrix} 0.075283 & -1.5892\text{ E-3} & 1.50655\text{ E-5} \\ 0.73239 & 0.0 & 0.0 \\ 0.0 & 0.0 & 6.14112\text{ E-6} \end{matrix}$$

8. Stops per Vehicle

The recommended formula for estimating the stop rate, i.e. the average number of complete stops per vehicle, is:

$$h = 0.9* \frac{(1-g/c)}{(1-v/s)} + \frac{No}{vc}$$

where: v = arrival flow rate, veh/sec

C = cycle time, sec

g/C = effective green time/cycle time ratio

v/s = flow/saturation flow ratio

No = average overflow queue, veh/sec

where:

$$NO = EXP\left(\frac{1.33* \frac{(1-VC/gs)}{(VC/gS)} *(S*g)^{1/2}}{2 - VC/gs}\right)$$

$$X = \frac{V/S}{g/C}$$

$$= (v/s) * (c/g)$$

$$= vc/gs$$

The total number of (complete) stops per hour is calculated from H = qh. A convenient formula for calculating the number of stopped vehicles directly is:

$$H = \left(\frac{3240}{C} * \frac{V_r}{1-g/S} + NO\right)$$

OUTPUT REPORTS

PASSER II calculates almost all of the signal timing information needed for plan development and field implementation. Figure 6 demonstrates the schematic layout of the new PASSER II-84 output, which is similar to existing PASSER II-80 format with the added fine-tuned ''Best Solution,'' time-space diagram, and fuel consumption estimates.

(This page is blank)

NEW PASSER II-84 OUTPUT

EXISTING PASSER II-80 PROGRAM

1. ECHO PRINTOUT OF INPUT DATA DECK
2. CODING ERROR MESSAGES

↓

NEW PASSER II-84 ENHANCEMENTS

OFFSET FINE-TUNING ALGORITHM

OFFSET OPTIMIZATION ALGORITHM

NEW NCHRP DELAY EQUATION

FUEL CONSUMPTION MODEL

↓

3. BEST SOLUTION
4. NEW M.O.E. INCLUDE FUEL CONSUMPTION ESTIMATION
5. NEW TIME-SPACE DIAGRAM

Figure 3 . Schematic Layout of the PASSER II-84 Output.

The output first presents an echo printout of input information describing the arterial. Then, the input data is followed by intersection signal timing and evaluation results. The progression values include the optimum cycle length, widths of the progression bands in seconds for the A and B directions, the average band speeds, and two other descriptions of the quality of the progression solution. These two descriptions are bandwidth efficiency and attainability. Efficiency is the average fraction of the cycle available for progression. Attainability is the average fraction of the arterial minimum through green used for progression.

The initial result given for an intersection is the progression offset of the start of the first arterial phase with respect to the start of the first arterial phase of the first intersection in the ''A'' direction. Also shown is the arterial phase sequence selected by the program for this intersection. Signal timing information is then presented for the intersection beginning with the arterial phases followed by the cross street phases. Each phase is defined by the movement combination forming the phase. The green time shown for each phase includes the yellow clearance time. Measures of effectiveness are calculated for all movements, intersections, and total arterial systems, and can be used to make an objective evaluation of expected traffic conditions. Check the general balance of all measures and the level of service for each analysis group. The movements experiencing level of service D or greater are likely to experience both large delays and numerous failures of the stopped queues to clear during the green.

The table of ''progression efficiency versus cycle length'' can provide the relative indications of the progression quality within the ''allowable cycle length range'' and the workable cycle length range suggested by PASSER II-84.

ADDITIONAL FEATURES

PASSER was written to design progression along an arterial. It can also be used to analyze single intersections. To analyze single intersections the user would input a dummy link with zero speeds and distances. The remainder of the input data should be the same as the input for the intersection to be analyzed.

To analyze existing signal timing the user should input the known cycle length, with no variation allowed and zero traffic volumes on the intersection detail cards. The program will then use the minimium green as the actual green. This feature permits the user to examine other traffic engineering improvements, such as installing median refuge zones to reduce pedestrian clearances or alterations in parking policies.

The model can also be run in the ''isolated'' mode, which will design phase splits based on minimum delays, but no offset optimization is performed and all arrivals are assumed to be uniform. This feature, however, requires a constant cycle length on all signals.

APPLICATIONS AND LIMITATIONS

PASSER is a tool to assist in analyzing individual signalized intersection operations or to determine optimum time-space based progression along an arterial. The program determines optimal values of all traffic signal timing parameters: cycle length, splits, phase sequences and offsets. Several program runs may be needed before a final progression solution is calculated.

The major limitation is the narrow range of cycle lengths that can be tried in a given run, but, as stated earlier, this is easily overcome by multiple runs. The reason for this is that infeasible solutions may result for certain cycle lengths. The restricted range of cycle lengths affords the user the opportunity of carefully examining ''optimal'' solutions at several cycle lengths, thereby eliminating the infeasible solutions.

Finally, while phase sequencing is automatically ''optimized,'' selection for the best sequences depends on so many factors requiring engineering judgment. On the other hand, the program can assist the engineer by giving the optimal solution under a variety of sequence strategies input in several runs.

PASSER II-84 Version 2.3B (MICROCOMPUTER VERSION)

PROGRAM DESCRIPTION - The PASSER II-84 Microcomputer Environment System was developed by the Texas State Department of Highway and Public Transportation to facilitate use of the microcomputer version of PASSER II-84 and is designed for use on an IBM PC or compatible microcomputer. The complete system fits on the floppy disk leaving the other disk free for storage of problem data and 132 column output. Due to disk size limitations, the system is currently limited to analyzing no more than 10 signals.

PASSER II-84 microcomputer program is exactly the copy of the PASSER II-84 mainframe version. The following describes the preprocessor for this program. For more information the user is referred to the mainframe writeup of PASSER II-84 (20).

INPUT - The input menu provides eight choices.

1. INPUT NEW DATA: This option will be used whenever there is a new system to analyze. There are two restrictions which are built into the system which should be noted:

   a) All movement data must be input and will be output using NEMA phasing numbers as defined in the new Traffic Control Systems Handbook.

   b) If only one intersection is input, it will be analyzed as an isolated intersection. Otherwise, a progression solution will be found.

   The first section of input corresponds to Arterial Header Cards used in preparing normal PASSER II input decks. If only one intersection is to be input though, you will be asked for only one cycle length since PASSER II cannot evaluate different cycle lengths in an isolated situation. You will also not be asked to provide any information about the progression.

The second section of input asks for the name of each cross street as well as the volume, saturation flow and minimum green for each movement.

In the third section of input, the system asks for the appropriate phasing patterns to be analyzed. The system only allows the choice of patterns which are compatible with the movement volumes you have entered.

The fourth section of input will ask for the geometric input data for the links of the system. After finishing this, the input is complete.

The PASSER II environment system will assist in choosing a phasing pattern. It first finds out which movements have had volumes entered for them and then references them to one of the phasing patterns shown in Table 7. It will then give a list of those phasing patterns which the table indicates are applicable. It will also give the actual phase sequence which applies to this phase pattern.

There are three phasing pattern tables shown in Table 7. The first two are for normal operation on the arterial and the cross street. These tables should be sufficient for 90 percent of PASSER II problems. Occasionally, though, the full range of phasing patterns available will be needed. In that case, choose the special phasing table. This table provides the phasing patterns needed to achieve any possible phase sequence for the movements that have nonzero volumes.

Table 7 Normal Phasing Patterns *

Arterial

Movements 1	2	5	6	!	Phasing Patterns LT	THRU	#1	#5
X				!	-	-	-	-
	X			!	-	1	-	-
		X		!	-	-	-	-
			X	!	-	1	-	-
X	X			!	-	-	-	-
X		X		!	-	-	-	-
	X	X		!	-	-	-	1
X			X	!	-	-	1	-
	X		X	!	-	1	-	-
		X	X	!	-	-	-	-
X	X	X		!	2	-	-	2
X	X		X	!	-	2	2	-
X		X	X	!	2	-	2	-
	X	X	X	!	-	2	-	2
X	X	X	X	!	3	3	3	3

Cross Street

Movements 3	4	7	8	!	Phasing Patterns LT	THRU	#3	#7
X				!	-	-	1	-
	X			!	-	1	-	-
		X		!	-	-	-	1
			X	!	-	1	-	-
X	X			!	-	1	1	-
X		X		!	1	-	-	-
	X	X		!	-	-	-	1
X			X	!	-	-	1	-
	X		X	!	-	1	1	1
		X	X	!	-	2	-	2
X	X	X		!	2	-	-	2
X	X		X	!	-	2	2	-
X		X	X	!	2	-	2	-
	X	X	X	!	-	2	-	2
X	X	X	X	!	3	3	3	3

Arterial or Cross Street Special Phasing Patterns *

Movements 1 / 3	2 / 4	5 / 7	6 / 8	!	Phasing Patterns LT / LT	THRU / THRU	#1 / #3	#5 / #7
X				!	-	-	1	-
	X			!	-	1	-	-
		X		!	-	-	-	1
			X	!	-	1	-	-
X	X			!	-	1	1	-
X		X		!	1	-	1	1
	X	X		!	1	1	-	1
X			X	!	1	1	1	-
	X		X	!	-	1	1	1
		X	X	!	-	1	-	1
X	X	X		!	3	1	1	3
X	X		X	!	1	3	3	1
X		X	X	!	3	1	3	1
	X	X	X	!	1	3	1	3
X	X	X	X	!	3	3	3	3

* EXPLANATION OF SYMBOLS

X Movement has a volume greater than zero
\- Pattern not allowed
1 Pattern allowed only without overlap
2 Pattern allowed only with overlap
3 Pattern allowed with or without overlap

2. READ OLD DATA FROM DISK: This option will read an old data set from the B: disk drive. The data set currently in memory will be lost.

3. EDIT DATA: This option is used when there was a mistake in the input or you wish to alter some of the variables for another PASSER II-84 run. By choosing this feature, a new menu will appear which will give eight choices of data to edit.

   a. Edit general network data,

   b. Edit intersection movement data,

   c. Edit intersection phasing data,

   d. Change an intersection's phasing level,

   e. Edit network geometry data,

   f. Add an intersection,

   g. Delete an intersection, and

   h. Change a cross street name.

4. STORE DATA ON DISK: Data should only be stored on the B: disk drive. Putting any additional files on the PASSER II-84 disk will take up the disk space needed for the output routine. There are two points which must be remembered: a) NEVER SAVE A FILE ON DISK A:, and b) only load data files which were written by the PASSER II-84 Microcomputer Environment System. Disk space on drive A: is critical (unless it is a hard disk), presence of a data file will cause a disk full error when PASSER II is run and the system will crash. Do not put a write protect tab on the system disk because the system writes the output to this disk.

5. PRINT CURRENT INPUT DATA: It will print the entire data set.

6. RUN PASSER II-84: The system will check your data to find if there are any major errors which would cause PASSER II to malfunction. This check insures that:

   a) The phase patterns selected correspond to the movement volumes. If not, inappropriate patterns will be deleted.

   b) The minimum green times are equal for the simultaneous movements in phase patterns without overlap. The larger green will be used if unequal greens were specified.

   c) The sum of the critical phase minimum greens is less than the lower cycle length. (FATAL ERROR)

   d) Each arterial and cross street has a phase pattern selected for it. (FATAL ERROR)

Errors c and d are fatal and will stop PASSER II from running. Otherwise, the system will go to MS-DOS with a message to type in gopasser. This batch file will load the PASSER II program and run it. The run time of PASSER II without the use of an 8087 math coprocessor is around three minutes for a four intersection system. After PASSER II is through, the standard 132 column output is converted to an 80 column output and stored on the A: disk drive as output.

7. GO TO OUTPUT MENU: This menu will allow viewing or printing of PASSER II-84 output. Available menu items are as follows:

   a) View Input Echo: Shows page by page the input echo which is generated by PASSER II.

   b) View Best Solution: Shows intersection by intersection the best solution generated by PASSER II.

   c) View Arterial Summary: Shows in one page the system-wide measures of effectiveness. This page will not be available when either one intersection is being solved or when only one cycle length is being tested.

d) View Efficiency vs. Cycle Length Table: Shows bandwidth efficiency for each cycle length. This page will not be available when either one intersection is being solved or when only one cycle length is being tested.

e) View Time/Space Diagram: Shows the time/space diagram corresponding to the best solution. Only 80 columns will be output on the diagram, so if the indicated bands are cut off, use a larger horizontal scale to compress the bands. If the message Time/Space Diagram Too Long . . . appears, then use a larger vertical scale.

f) Print All Output: All output will be routed to the printer. Output is in 80 columns no special printer is required.

g) Go to Input Menu: Returns to the input menu at any time to compare the results of the PASSER II run to the input data then returns to the output menu by selecting the appropriate item on the input menu.

h) Quit: This item performs the same function as on the input menu, EXCEPT that it will not check to see if data have been edited.

8. QUIT: Opportunity to save the new data.

SPECIAL FEATURES AND SHORTCOMINGS - PASSER II-84 is the latest version of PASSER II program available on the market. A few of its special features are listed below:

1. It has a very easy and at the same time powerful preprocessor available.
2. It has a very powerful and easy to use editing feature.
3. Its capability to assist the user in choosing the phasing patterns makes this preprocessor very user-friendly and powerful at the same time.
4. The Output menu is very useful and sufficient, and user can view the results on the screen and printer.
5. The output is converted to 80 column, therefore no need for special printers anymore.
6. It has a powerful error detecting capability.
7. In order to be easy to use for technician, the program only utilizes the new NEMA movements.

Among the very few shortcomings, the following can be mentioned:

1. Program asks for the movements number and not the direction that makes it harder for the user.
2. It has the disk size limitation.
3. The 80 column printer makes it harder to read a Time/Space diagram.

SIGART AND TIMESPACE II

PROGRAM DESCRIPTION - An effective tool has been developed for use in coordinating traffic signals along an arterial (21). The tool is a computer program called SIGART (for SIGnal along an ARTerial). SIGART was developed by James H. Koll while at the University of California, Berkeley. This program produces all possible through bands (wider than some desired minimum) within a specified progressive speed range for all feasible cycle lengths.

TIMESPACE II accepts band information from card image input combined with SIGART generated results to produce printer type timespace diagrams.

SIGART and TIMESPACE II are two programs which, when operated together, can produce effective timing plans for typical arterial type traffic signal systems. SIGART is a stand along program which can be operated separately if graphical output is not desired. TIMESPACE II, however, must be operated with an input file created by SIGART. Timespace plots are designed for use on 132 column printers scaled for 14 inch wide paper.

Program SIGART is well suited to situations which can be reasonably described as two phase traffic signal operations. Essentially, SIGART considers the coordinated phase as a green along the arterial with all other phases included in the red against the arterial. This assumption means that arterial left turns, side street left turns and side street through greens are treated as red against the arterial. This assumption is reasonably easy to handle in the preliminary design phase and may require special consideration only in final field key setting development.

SIGART permits the user to deal with both a desired speed and a range of acceptable speeds. This is one of its strengths permitting the designer to test a broad range of values easily and quickly. The resulting solutions are through bands, of equal size in each direction, with the same speed in both directions. Band directional split adjustments are accommodated by specifying the percent of directional traffic in TIMESPACE II.

The offset solutions produced by SIGART are based on time and distance with no direct traffic or capacity considerations. The solutions which are produced rely upon the traffic engineer to determine which of the solutions is most suitable for field installation. This procedure eliminates the drudgery associated with candidate offset plan and timespace diagram development and retains the engineer in the design cycle as the decision making authority.

The program operates in the following manner. It reads in the input data and outputs a data summary. This step permits user verification to ensure that the data are properly input to the program. If one of the available options has been exercised, the program then computes the maximum band possible for the desired speed for each of the input cycle lengths. For each cycle, the program outputs the maximum band width, the individual offsets and the critical signals that define the through band.

The program then computes every possible through band within the specified speed range for each requested cycle length. As each possible through band is found, it is tested against the minimum acceptable band width. This minimum may be set to the value input to the program, the value found for the maximum band at the desired speed, or the larger of these two values, depending upon which program option has been selected. Bands less than this minimum are rejected. As each feasible band is found, the program outputs the appropriate data, including the progression speed, the through band width, the various offsets, and the critical signals that delimit the band.

The program then prints a summary of the feasible bands found for each of the requested cycle lengths. For each band, this summary contains the speed of progression, the through band width, and the relative band efficiency (i.e., the band width expressed in terms of percent of cycle). This summary enables the traffic engineer to quickly determine the best through bands available for that specific arterial.

INPUT - Input to the SIGART consists of data readily available from the traffic engineer's files. These data include the following:

- the number of signalized intersections along the arterial
- the number of cycles to be tested
- three speed values in mph or fps (the desired speed, the maximum speed to be considered, and the minmum speed that would be allowed
- the minimum bandwidth
- the cycle lengths in seconds
- the percent of cycle that is red at each individual intersection
- heading that appears on each page of output
- various items that specify program options
- the names of intersecting streets at the signalized intersections, and
- a title for timespace diagram.

The input to TIMESPACE II include a directional split band adjustment capability. ,

OUTPUT - A typical SIGART results are as follow: The first group of information summarizes data input to the program. Where the calculated split is based upon the pedestrian crossing time a ''*'' is printed adjacent to the split value.

The maximum bandwidths, offsets and critical signals for the desired speed and cycle length are printed next. The critical signals are those signals that would be affected if the band widths are widened.

The last block of information listed is the Summary of Feasible Bands. For each cycle length the desired speed and band width along with all feasible bands over the test band minimum are listed.

The typical TIMESPACE II output include timespace diagrams and tables of time differences between greens for signals along the SIGART route.

A printer is used to produce timespace diagrams. The user, after examining SIGART output, selects bands of interest. Band identifier and directional split information are entered into data file. TIMESPACE II is then run. The output contains band descriptive information including speed of progression, band width, critical signals, cycle length, band number, percent directional split information and a tabular summary of differences between greens.

PROGRAM HIGHLIGHTS AND SHORTCOMINGS - SIGART accepts arterial descriptive data and performance range limits and calculates all possible progression offset patterns within the specified limits. This program produces offset patterns for two way equal and opposite progression schemes. The input to this program is a card image file produced from manually coded forms prepared by the engineer. The output from this program consists of an input data summary, offset plan listings, band descriptions, statistical data and optionally a data file for preparation of timespace plots.

TIMESPACE II requires results generated by the program SIGART. It accepts band information from card image input combined with the SIGART generated results to produce printer's type timespace diagrams. The input includes a band adjustment capability to adjust, within limits, the normally produced 50/50 directional offsets. The output includes timespace diagrams and tables of time differences between greens for signals along the SIGART route. Multiple timespace plots may be produced with each program run.

SIGART has an advantage of multiple runs, and provides two-way progression. It can handle up to 10 different cycles per run. The major disadvantages are: 1) the program needs an external editor to create the data file outside the program, and 2) it does not take lead or lag green into consideration in calculating band width.

SPAN

PROGRAM DESCRIPTION

The Signal Progression ANalysis program is part of the McTRANS package and it is programmed in integer basic for Apple II computers (22). It performs a complete analysis of an arterial signal system using a simple optimization of offsets. The program utilizes simple travel time projections to display the position of signal phases and time location diagrams. Optimization of signal offsets is based on projecting the center of the progression band in both directions allowing for directional weighting and through band weighting.

INPUT

The program requires artery data entry such as speeds, distances, phasings, and system cycle length. The input stream for span has to be in the form of the SOAP/M data input. User has to run SOAP/M for each intersection separately and combine the results, in order to run the program.

OUTPUT

Three items of output are produced by the program:

1) Display of a Time-Location Diagram on the screen for the arterial signal system
2) A Time-Space Diagram showing the characteristics of signal progression
3) A simple optimization of offsets for progression.

SPECIAL FEATURES AND SHORTCOMINGS

The SPAN program permits the user to develop arterial progression plans by means of optimizing signal offsets. It provides for manual adjustments of offsets, and the summary report provides detailed information such as band width efficiency per direction and for combined directions.

On the other hand, the program is designed to read data generated only by SOAP/M program. Furthermore, it is developed only for Apple II computers.

TIMDIS II

PROGRAM DESCRIPTION – TIMDIS is a computer program which draws detailed TIMe-DIStance diagrams and an interactive time-location diagram for use in the design and presentation of traffic signal coordination plans (23). It also produces a formatted digital record of signal timing at each intersection, can automatically adjust offsets for oneway progression, and enables easy modification of variables such as phase times, offsets, phase sequence, cycle length and speeds.

The program was conceived and initially developed in 1981-82 by consulting traffic engineers engaged in the design of signal coordination plans for various traffic engineering agencies. The variety of road conditions, signal equipment, design philosophies and presentation requirements encountered in their work necessitated the development of an evaluation tool. The work face environment in which it was developed ensured that it was also simple and efficient to use.

The primary objectives in writing the program were:

- To increase the productivity of staff engaged in the design of signal coordination plans.
- To improve the consistency, accuracy, and appearance of time distance diagrams, and signal timing data taken from time distance diagrams.
- To relieve staff of the tedious and repetitious task of manually drawing and redrawing time distance diagrams.

The program was designed to interface with TRANSYT so that time distance diagrams of optimized settings could be produced quickly and simply. This assists in the checking, interpretation, and evaluation of TRANSYT output and the selection of alternatives for input to TRANSYT. Compatibility with TRANSYT in no way reduces its capabilities or usefulness as a stand alone program. Perhaps its greatest value is in the manual design of signal coordination plans, which conventionally involves the manipulation of time distance diagrams to achieve the desired result.

INPUT - The initial input data consist of the following (* indicates the data is optional):

* * Title
* * Subtitle
* * Date
* Viewing direction
* Cycle length
* * Time axis length
* * Default speed
* * Block length definition (centerlines or stoplines)
* * Default intergreen (change interval) length
* * Distance and speed units (metric or English)
* * Time units (seconds or percent)
* * Distance between axes and adjacent intersections
* * Phase symbol library file name
* Intersecting street name or other identification
* Number of phases
* * Single or double cycle
* Phase times
* Phase symbol code
* * Phasing and timing directional independence
* * Sync phase
* * Sync point (start of green or yellow)

Progression window start and end phases, and speed

Block lengths

* Intersection width

Interactive input during run time can be as simple as answering one question inputting a data file name and selecting an option from two menus. Alternatively, the program can be kept running endlessly by selecting various features, drawing a diagram, making data modifications, redrawing the diagram, calling or creating new data files, etc.

Extensive use is made of menus, explanatory prompts and default values to make the program easy to use. Input data is checked comprehensively for type, range and logic. Program termination cannot normally be used by incorrect data entry. Data errors are explained and an opportunity provided to correct them interactively. A help screen is provided for time location diagram.

Input data can be modified via either the 'Modify Data' or 'Time-Location Diagram' options.

<u>Modify data Option</u> - There are five options for modifying the input data in this mode:

- Change offsets or baseline
- Change pulse times
- Change start and end times
- Automatic one-way progression
- Change title or subtitle

<u>Time Location Diagram Option</u> - Choosing this option causes a time location diagram to be displayed on the computer or terminal screen. Rightbound

phase times are shown on the left side of the screen where the direction of travel is down the screen. Leftbound phase times are shown on the right side of the screen where the direction of travel is up the screen. Each line represents one intersection. The intersection's consecutive number in the rightbound direction and the rightbound identification are displayed at the center of the line.

Progression for through traffic is achieved when the characters representing the progression movement at adjacent intersections are aligned. Progression for traffic turning left onto the route is achieved when the entry movement characters are aligned with progression movement characters at the adjacent downstream intersection. Progression for traffic turning left from the route is achieved when the exit movement characters are aligned with progression movement characters at the adjacent upstream intersection. Other characters represent non-progression movements such as traffic in the opposite direction, or side street through traffic.

There are two types of action that can be taken in the time location mode. One involves just one intersection at a time - change offsets, change phasing, change windows, change splits, or check the data. To invoke these actions, the intersection number must be entered first, followed by the action required. The second type involves all intersections--move rightbound, move leftbound, change time scale, change cycle length, change speed. To invoke these actions, the action must be specified first, followed by a number indicating the magnitude of the change required. The periods shown in the menu prompt indicate a number to be supplied by the user when making a selection.

OUTPUT – Time-distance diagrams can be displayed on the terminal, or drawn on a printer.

There are several alternative diagram styles offered in the STYLE menu. These are as follow:

''One second/percent of travel'' option gives a distance axis scale such that a diagram line corresponds to the distance traveled in one second or percent of travel time in the rightbound direction.

''User supplied scale'' option provides a distance axis scale equal to a fixed number of meters or feet per line.

The ''Show bands,'' ''Show marks only'' and ''No bands or marks'' options refer to the amount of detail to be shown for the purpose of indicating bands. The ''Show band'' option produces a shaded band for each window. The ''Show marks only'' option causes small marks to be printed adjacent to each intersection to indicate the width and location of the band at its downstream end. The ''No bands or marks'' option shows nothing to indicate band width or slope.

Data can be displayed on the terminal or printed on a printer. This allows the input data to be checked and a hard copy made for other uses, especially the programming of controllers, and as a permanent record. There are four optional formats for presenting the data:

- Unformatted copy of the entire data file
- Formatted copy of the entire data file
- Green start and end times only
- Phase times and offsets

PROGRAM HIGHLIGHTS AND SHORTCOMINGS – Details of the program's capabilities include the following:

1. Phase times (in percent or seconds) input interactively either as split and change intervals, or as start and end of green.
2. Up to 7 traditional single ring phases or up to 14 dual ring (NEMA) phases can be specified for each intersection.
3. Distance and speed units can be either metric or English.
4. Different block lengths in each direction if required, measured between either centerlines or stoplines.
5. Any phase can be specified as the sync phase, and offsets can be referenced to either the start of green or the start of yellow.
6. Independent phase descriptions in each direction if required such as at midblock pedestrian crossings on divided roads or on one way street.
7. One second (or percent) resolution in phase timings, one meter (or foot) resolution in block length, and one km/h (or mph) in speeds.
8. Up to three progression windows and bands in each direction, each with a different progression speed if required, such as for repeated phases, separate bus progression, speed sensitivity analysis, turning movements, etc.
9. Each intersection's width specified and shown accordingly on the diagram if required.
10. Offsets and baseline position can be changed by interactive input of the increment only.
11. Offsets giving reasonable oneway progression can be automatically generated as a starting point for manual designs.
12. A time location diagram can be displayed on the computer screen and used to instantly see the effects of changes in offsets, phase splits, phase sequence, cycle length, or speeds.

13. The time location diagram allows the user to observe progression opportunities for turning traffic as well as through traffic.

14. Any number of phase symbol libraries can be specified, each containing 25 phase symbols.

15. Time distance diagram time axis length limited only by capabilities of printer or display device. Distance axis unlimited in length.

16. The ''Distance'' axis can represent travel time if required for constant band slope.

17. Progression ''bands'' can be drawn automatically, guide marks only provided, or nothing at all.

18. Phase symbols can be drawn independently on either side of intersection, or not at all.

19. Phase green intervals, change intervals and progression windows can be shown on the time distance diagram in a variety of ways specified by the user.

20. Output can be directed to a printer, or the computer screen, and the diagram is automatically adjusted accordingly.

21. All data can be stored and saved on files named by the user.

22. There is a variety of summary data formats for documentation of controller timings and phasing.

TIMDIS is a user friendly and versatile tool for use in the design and presentation of signal coordination plans. Time distance diagrams can be produced as quickly as the printer can print them. Input data can be modified interactively with the aid of a time location diagram and the time distance diagram redrawn. The latest data (and former if required) are recorded in digital form both on computer file and as hard copy output.

Flexibility in input and output parameters enable a wide range of user requirements to be met.

TIMDIS has two major disadvantages:

1. It only can handle one-way progression automatically, and
2. It does not calculate band width or band efficiency.

MAXBAND

MAXBAND is based on the traffic signal model developed by J. D. C. Little of the Massachusetts Institute of Technology (24).

MAXBAND can determine optimal signal settings for arteries or loops as described below:

Arteries - An artery is a series of 2 to 20 signalized intersections along a street, together with connecting roadways. Intersections may be 4-way or T-shaped, or there may be no cross street at all. MAXBAND analyzes each intersection as though there are at most four phases: a main street phase, a cross street phase, and up to two 'all red' phases during which both the main and cross streets have a red light. The main and cross street phases are in turn divided into green and left phases in each direction.

The two directions of an artery are referred to as outbound and inbound. An artery may be one-way for all or part of its length.

Loops - In MAXBAND, a network of three arteries in which each artery intersects the other two at distinct intersections is called a loop. The total number of intersections in the loop must be in the range 3 - 17, with each artery having 2 to 16 intersections. In general, each of the three arteries is analyzed in the same way as the one artery of the single artery case.

MODEL DESCRIPTION

MAXBAND, or Maximal Bandwidth Traffic Signal Setting Optimization Program, is a system which finds traffic signal settings on arteries, or certain networks of three arteries, which result in maximal bandwidths. Problems are formulated as mixed integer programs.

MAXBAND is written in FORTRAN using structured programming techniques. At MIT, MAXBAND was run on an IBM 370/168 using the Conversational Monitor System (CMS) operating system.

The MAXBAND system consists of five modules: an overall control module (MAXBAND): and four modules which handle specific subtasks and will be described in the next section. The latter four modules execute sequentially.

The MAXBAND MODULE - It contains the main program and several subroutines. It provides overall control of the system.

The INPUT Modules - The INPUT module reads the input cards on FORTRAN file 4, and places their card images in a temporary disk file on FORTRAN file 5, checks them for errors and converts the information they contain into the internal form used by MAXBAND. This conversion includes the calculation of splits from volume and capacity information, when this is to be done, and the modification of splits to satisfy minimum split times.

The INPUT module produces the first portion of the MAXBAND output on FORTRAN file 6. It first echoes the input cards. Interspersed among and following the echoes are any warning and error messages produced when errors are found in the input cards. If no serious errors are detected, the module then prints an input data summary. This contains the final version of the input information which will be passed to the Matrix Generator (MATGEN) module.

The user can specify that MAXBAND halt after the INPUT module has been executed. The system will produce only the input card echoes and the input data summary.

The Matrix Generator (MATGEN) Module - MAXBAND formulates problems as mixed integer mathematical programs. These are solved by the MPCODE module, which contains the MPCODE Mathematical Programming System. After the input data has been translated to the internal form by the INPUT module, the Matrix Generator (MATGEN) module builds this information into a mixed integer formulation, which is used as input to the MPCODE module. The MATGEN module places this formulation on FORTRAN file 11.

The MPCODE Module - MAXBAND uses the MPCODE mathematical programming system, written by A. Land and S. Powell (1973), to solve the problem formulations produced by the MATGEN module. The mixed integer portions of the MPCODE system have been placed in the MPCODE module.

The module reads the problem formulation produced by the MATGEN module on FORTRAN file 11, solves the problem, and passes the results back to the MAXBAND control module.

The output capabilities of the original MPCODE system have been retained. The MPCODE module writes the MPCODE output described by Land and Powell on FORTRAN file 12. This file may be dummied if not required.

The MPCODE module also incorporates the restart capabilities of the original MPCODE system. If the upper limit on the total number of iterations is about to be exceeded, the module will stop trying to solve the problem and will write a restart file to FORTRAN file 7. The restart file contains the state of the module just before the file is written. If the user later chooses to restart the problem, the MPCODE module will read the latest restart file on FORTRAN file 8. This prevents the system from having to repeat work already done.

The OUTPUT Module - After the MPCODE module has completed, MAXBAND calls the OUTPUT module to report any solution found.

The OUTPUT module first reports whether the MPCODE module has found a solution and whether any error conditions were encountered. It then gives statistics describing the operation of the MPCODE system. Finally, if MPCODE was able to find a problem solution, the OUTPUT module prints the signal settings and other values given by this solution.

Table 8 gives the minimum amounts of core required by each of the five modules.

MAXBAND, INPUT, MATGEN and OUTPUT were compiled using the FORTRAN IV G compiler; MPCODE was compiled by the FORTRAN H Extended compiler, using the optimize (2) option.

The table does not include any core which may be required by the operating system.

Since the INPUT, MATGEN, MPCODE and OUTPUT modules execute sequentially, they may overlap each other. The MAXBAND module must be in memory at all times.

INPUT REQUIREMENTS

Arteries

MAXBAND requires the following information in single arterial problems:

1. Number of intersections

Table 8 Minimum Core Required by the Modules

MODULE	MINIMUM CORE (K)
MAXBAND	8
INPUT	114
MATGEN	84
MPCODE	245
OUTPUT	89
Total	540

2. Units (English or metric)

3. Map direction from which intersection is entered

4. Cycle time limits

5. Artery-wide design speeds and tolerances on links, and allowed changes in speed between links

6. Either main street splits, or volume and capacity information on intersection approaches from which main street splits will be calculated

7. Left turn phasing patterns to be considered, if there are left turn splits

8. Link lengths

The user may allow some of this information to default.

The following information is optional in single artery problems:

1. Design speeds and tolerances for specific links

2. Minimum split durations

3. Bandwidth period advance times to allow for queues

4. Values to control the relative importance of the outbound and inbound bandwidths in the solution

5. The artery name and cross street names and node numbers at the intersections

Loops

Loop input consists of general information about the loop as a whole, plus specific information about each of the three arteries. The input for each artery is the same as in the previous section, except that cycle time limits are excluded, being given for the entire loop instead of each artery. The following information is required in the loop case:

1. Number of intersections

2. Cycle time limits

3. Loop geometry, i.e., where the three arteries intersect with each other

4. Single artery information (except for cycle time limits) for each of the three arteries

In addition, the following information is optional:

1. Values to control the relative importance in the solution of the three outbound bandwidths

2. Name of the loop

The input data cards are divided into the six function groups shown in Table 9 . All MAXBAND cards are 80 characters long. The name of the card begins in column 1 and must appear exactly as shown in Table 9. Numbers must be entered right-justified into all numeric fields.

In the artery case, the deck begins with the START card. Optionally, this may be followed by the MPCODE card. Next are the cards which establish the problem as an artery problem and give general artery information (group 3). Following these are sets of cards giving information about a specific signal and its approaches (group 5). The deck ends with the END card.

The loop deck also begins with the START card, followed optionally by the MPCODE card. Next come cards which indicate that the problem deals with a loop and given general information about the loop (group 4). Following these are cards introducing the first artery (group 3), which are followed by sets of cards with information about each intersection in the artery (group 5), just as in the single artery case. The group 3 and group 5 cards are then repeated for the other two arteries. Note that the three artery sections begin with SETUP1, SETUP2 and SETUP3 cards, respectively, and not

Table 9 The Six Types of MAXBAND Input Cards.

1.	Starting and ending the job, and controlling output.	START END
2.	Controlling operation of the maximization routine (MPCODE).	MPCODE
3.	Starting an artery section and giving general artery information.	SETUP SETUP1 SETUP2 SETUP3 ARTERY ART2
4.	Starting a loop problem and giving general loop information.	SETUPL LOOP1 LOOP2
5.	Giving information about a specific signal and its approaches.	MAP SPECIFY VOLUME CAPACITY MINGREEN LEFTPAT LENGTH SPEED QUEUE
6.	Other cards.	COMM

SETUP cards, as in the single artery case. The deck is completed by an END card.

OPERATIONAL SUMMARY

MAXBAND uses the maximization routine, MPCODE, to optimize its formulations of the artery and loop problems. In finding the optimal solution, MPCODE solves a large number of subordinate linear programs. Each of these, in turn, requires a certain number of steps, or iterations, for solution. In addition, MPCODE occasionally performs a procedure called a ''reinversion,'' which allows it to maintain the accuracy of its representation of the MAXBAND formulation. In the 'MPCODE Statistics' portion of the output, MAXBAND indicates:

1. How many iterations the entire maximization routine requires.
2. The largest number of iterations required by a single linear program.
3. How many reinversions were performed during the entire maximization routine; and
4. The largest number of reinversions performed during a single linear program.

MAXBAND allows the user to specify upper limits on the size of each of these four quantities that can be attained during the performance of the maximization routine. This is done by entering values in the appropriate fields of the MPCODE card.

If the MPCODE card is not used, MAXBAND uses the following defaults for parameters for which no value has been entered:

MAXRBB - 10000

MAXIBB	-	100
MAXITR	-	1000
MAXINV	-	100

If one of MAXIBB, MAXITR, or MAXINV is exceeded, the maximization routine halts, a message is printed in the MAXBAND output, and the program halts.

COMPUTATIONAL ALGORITHMS

The optimization algorithms used in MAXBAND utilize two basic mixed integer linear programs. The first is a single optimization algorithm that handles two-phase signals and the second is a more generalized formulation which takes into account left turn phases. The general formulation of the first algorithm is as follows:

Find b, $\bar{b}$, W_i, $\bar{W}_i$, and m_i

$$\max b$$

subject to $\bar{b} = b$

$$w_i + b \leq 1 - r_i \qquad i = 1, 2, \ldots, n$$

$$\bar{W}_i + \bar{b} \leq 1 - \bar{r}_i$$

$$(W_i + \bar{W}_i) - (W_{i+1} \quad \bar{W}_{i+1}) + (t_i + \bar{t}_i) + \Delta_i - \Delta_{i+1}$$

$$= - (1/2)\ (r_i + \bar{r}_i) + (1/2)\ (r_{i+1} + \bar{r}_{i+1}) +$$

$$(\bar{\tau}_i + \tau_{i+1}) + m_i, \qquad i=1,\ldots,n-1$$

$$m_i = \text{integer}$$

$$b,\ \bar{b},\ w_i,\ \bar{w}_i \geq 0 \qquad i=1,\ldots,n$$

where:

$b(\bar{b})$ = Outbound (inbound) bandwidth (in cycles)

$r_i(\bar{r}_i)$ = outbound (inbound) red time at S_i (in cycles)

S_i = ith signal $i=1,\ldots,n$

$W_i(\bar{W}_i)$ = time from right (left) side of red at S_i to left (right) edge of outbound (inbound) green band (in cycles)

$t_i(\bar{t}_i)$ = travel time from S_i to S_{i+1} outbound (S_{i+1} to S_i inbound) (in cycles)

Δ_i = time from center of $\bar{r}_i$ to nearest center of r_i. Positive if center of r_i is to right of center of $\bar{r}_i$ (in cycles)

$\tau_i(\bar{\tau}_i)$ = queue clearance time, an advance of the outbound (inbound) bandwidth upon leaving S_i (leaving S_{i+1}) (in cycles)

The details of each term can be seen in figure 4 . The formulation described above has 3n constraints, 2n+2 continuous variables and n-1 unrestricted integer variables.

OUTPUT REPORTS

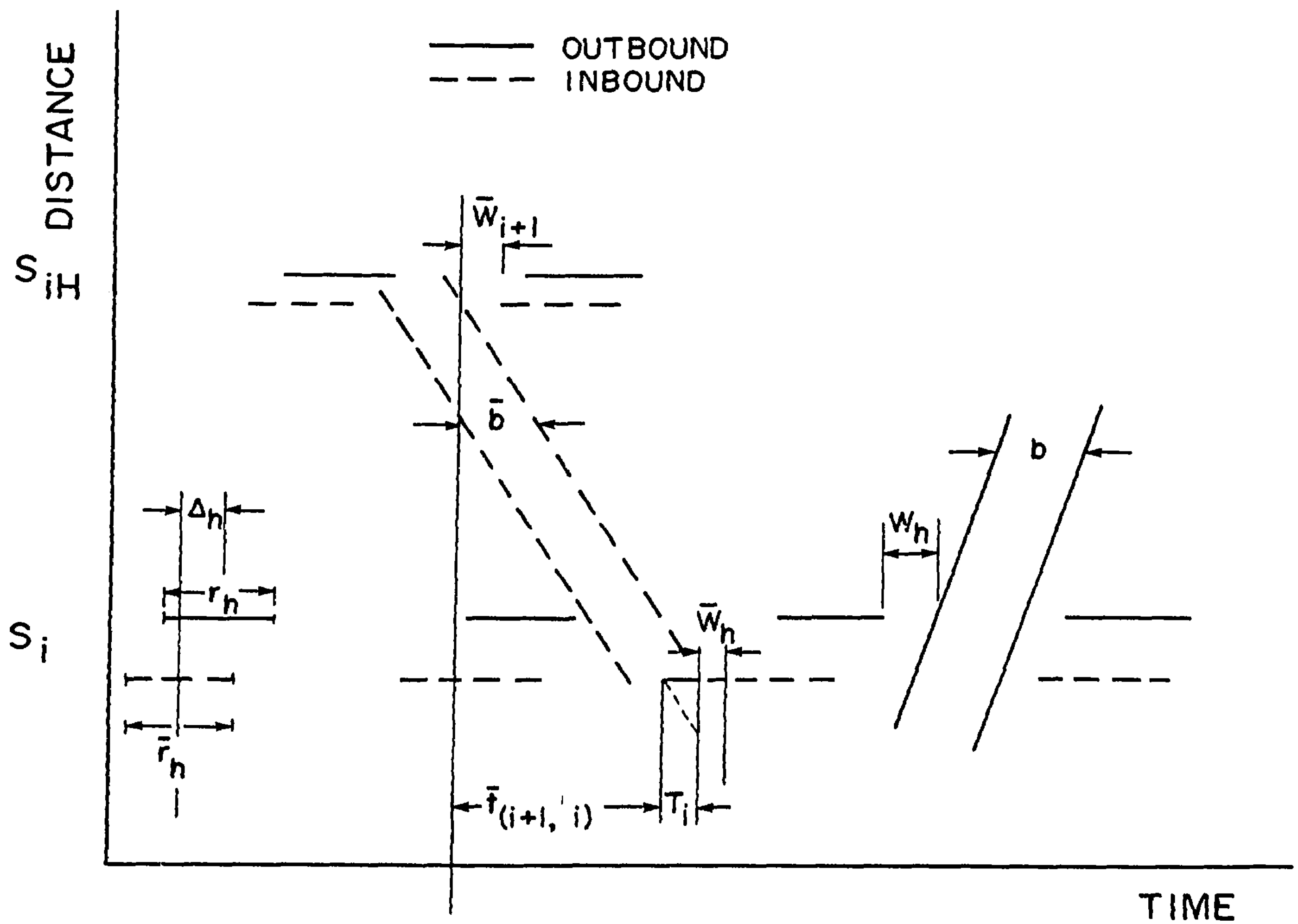

Figure 4 Time-Space Diagram Terminologies of MAXBAND

MAXBAND produces three output reports. The user has the option to select which reports are to be printed by the program. The following describes these reports:

<u>MAXBAND Input Cards</u>:

This output echoes the input card exactly as it was read by the program.

<u>INPUT Data Summary</u>:

This report contains the following information:

(a) MPCODE values used by the maximization routine

(b) For an artery:

(1) General information: such as the name of the artery, the number of signals, limits on cycle length, units, target bandwidth ratio, and bandwidth weights

(2) Artery-wide values such as design speed, tolerances, and limits on changes between links

(3) Intersection values including splits with an indication of their origin, queue clearance times, minimum greens, and the permitted patterns for left turns

(4) Link values as actually used including length, design speed and speed tolerances

(5) Volumes and capacities on all approaches, when provided

(c) For a loop:

(1) General loop information including upper and lower limits on cycle time and where the arteries meet

(2) The same artery information described above but for each of the three arteries

The solution report presents the following data:

(a) An indicator whether the problem has been solved successfully.

(b) MPCODE statistics describing the number of iterations, etc., used by the Land and Powell algorithm to solve the problem. ''Number of solutions'' is the number of integer solutions (including the optimal integer solution) found for the problem.

(c) For an artery:

(1) General information including name of artery, number of signals, and type of units

(2) Cycle time and bandwidths

(3) Left patterns selected as optimal

(4) Duration and offsets of splits in both fractions of a cycle and seconds

(5) Traversal times and speeds on links

(d) For a loop:

(1) Loop information including chosen cycle time, bandwidths, and objective function

(2) Same information as for a single artery for each of the three arteries

(3) Repeat of duration and offsets of splits for signals at artery meetings

APPLICATIONS AND LIMITATIONS

As previously stated, the MAXBAND program is designed for use only on arterial streets and three legged networks. Due to the nature of the program there are minimal data requirements. The program attempts to provide the best possible bandwidth for the arterial being studied. This results in ignoring the sidestreet traffic crossing the arterial. This may not be the most desirable situation for some localities.

TW-BANDWIDTH (MICROCOMPUTER VERSION OF MAXBAND)

PROGRAM DESCRIPTION - TW-Bandwidth is a microcomputer version of FHWA's MAXBAND program. TW-Bandwidth maximizes main street through green band by optimally selecting cycle length, offsets, and most importantly, left-turn phase sequence for an arterial of up to 20 intersections or a triangular loop. TW-Bandwidth is compatible with MAXBAND's data input format, and it produces a printed time-space diagram (24).

TW-Bandwidth is a very large and complex program. Its size necessitates that the program be split into two parts, BAND1 and BAND2, in order to fit on floppy diskettes. The complexity of the program and its options also requires the usage of a number of temporary data files. The following paragraphs describe the operational and file structure of TW-Bandwidth.

BAND1 is a data preprocessor for BAND2. BAND1 prompts you for the input data filename and the output filename (Default=PRN). After reading the input file, BAND1 writes out two temporary data files. BNDWIDTH.TMP contains program control information to be passed to BAND2. MPCODE.IN contains the problem formulation for the MPCODE mathematical programming system written by A. Land and S. Powell [1973] contained within BAND2. If using a hard disk and want these temporary files to be written to and read from a hard disk, then it is necessary to change the single character ''A'' in file BANDWIDTH.DRV to the hard disk drive specification—normally ''C.''

BAND2 should be on the default drive after executing BAND1. If using a two floppy drive system, this would be the same drive where BAND1 executed from, this is normally drive B. BAND2 contains the MPCODE mathematical programming system which performs the actual optimization and also contains the output module. BAND2 first prompts for the output filename

(Default=PRN). The output capabilities of the original MPCODE system have also been retained. BAND2 asks (Y/N) if the MPCODE output is to be written. If the reply is yes (Y) to the prompt, the MPCODE output MPCODE.OUT will be written to the working data drive as specified in file BNDWIDTH.DRV. BAND2 also incorporates the restart capabilities of the original MPCODE system. This is a very important feature for the use of this program on microcomputers. If the upper limit on the total number of iterations is about to be exceeded, BAND2 will stop trying to solve the problem and will write a restart file RESTART.BND to the working data disk as again specified in file BNDWIDTH.DRV. To restart the problem simply rerun BAND1 and BAND2, and BAND2 will automatically read the restart file. There is no limit to the number of restarts that can be performed.

TW-Bandwidth's Hardware Requirements - TW-Bandwidth requires PC/MS-DOS 2.xx, 2 5-1/4 inch 360K DSDD disk drives or one 360K DSDD disk drive and a hard disk, 348K of RAM exclusive of the operating system, an 8087 math co-processor, and 132 column print capability.

TW-Bandwidth Execution Hints - For moderate to large problems, the following points should be considered before running TW-Bandwidth.

1. Cycle length is treated as a continuous variable in the optimization procedure. A wide range between minimum and maximum cycle length can result in long execution times. Consider making separate runs at fixed cycle lengths of interest and within the capabilities of the traffic controllers.
2. Consider not specifying left turn volumes and capacities if the left turn volume can be serviced without a left-turn phase. This

elimination of unnecessary phases can significantly reduce execution times.

3. The math co-processor computes real numbers with a higher precision than double precision on an IBM mainframe which thus results in a greater number of iterations (25 percent) and feasible solutions found (though not significantly better). Consider using a TOLRN card with tolerances as much as 1,000 times greater so as to reduce the number of iterations and thus execution time. Note, however, that these larger tolerances could result in a runtime error. If so, use tolerances only 100 times greater.

Extra Protection Against Power Interruptions - When executing TW-Bandwidth with a large problem, the use of TW-Bandwidth's restart capabilities is strongly recommended so as to protect against electrical power interruptions which could result in having to rerun TW-Bandwidth from the beginning. BATCH file TWBAND.BAT has been designed to provide a high level of protection for the restart file. If a power interruption should result in the destruction of RESTART.BND, file RESTART.BAK can be renamed to RESTART.BND and the BATCH file restarted. The BATCH file has been designed to operate from a hard disk where all of the TW-Bandwidth files should reside. To execute this BATCH file, enter TWBAND ''input file'' where ''input file is the name of the file containing input data for TW-Bandwidth. Review the BATCH file for an understanding of how it works.

MAXBAND User's Manual - Because of TW-Bandwidth's 100 percent data and functional compatibility with the Federal Highway Administration's MAXBAND

(November 1984 March 1985) program, the MAXBAND User's Manual may be used as TW-Bandwidth's Reference Manual.

EDBAND - Interactive Full Screen Input Editor for Arterials - Edband is a full-screen editor for preparing an input data file for Transware's TW-Bandwidth program or FHWA's MAXBAND program for mainframes. A menu driven series of screens that allow full cursor control to navigate about the screen quickens data entry with help files available for almost every data field.

Compress Printing Batch Utility Files - Included on the floppy diskette are two batch files and their corresponding data files for easily controlling compressed (17 characters per inch) and normal (10 characters per inch) printing. Type COMPRESS to transmit the corresponding control code for compressed printing and NORMAL to return to normal 10 characters per inch printing.

FORTRAN Output File Print Utility - FORTRAN output files on disk with carriage control characters can be printed later without loss of carriage control by using PRINTOUT.EXE.

PASSER III

PASSER III was developed to assist the traffic engineer in determining the optimal traffic signal timings for signalized diamond interchanges. The program is applicable to isolated interchanges as well as a series of interchanges through which progression is desired along one-way frontage roads. PASSER III, like PASSER II(80), was developed at the Texas Transportation Institute for use in the Dallas Corridor Project which was sponsored by the Federal Highway Administration (FHWA) and the Texas State Department of Highways and Public Transportation (SDHPT) in cooperation with the City of Dallas. PASSER III was adapted and improved upon in HPR Project 178 which was also sponsored by the Texas SDHPT and FHWA. The Texas SDHPT maintains the model and is used extensively by its staff (25).

MODEL DESCRIPTION

In urban areas, most diamond interchanges are signalized at the ramp terminals, where the ramps intersect the cross street. Diamond interchanges are normally characterized by their close spacing of the ramp terminals and the resulting small storage areas between the signals. In the early 1960's the Texas Transportation Institute developed a novel signalization strategy for diamond interchanges which took into account the fact that the throughput (or capacity) of the system could be increased by allowing several potentially conflicting movements at the separate intersections to occur simultaneously for a short time. This period was termed the ''overlap phase'' for obvious reasons, and the underlying concept has become a standard in the profession.

PASSER III, which is an acronym for Progressive Analysis and Signal System Evaluation Routine, Model III (Diamond Interchange), determines the optimal phase patterns, splits and internal offsets at single interchanges (for given cycle lengths) and additionally the optimal system cycle length and progression offsets for the frontage road progression. The physical system considered is the signalized diamond interchange, with or without through frontage roads or a series of interconnected interchanges with progression on the parallel (frontage) road.

The computer program is written in FORTRAN IV and consists of about 3100 statements. It is estimated that machines with core storage of 168K can accommodate most problems.

INPUT REQUIREMENTS

The input data required for this program are similar to those needed by the PASSER 80 model. The program uses data that are normally collected and used for signal analysis at diamond interchanges, with some special requirements. The current program can handle up to 15 interchanges in a single run.

Three types of input cards are used for PASSER III:

1. Freeway Header - This card identifies the freeway and defines some general parameters and options.
2. Interchange Header - This card provides signalization and geometric information for each signalized interchange in the data set. Link data for the frontage road must be provided if a frontage road progression is desired. One card is required for each interchange.
3. Interchange Detail Cards - Three cards are required for each interchange. Card one contains traffic volumes, card two contains the effective number of lanes for each movement and card three presents the minimum green time in seconds for each signal phase.

Most of the input are self-explanatory, but there are a few peculiarities which should be noted. PASSER III has two primary functions as noted earlier: a) isolated interchange optimization and b) coordinated progression on frontage roads. These modes can be run simultaneously for a total system analysis, but this is quite expensive. The preferred approach is to run the isolated designs first and then using these results, run the progressive analysis (if the latter is needed). Accordingly, the user has to be aware of what input should be included in the respective runs. The two modes are discussed briefly below, then followed by some general remarks.

Isolated Interchange Mode

When one or more interchanges are being optimized independently the essential input requirement is to code a minus one (-1) in each of the five two-column fields to cause PASSER III to determine the optimal internal offset. If analysis only is desired, code actual splits and offsets as ''minimum greens'' and ''priority phasing'' as applicable.

Progressive Frontage Road Mode

In this mode the lower and upper cycle length limits entered may be based on the results of the isolated interchange runs, but should not be more than ten seconds difference for one run. Directional preference for the progression band may be specified for either one-way or two-way, (with or without preference to direction). The speeds input should be based on field studies under ''nonstop'' conditions to obtain ''free speed'' during the time period under study. However, if different link speeds occur and it is not desired to vary the band speed, the average speed should be used (unless it is anticipated that drivers will adjust to slightly different speeds).

General

Options input on the freeway and interchange header cards are used by PASSER III to perform the requested analysis. The volumes can be obtained from field studies or projections; however, the user must be careful to obtain the appropriate counts. Just above the coding columns for this card (see Figure 5) are diagrams showing the 18 movements required. Note that in some cases a movement must be traced through both sides of the interchange.

The second detail card gives the equivalent number of lanes. This is how capacities are input. PASSER III uses a constant lane saturation flow of 1800 vehicles per hour of green time. This may be adjusted (for a single lane and movement) by inputting a factor in the appropriate field. The factor is found by dividing the user's desired saturation flow by 1800. For example, if a left turn lane saturation flow of 1200 vph for movement 15 (see Figure 5) is desired, enter 67. Movements which share several lanes must be assigned their proportional capacity. Finally, the conflicting minimum greens input must not exceed the cycle length (or minimum cycle length) specified. Minimum greens include green, amber and all-red intervals. Sufficient time must be provided for any pedestrian movements.

OPERATIONAL SUMMARY

PASSER III is a macroscopic deterministic time-based optimization model. Since the isolated interchange analysis is distinctly different from the progressive analysis on the frontage roads, it is simpler to discuss them separately.

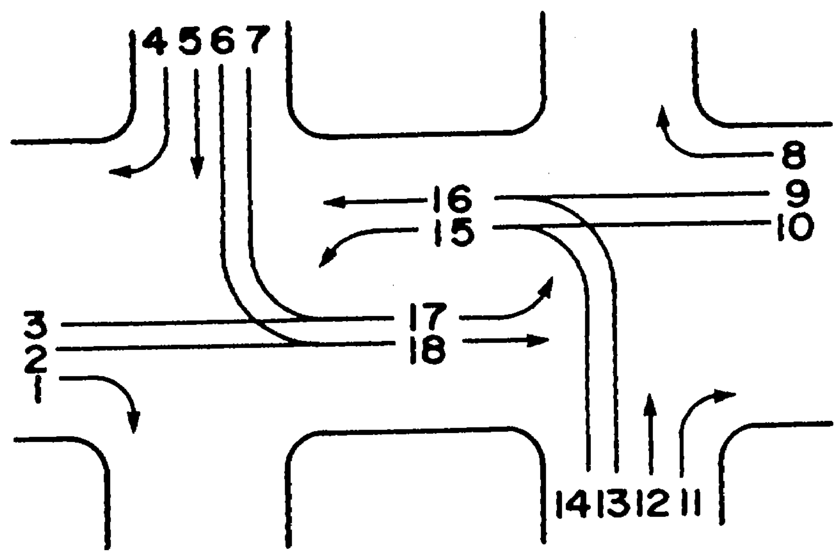

Figure 5 PASSER III Traffic Movements

Isolated Interchange Mode

The interchange optimization is based on the fact that there can exist at each interchange only three basic phases, or allowable greens (excluding pedestrian phases). These are shown for the left-side intersection in Figure 6. These may occur in the order of either ABC (leading left-turns) or ACB (lagging left turns), where the off-ramp traffic either leads or lags the left-turns to the on-ramp. Three similar phases are available at the right-side intersection.

Only certain movements can exist simultaneously at both intersections for any period of time. Thus, the complete set of possible patterns is four, as shown in Figure 7. The fifth code (1A) is a special case of the lead-lead pattern, discussed later. For phasing code 1, queues are forming on the ramps during phase A, in the connecting street on Phase B, and on the ramps during phase C. When overlap is permitted, some conflicting movements can move simultaneously, as shown in Figure 8. Note that the offset is defined as the time between the beginning of phase A on the left side to the end of phase B on the right side.

PASSER III examines all possible combinations of phases (i.e., patterns) and varies the offset to find the pattern and offset which results in the minimum delay in the interchange.

The fifth phase code (1A) shown in Figure 8 is the well-known ''four phase with over-lap'' pattern where the overlap is equal to the internal travel time (i.e., from the stop bar at one intersection to the stop bar of the other). In other words, perfect progression is provided for the through traffic (this may not be the case if phase code 1 (only) is specified).

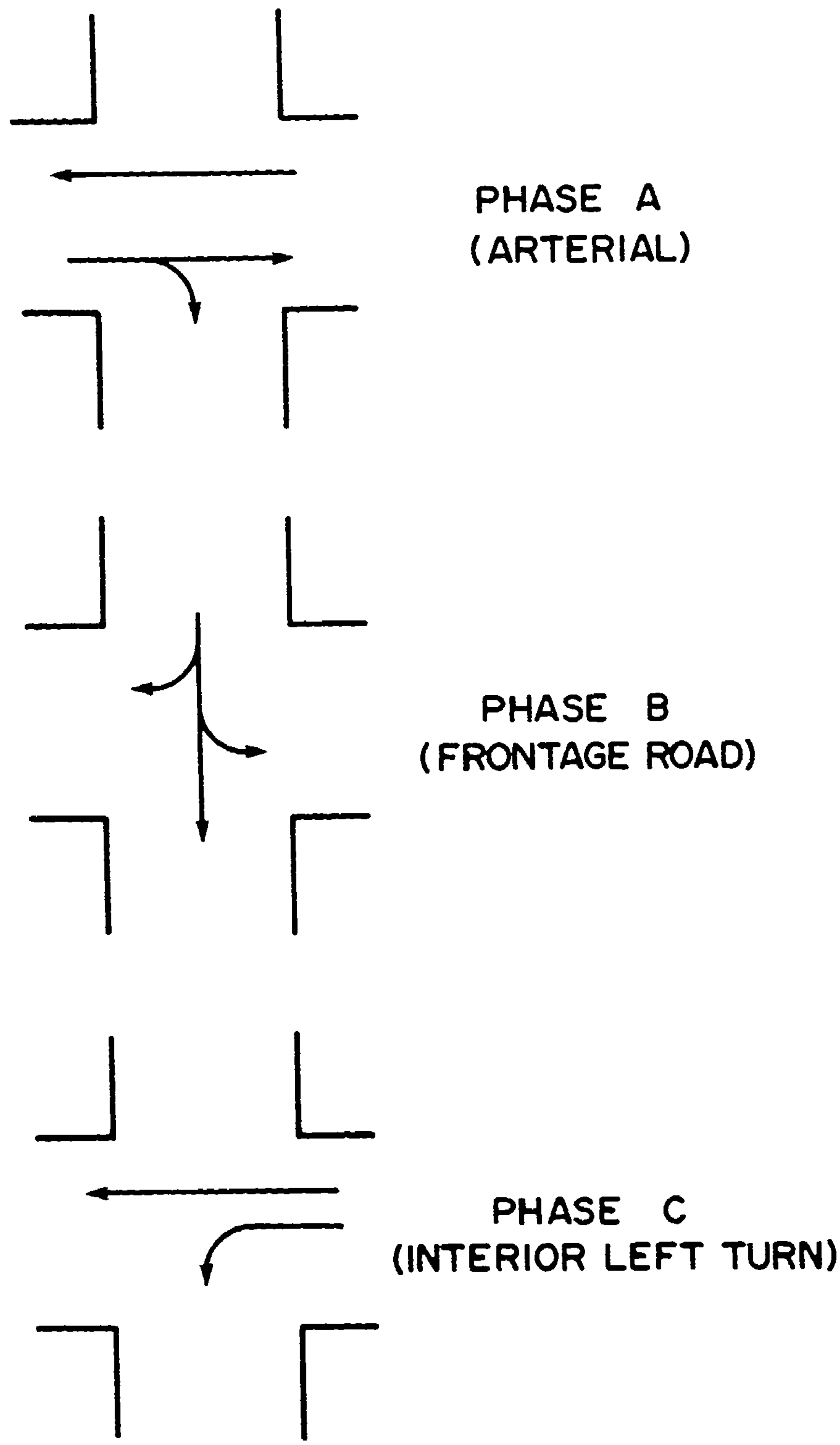

Figure 6 Three Basic Phases at Left-Side Intersection of Interchange

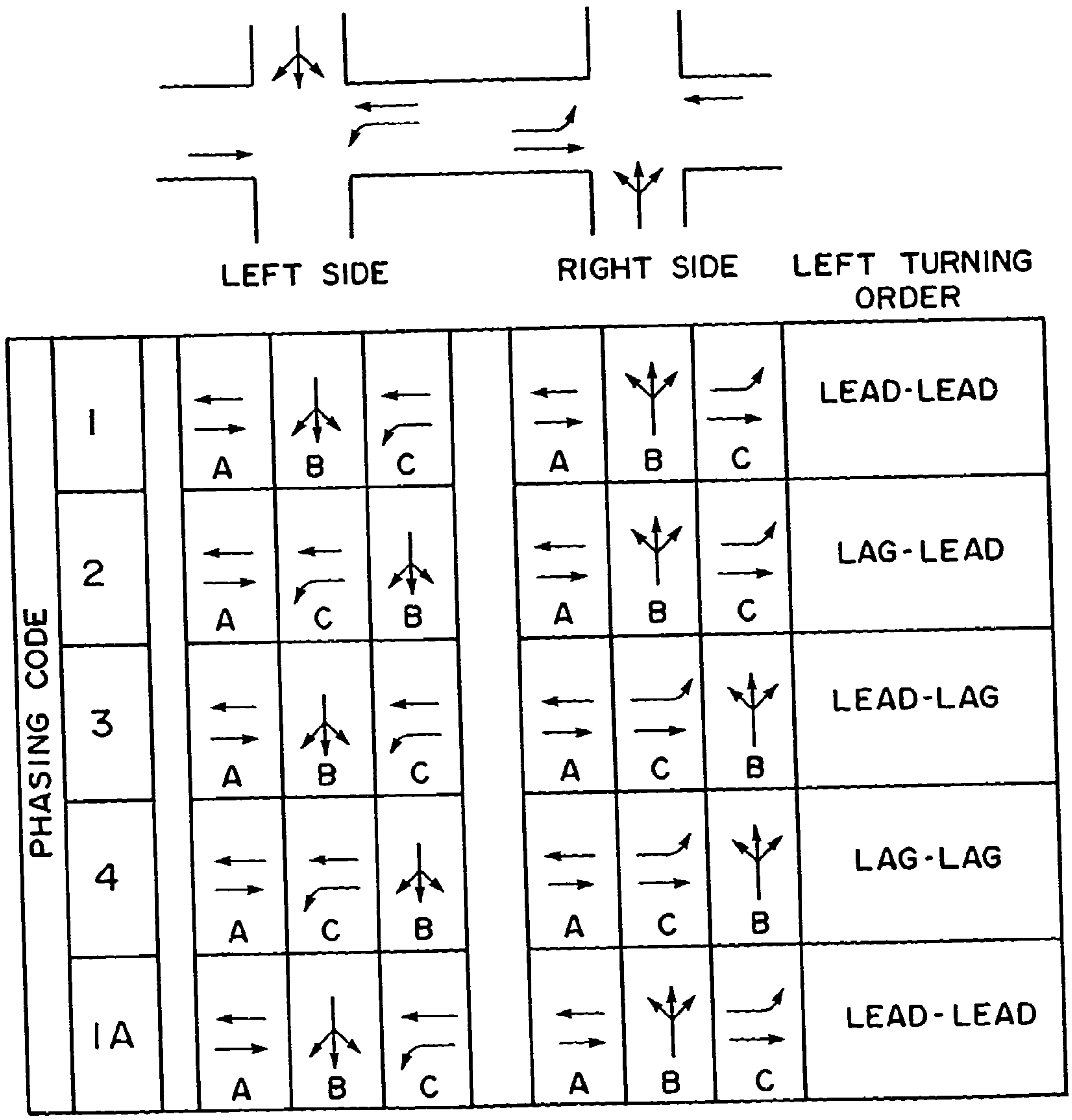

Figure 7 Phase Sequences for Phasing Codes Used by PASSER III

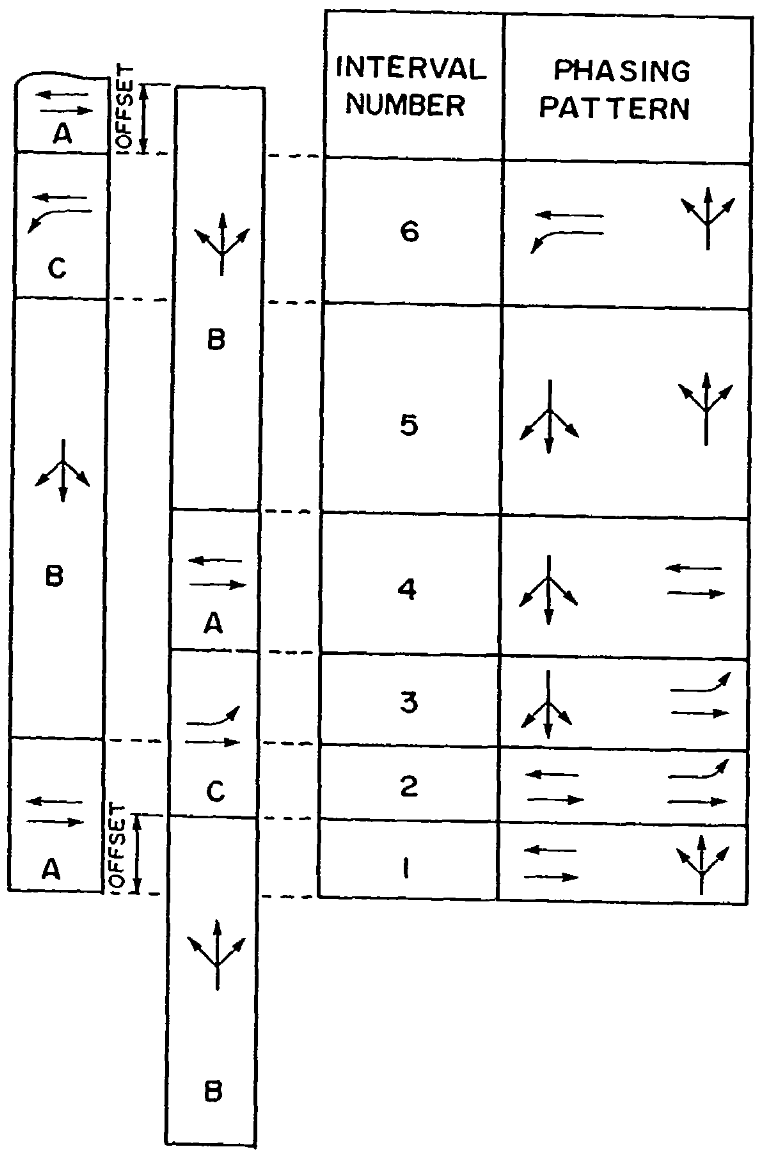

Figure 8 Development of Diamond Interchange Phasing Pattern and Phase Interval Chart from Phasing ABC:ABC and Offset

Progressive Frontage Road Mode

The frontage road progression is independent of the interchange optimization, although the latter should be run to obtain the appropriate phasing and minimums for the progressive analysis. Both analyses may be run together, but the output is extensive and run time high, so the two step method is preferred. The optimal progression design is that which provides the largest ''bandwidth efficiency,'' defined as the sum of the bi-directional bandwidths divided by twice the cycle length.

COMPUTATIONAL ALGORITHMS

The computational algorithms differ somewhat between the isolated interchange and progressive frontage road modes. In the isolated mode the green times are found using Webster's method.

$$G = \frac{Y}{\sum Y} \cdot (C-L)+\ell$$

where: G = green time (sec)

y = volume/saturation flow

$\sum y$ = sum of all y at intersection

C = cycle length

ℓ = lost time this phase (sec)

L = sum of all lost time at intersection

When the four-phase with overlap pattern (Code 1A) is introduced, the green times are calculated using a slightly different formula. An additional term is inserted in the parenthetical expression, which is then $(C + \emptyset - L)$ and $\emptyset$ = sum of interchange overlap (offset) times.

Exterior delay is the delay to all approaches into the interchange (movements 1-14). These are calculated by Webster's method, namely,

$$d = \frac{C(1-\lambda)^2}{2(1-\lambda X)} + \frac{X^2}{2v(1-X)} - 0.65 \left(\frac{C}{v^2}\right)^{1/3} \cdot X^{(2+5\lambda)}$$

where: d = average delay per approach (sec/veh)

C = cycle length

v = approach volume (vps)

λ = proportion of cycle green for this approach, and

X = saturation ratio v/c (c = capacity)

The internal delay for movements 15 through 18 is calculated by the delay-offset technique. However, this technique is too lengthy to discuss here (interested readers are referred to the Documentation).

In the progressive mode the objective is to find the optimal bandwidth efficiency, or

$$\text{Maximize } E = \frac{B_A + B_B}{2C}$$

where: E = bandwidth efficiency

B_A = bandwidth in ''A'' direction

B_B = bandwidth in ''B'' direction

C = cycle length

Probability of clearing the queue is calculated using the following formula:

$$P_c = 1 - e^{-1.58\,\phi}$$

where: P_c = Miller's probability of clearing the queue on the approach

$\phi = \frac{1-X}{X} S\,g$

x = demand volume to signal capacity ratio for the critical lane on the approach

s = saturation capacity flow rate for the critical lane on the approach, veh/sec

g = length of the effective green phase on the approach, sec

Interior storage ratio (S_R) is calculated as follows:

$$S_R = Q_M / S_A$$

where: S_R = Interior storage ratio

Q_M = Maximum queue per cycle on the approach

S_A = Available storage capacity on the approach, veh

For further discussion of the progressive optimization, see PASSER II(80) manual.

OUTPUT REPORTS

There are a total of eight output reports available from PASSER III, but not all are produced in a single run since they vary by mode of analysis (i.e., isolated or progressive). The distinctions are included in the discussion below.

Input Data Report

All input data are printed in well formatted tables. These reports are output in both modes. One table is produced for each interchange. The content of these reports is self-explanatory.

General Signalization Information

In both modes of operation the general intersection information is output for each interchange. This table reports the measures of effectiveness (MOE) for each movement (phase) of the two intersections, along with a corresponding level of service. The first three phases (A, B, and C) are the normal three phases in the pattern. The fourth ''phase'' labeled ''D'' is the time available for the interior through traffic, or the sum of phases A and C.

The green time is the amount of the available cycle for each of the phases (including amber and all-red). The volume/capacity is the ratio of demand to capacity flow in the critical lanes. Delay is the estimate of delay calculated by Webster's method or the delay-offset technique, as appropriate. The probability of clearing the queue values refer to the likelihood that all queues will be cleared on a given cycle for the particular phase. These three MOE's all have a level of service associated with them. The levels of service are determined from Table 10.

The fourth MOE is only available in the isolated mode. The interior storage ratio is the ratio of the length of the maximum queue per cycle for the C and D phases to the available interior storage capacities for these phases. Storage ratio should not exceed 0.8, with 0.6 being a preferable maximum.

Table 10 - LOS Criteria
for MOE's on Signalized Movements
used in PASSER III

OPERATIONAL MEASURES	LEVEL OF SERVICE					
	A	B	C	D	E	F
Saturation Ratio X	$\leq .6$	$\leq .7$	$\leq .8$	$\leq .85$	≤ 1.0	≤ 1.0
Probability of Clearing Queues, Pc	$\geq .95$	$\geq .90$	$\geq .75$	$\geq .50$	$<.50$	---
Average Approach Delay, d, sec/veh	≤ 15	≤ 30	≤ 45	≤ 60	>60	---

Below the output table are the phase orders analyzed, the internal offset identified as having the minimal delay (i.e. the optimal offset) and the total interchange delay in veh-hrs/hr.

Phase Interval Report

A Phase Interval Report is given for each interchange which shows the complete phase pattern including overlaps and the length of intervals.

Optimal Progression Solution Report

When the frontage road progressive mode is run the aforementioned reports are output. Additionally an Optimal Progression Solution report is output. The report includes the optimal cycle length, the progression speed and bandwidth for each direction, the bandwidth efficiency and the attainability. The last value is the average percent of the minimum frontage road green time used in each direction for progression.

Frontage Road Progression Information

Additional internal phasing information is provided for the progression solution. The results are self explanatory. The phase orders and offsets will have been input by the user unless the delay-offset analysis was called for; but as noted earlier, it is strongly recommended that this analysis not be requested simultaneously with the progression analysis due to extreme computer run times.

Time-Space Plot

If requested, a printer or line terminal plot of the time space diagram can be obtained.

ADDITIONAL FEATURES

The two major options have already been mentioned: a) optimization of isolated interchanges and b) optimization of progression on parallel frontage roads.

In the isolated mode, PASSER III can analyze five phasing patterns. The two most popular (but not necessarily always the ''best'') are the four phase with two overlaps (pattern 1A) and the three-phase ''lag-lag'' pattern (pattern 4). This is because the interior of the interchange is always cleared in both directions after the ramp traffic has entered.

In the progressive mode the optimal cycle length is determined to maximize progression. Progression may be one-way or two-way with or without preference to one direction.

Output options include printer or line terminal plots of the time-space diagram of the progressive mode.

PASSER III can be used to evaluate alternative interchange improvements by simply changing the inputs to reflect proposed conditions, such as adding new lanes.

APPLICATIONS AND LIMITATIONS

For design purposes, when the isolated interchange mode is used, the interchange will operate optimally if the resulting offset is used for the particular cycle length and phase pattern specified. To examine alternative solutions, several runs may be made specifying different parameters. The ''best'' solution is that which results in the best overall value of the appropriate MOE's, usually total delay. Other MOE's may be used to override a decision based on delay. For example, if there is a high probability that queues may not be cleared and the internal storage may be exceeded. Other improvements can be analyzed by altering inputs, such as adding lanes.

Although PASSER III is designed primarily to study fixed-time and fixed-sequence control, the delay-offset analysis can also be used to study various full-actuated phasings and to determine the effects of different interchange approach lane configurations, left turn configurations and U-turn lane provisions.

Similarly, the progressive mode is used to design the optimal progression scheme on a system of interconnected interchanges with continuous frontage roads. In this case the optimal cycle length is computed by PASSER III, as are the offsets to obtain progression at a specified speed ($\pm$ 2 mph). Progression may be one-way or two-way depending on the input parameters.

As stated earlier, these two modes should not be run simultaneously, but this is not really a limitation because it is more practical to design the individual interchanges first, and ''fine tune'' them before proceeding to the progression design.

PASSER III (MICROCOMPUTER VERSION)

PASSER III is a computer program developed to assist the traffic engineer in determining optimal traffic signal timings for pretimed or traffic-responsive, fixed-sequence signalized diamond interchanges. The program is applicable to isolated interchanges as well as to a series of interchanges through which progression is desired along the frontage roads. In addition, design engineers may use PASSER III to evaluate the effectiveness of proposed interchange design alternatives.

The use of PASSER III has been limited to users who have access to a mainframe computer. Often, the lack of opportunity or cost of using a large computer has made the use of PASSER III impractical. Because of the growing number of microcomputers in use, a version of PASSER III that runs on a microcomputer would make the program available to a larger number of users. DKS Associates have developed such a version of the program on the IBM-PC.

The PASSER III User's Manual for mainframe version is also applicable to the microcomputer version and is available from the Texas DHPT.

EIBAND

PROGRAM DESCRIPTION - EIBAND is an interactive data processor that assists the user in creating input data files for the arterial case of TW-Bandwidth or MAXBAND (26). EIBAND and TW-Bandwidth run under the MS-DOS operating environment on IBM-PC compatible microcomputers. The data files can be uploaded to a mainframe computer for use with MAXBAND, but this is a slow process and it requires telecommunication devices and special software.

EIBAND and TW-Bandwidth make analysis easy. Before running EIBAND, the user should be familiar with MAXBAND or TW-Bandwidth.

EIBAND presents two types of screens for data input. The first is the Arterial and Program Control Data Screen and the second is the Intersection Data Screen. The second screen is repeated for each intersection in the network. The user simply fills in the blanks to create a properly formatted data set and EIBAND will do the preliminary error checking.

EIBAND will prompt the user for corrections if inconsistent data is input. Certain fields are mandatory, but some may be left blank to use the program defaults. EIBAND also contains default values for many fields that can be input with only one keystroke. Several of the required items were not originally required by MAXBAND, but were included to increase the readability and understanding of the input. Furthermore, they eliminate the need for the user to remember a multitude of conditional defaults within TW-Bandwidth or MAXBAND.

Once a data set has been created, it can be edited using EIBAND. Major changes, such as the addition or removal of several intersections, are as easy to make as minor changes, such as a change in speed.

INPUT - After answering a few initial questions about the data file to be used, the Arterial and Program Control Data Screen appears, and the data entry commences. The data fields are blank when creating a new file. When editing an existing file, the fields are filled-in with the information from the file. The last line of the screen contains messages for the user. This is where error messages may appear. A legend is displayed above the message line and serves as a reminder of the one-key EDBAND commands. These commands are described in detail below.

MOVING AROUND THE SCREEN: The four arrow keys are used for moving from one data field to another.

ENTERING OR MODIFYING DATA: When the cursor is at a field where data is to be entered or modified, the user should start to enter the data. EDBAND will jump to the next field when the data field is filled. If the data is less than the entire field, a carriage return completes the entry. EDBAND then redisplays the data in a uniform format.

CLEARING A DATA FIELD: To clear a data field, and leave it blank, the user should enter a carriage return at the start of the field. This is not necessary when changing data. New data may be entered over old data.

USING DEFAULT VALUES: Many data fields have been assigned default values that may be entered automatically by pressing the F2 Function Key. Default data represents commonly used values which should not be used without a full understanding of the data.

GETTING HELP: A help file is available for almost every data field on the screen. Pressing the Function Key F3 clears the monitor and displays a help file for that field. The edit screen is restored by pressing the F1 Function Key.

ACCEPTING THE DATA ON THE SCREEN: After entering all of the data desired, the user should press the F1 Function Key. EIBAND will perform data checking. If no errors are found, the data will be accepted and a new screen will appear. If there are errors, the computer will beep and an error message will appear at the bottom of the screen.

CORRECTING ERRORS: If EIBAND finds an error after the F1 Function Key was depressed, an error message appears at the bottom of the screen and the cursor moves to the field in error. The user should correct the entry in that field and/or any other fields that caused the error.

ABORTING THE DATA INPUT: If the user wishes to immediately terminate the data input or edit session, the escape key (Esc) should be pressed. Aborting will leave a new file incomplete or empty and will return an old file being edited to its last version.

PRINTING THE SCREEN: A copy of the screen can be made by pressing the F4 Function Key. This is similar to using the Print Screen function of the computer, but presents a slightly different format than appears on the terminal.

The input data needed are: Artery-wide information, volume and capacities or user specified splits, link lengths, and possible left-turn phase patterns to be tested. Default values are available for many of the data fields.

OUTPUT - Input data set for FHWA's MAXBAND program or TransWare's microcomputer program TW-Bandwidth.

SPECIAL FEATURES AND SHORTCOMINGS: EIBAND is a full-screen editor for preparing an input data file for FHWA's MAXBAND program or TransWare's microcomputer program TW-Bandwidth. A menu driven series of screens that allow full cursor control to navigate about the screen quickens data entry with HELP screens available for almost every data field. In addition, extensive error checking is performed prior to saving data to disk.

EIBAND prepares input data files for arterials only, and not for triangular loops.

EZ-TRANSYT 1.2

PROGRAM DESCRIPTION - Since TRANSYT-7F program was introduced to the U.S., it has been widely used by traffic engineers to analyze the traffic condition in a network. Even though it is a powerful tool, TRANSYT-7F has several shortcomings. One of which is the complexity of the input process. The special link-node and phase notations used by the program can be confusing to the users, and special training courses are being offered to aid users in learning how to use the software.

Aiming at the above problems, EZ-TRANSYT is designed as an assistant programmer that can translate the program-specific notations into terms understood by traffic engineers. It also reduces the amount of data that the users have to deal with to the minimum (27).

EZ-TRANSYT is a window-oriented, user-friendly input data processor for TRANSYT-7F. With EZ-TRANSYT, it is easy to maintain a traffic data base of a roadway system. The program allows the user to create a data base file for the network, update the file, print a data summary in readable form, and generate a virtually error-free input data file for TRANSYT-7F runs.

EZ-TRANSYT handles two types of files: A data base file that allows the user to store and update the street network, and an input data file for TRANSYT-7F runs. All the data base files have the extension ''.T7F'' as part of the file name, while TRANSYT-7F input data files have the extension ''.DAT''. It is the responsibility of EZ-TRANSYT to generate ''.DAT'' files, the users only interface with the ''.T7F'' files.

In order to reduce the amount of input data to the minimum, the following default conditions are used by the EZ-TRANSYT:

1. The seconds per step size resolution to be used by the TRANSYT-7F traffic flow model for cycle length evaluation is 3.

2. The seconds per step size resolution to be used by the TRANSYT-7F traffic flow model for final optimization or simulation only runs is 1.

3. Output level will be determined by the program based on Table 4.2.A in TRANSYT-7F Manual.

4. Network-wide start-up lost time and extension to effective green into the clearance interval will be calculated based on the driver behavior specified by the users and typical values listed into T7F manual.

5. Period length is 60 minutes.

6. English units are used in EZ-TRANSYT. The offsets (or yield points) and interval durations are in units of seconds the average cruise speed is in miles per hours and the distance between intersections is in feet.

7. Stop reduction, and special links are not supported.

8. Network-wide saturation flow is set to 1700 vphg/lane.

9. Network-wide platoon dispersion factor will be computed by the program based on the roadway friction characteristics specified by the users.

10. Fuel consumption adjustment multiplier is set to 100 percent.

11. The offset or yield point for every intersection is referenced to the start of the first interval.

12. Double cycle flag is not supported.

13. Maximum number of phases at any intersection is 6. Exclusive pedestrian phase is not supported. There can be up to four intervals for each phase. (Green, Flash Don't Walk, Yellow, and All-Red).

14. Link saturation flow will be calculated by the user-specified phase pattern, and lane numbers. If the left-turn is not protected, the left-turn saturation will be computed based on Tanner's Curve.

15. Input flow from the upstream links will be estimated by the program. The user need only input the volume counts node by node.

16. Bottleneck is not supported.

17. Card 35, 36, 37, 38, 39 are not supported (Speed Multiplier, Flow Multiplier, Delay Weight Modification, Stop Penalty Modification, and Platoon Dispersion Modification).

18. EZ-TRANSYT generates the link lists for flow profile plots and time-space diagram based on the street name specified by the users.

19. Run action 59 (Input Data Check Only) are not used. This option is replaced by the comprehensive error-checking capability of EZ-TRANSYT.

20. Time unit for time-space diagram is in seconds. Time axis scale factor is 3 percent, and distance axis scale factor is 67 feet.

21. Only single run is allowed. The last card in the input data deck is always ''90''.

The page numbers in the parenthesis give the page numbers in the TRANSYT-7F User's Manual where more information is available. If it is decided to change these default values or add more cards, modify the input data file generated by EZ-TRANSYT using commerical word-processing or program editors. The default parameters set by EZ-TRANSYT should be adequate to handle most of the situations, except for a very unusual network system.

SPECIAL FEATURES AND SHORTCOMINGS - EZ-TRANSYT has the following features:

1. No link numbers: Instead of designating a traffic link as meaningless 301 , the link will be referred in EZ-TRANSYT as the North-Bound-Through approach at Main Street 9th Street . The program will make all the necessary conversion to generate an error-free input data file for TRANSYT-7F. The program is intelligent enough to handle special cases such as T-intersections, and some irregular network configurations.

Since the link number notation is not used in EZ-TRANSYT, the relationship between upstream and downstream links are handled automatically by the program. EZ-TRANSYT is also capable of generating the link lists along any arterial in the network that can be used in the flow profile plots or time-space diagrams.

2. Easy Input procedure: First, select a phase combination from a POSIT-like phase menu. Then, instead of using the variable/fixed interval notation, input data directly to Green Time , and Green/Flashing Don't Walk intervals. The program will map these data to the required interval times and generate the link lists that receive green at each phase.

3. On-Line Help Manual. Call up an on-line manual whenever confused. The manual also indicates the page number of the TRANSYT-7F User's Manual for more detailed information. The EZ-TRANSYT manual is not intended to replace the TRANSYT-7F User's Manual published by the Federal Highway Administration which has all the background information needed.

4. Multi-Window Color Display. The program is designed on multi-window technology which is able to clearly display the hierarchy of

the input process on the screen. All the data are displayed in various color windows which can be manipulated by the users. EZ-TRANSYT can work on both color and monochrome monitors.

5. Readable Summary Report. The program can generate a data summary report in a more readable format.

The program is user friendly, and it has an on-line help feature that is very useful.

In order to reduce the amount of input data to the minimum, EZ-TRANSYT uses many default values, which requires that the user be familiar with TRANSYT-7F.

EZ-TRANSYT requires the user to input Green Times in the range of (5-60) seconds. In the case of optimization, the user has to use an external editor to change this field to zero.

NETSIM

MODEL DESCRIPTION

NETSIM, which is an abbreviation for NETwork SIMulation model, composed of the prefix NET for surface street network and the suffix term SIM for microscopic simulation. It is written in FORTRAN IV for IBM OS/360/370 and CDC 6600 computer systems. The current version contains 74 separate routines with a total of approximately 11,000 executable FORTRAN statements and 84 data blocks. The total program length, including comments, continuations, etc., is 14,000 records. The core requirement varies slightly, but (IBM) computers with 280k bytes should be able to execute NETSIM with overlays (28).

Execution time is highly variable. It depends upon the number of links, nodes and vehicles to be simulated. Depending on the complexity, the efficiency may range from about 1:13 (seconds of computer time to seconds of simulated time) to nearly 1:1, but average about 1:2 on large applications. Run time is most sensitive to the number of vehicles simulated.

The model is based on a microscopic simulation of individual vehicles which are moved through the system along the links, according to specified controls at nodes (intersections), stochastically determined turning movements and deterministic car following. No set paths modeled as turning movements are purely random.

The model can investigate a wide mix of traffic control and traffic management strategies, including fixed or actuated signal control, and sign control; special-use (i.e., turn) and general-use lanes; and standard or channelized geometrics.

The capacity of the model may be expressed in the maximum number of nodes (99), links (160) and vehicles (1600 in the network at any instant).

The model is divided into three major components or ''modules'':

Module #1 - NETSIM Pre-Processor: is designed to simplify the process of preparing and checking data inputs. It includes a comprehensive set of automatic ''diagnostic checks'' which are performed on all data inputs.

Module #2 - NETSIM Simulator: contains the main simulation program. It consists of 60 separate routines, which may be linked together in a variety of optional configurations depending on the requirements of the user. The simulator requires as input a coded description of a street network, together with a pre-specified control plan and a set of input volumes. Its output includes a set of standard measures of traffic performance, expressed as both link-specific and network-wide values.

Module #3 - NETSIM Post-processor: consists of a set of standard data manipulation and evaluation routines designed to operate on the outputs of the main simulation program to compare the results of two or more simulation runs, construct a ''historical'' data file summarizing their results and subject the resultant data set to a set of standard statistical analyses.

A variety of traffic controls may be imposed on this network. Intersection controls may take the form of ''Stop'' or ''Yield'' signs; simple, fixed-time traffic signals operating either independently or as part of a coordinated system; vehicle-actuated signals; or more complex signal systems operating under dynamic, real-time control. Up to nine different signal phases may be incorporated within any given signal cycle.

The major features of NETSIM model are listed as:

1) Microscopic, Stochastic simulation of individual vehicle movements.

2) Simulation of full range of control features, including:

   a) ''Stop'' and ''Yield'' Signs

   b) Turn Controls

   c) Parking Controls

   d) Fixed-Time Signals

   e) Vehicle-Actuated Signals

   f) Real-Time Traffic Control and Surveillance Systems

3) Modular Structure Incorporating Detailed Treatment of:

   a) Car Following Behavior

   b) Network Geometry

   c) Grades

   d) Bus Traffic

   e) Queue Formation

   f) Intersection Discharge

   g) Intra-Link Friction and Mid-Block Blockage

   h) Pedestrian-Vehicular Conflicts

4) Provision for flexible mix of standard output measures

INPUT REQUIREMENT

The basic model input is a coded street network which must be accompanied by information about the system traffic control(s) to be studied. Average flow rates must be specified for both the ''entry links'' on the periphery of the network and the ''source/sink'' nodes within the network. In addition, presumed performance characteristics which may include such things as gap distributions, discharge rates, etc., must be input for the traffic movements along each link and through each intersection approach.

The input data may be classified in two ways. First, a distinction is made between characteristics which are considered to be ''location-specified,'' that is unique to a particular link (or node), or ''network-wide'' constants that apply to all points within a network.

A further distinction is made for those two types of data that may be expressed in the model either as exogenous or embedded input. Exogenous input must be specified by the user for each application, and must be read into the model using input control cards. Embedded input are directly incorporated within one or more of the main simulation routines. The embedded data may be changed to suit the user's particular requirements.

The following is a list of the card input requirements grouped by function for the NETSIM (some of these being optional):

- Identification cards - title and network name cards
- Link cards - link name, link geometry, link operation, link turning movements, and opposing link identification cards
- Signal cards - fixed-time signal and traffic actuated signal cards
- Flow rate cards
- Control card - execution control, network priming and simulation control cards
- Surveillance cards
- Bus system cards - path, bus station, bus route, bus flow and dwell time cards
- ''Rare'' event cards
- Embedded data change cards
- Updated data cards

OPERATIONAL SUMMARY

NETSIM is a microscopic, stochastic, simulation model with fixed time-scan updating.

The network is described as a series of unidirectional links and nodes. Each link represents a particular approach to a node and changes in link characteristics (e.g., added or dropped lanes) may be modeled by inserting mid-block nodes. Traffic generators, such as parking lots may be included as ''sink/source'' nodes. A link may contain up to five lanes of traffic plus a left and a right turn pocket.

Traffic demand is initially input to the network via ''entry'' links on the periphery of the system or ''source'' nodes within the network. Upon reaching the periphery or internal sinks, vehicles are processed out via ''exit''links and ''sink'' nodes, respectively.

Within the network, vehicles are propagated through the system along the various links every second, with their time-space trajectories being recorded at 0.1 second resolution. The internal simulation is extremely complex and vehicle motion is governed by a series of car-following, queue discharge and lane changing algorithms.

Within any sub-interval, all conditions (e.g., input flow rates, turning movements rates, signal timing, etc.) are constant. To allow for variation in such variables, several sub-intervals, which may be as short as one minute, or as long as desired, are input.

In order to predict the performance of individual vehicles within the network, each vehicle is randomly assigned various characteristics upon entry into the system. These characteristics, noted in the previous section, are vehicle type, average discharge headway, average acceptable gap, etc.

Nodes are operated according to the type of traffic control specified. Nodes may be Yield or Stop sign controlled or signalized with fixed-time, actuated (both isolated or coordinated) or volume-density controlled. The latter two may involve detectors in either pulse or presence modes.

Depending on the control status and queue length, vehicles are either queued, discharged or processed through the node. Turning movements occur randomly--that is, based on the input proportions of turns, individual vehicles are selected to execute left or right turns. Turns may be protected or unprotected, as specified by the user. In the case of signalized control, up to nine phases may be programmed for any given signal controller.

As the time scan proceeds, data are recorded in vehicle and link arrays. For example, for each vehicle, cumulative time, distance, delay and number of stops are maintained. Additionally, the vehicle's present position (link, lane, position in queue) and projected action at the next node are noted, as applicable.

Link statistics are similar, but additionally include the cumulative number of vehicles and turning movements processed, as well as the current link occupancy, queue lengths and signal status.

In addition to the above statistics, many of which are used for the statistical summaries output by NETSIM, several other aspects of traffic flow are treated to allow a detailed evaluation of the quality of system operation and traffic behavior. These include intersection discharge and queuing behavior, responses to temporary blockages, vehicle-pedestrian conflicts, impact of buses in the traffic stream and impact of various signal control strategies.

The overall operation of the model is summarized in the following seven steps which are performed at each one-second interval within a ''sub-interval.''

1. All vehicles that were located in queues at the commencement of the time step are processed.
2. All remaining vehicles already on the network, but not ''in-queue,'' are processed.
3. Any new vehicles are emitted onto the network via entry link in accordance with the specified flow rates for each entry link.
4. Any new vehicles to be emitted onto the network from any internal source nodes are processed.
5. The status of all traffic signals in the network is updated.
6. The set of standard vehicle and link statistics contained within the vehicle-array and link-array are accumulated and a series of diagnostic checks performed.
7. Finally, if a point has been reached in the simulation run where a statistical output is called for, the necessary results are printed.

These steps are repeated (as appropriate) for each time step and updates of the input conditions are made at the beginning of each subinterval.

COMPUTATIONAL ALGORITHMS

A) Queue Processing

All links, and lanes on each link, are scanned for the presence of queues. When a queue is found, the queue leader is identified and it is determined whether it discharges at this interval or not. If the lead vehicle can be discharged, it is so processed. In this case, the status of all vehicles in the queue is updated to begin moving, and/or record storage time. If the leader is ''blocked,'' vehicle and link statistics are simply updated.

B) **Moving Vehicle Processing**

This is the most complex step in the simulation, as the status of all moving vehicles must be updated. Vehicles are processed from downstream to upstream to allow for car-following, lane changing and the like. For example, the first vehicle on a particular link and lane to be processed will be the next vehicle which would encounter the queue (which has already been processed). Vehicle and link status updates are performed as each vehicle is processed. A variety of actions can occur depending on a vehicle's location, speed (actual and desired), lag, turning assignment, etc. Simplified, a vehicle may follow one of the following actions:

- Speed may be adjusted by a car following rule
- It may join the queue
- It may discharge to another link
- It may change lanes
- It may be designated to exit at a ''sink'' node, or
- If a bus, it may stop at or leave a bus stop

The car following logic is the most complex (that is, all other actions are based on deterministic acceleration, deceleration or lane change rules). The natural variation of desired speeds requires that the car following rule consider the relative position, velocity and desired speeds of vehicles which are interacting with one another (otherwise a simple acceleration/steady-state/deceleration rule can be used).

The basis for this logic is the assumption that a speed profile comprised of deterministic values of acceleration and of deceleration, obtained from the literature, combined with a probabilistically determined desired free-flow speed, is a valid descriptor of interlink vehicle performance in urban networks. The form of this speed profile is shown in Figure 9.

A typical speed profile is composed of five sections:

A) Acceleration from a speed less than or equal to 20 ft/sec: An acceleration of 8 ft/sec^2 is applied (for automobiles) until the vehicle's speed reach 20 ft/sec.

B) Acceleration from a speed in excess of 20 ft/sec: An acceleration of 3 ft/sec^2 is applied (for automobiles) until the vehicle attains its desired free-flow speed, U_F.

C) Constant Speed: Each vehicle is assigned a desired free-flow speed based upon its driver characteristic and the mean desired free-flow speed specified for that link. This speed is maintained until the vehicle is forced to decelerate in response to a standing queue downstream, a red signal, or a lead vehicle.

D) Deceleration from free-flow speed to stop (no moving) lead vehicle: A deceleration of -1 ft/sec^2 is applied until the vehicle's speed has dropped to 10 percent. The distance travelled during this period is found as follows:

$$S_2 = U_f t - \frac{1}{2} a t^2 \qquad \text{and} \qquad U = U_f - at$$

Then, with $.9U = U_f - at$, $t = \frac{U_f}{10a}$ and with $a = 1$,

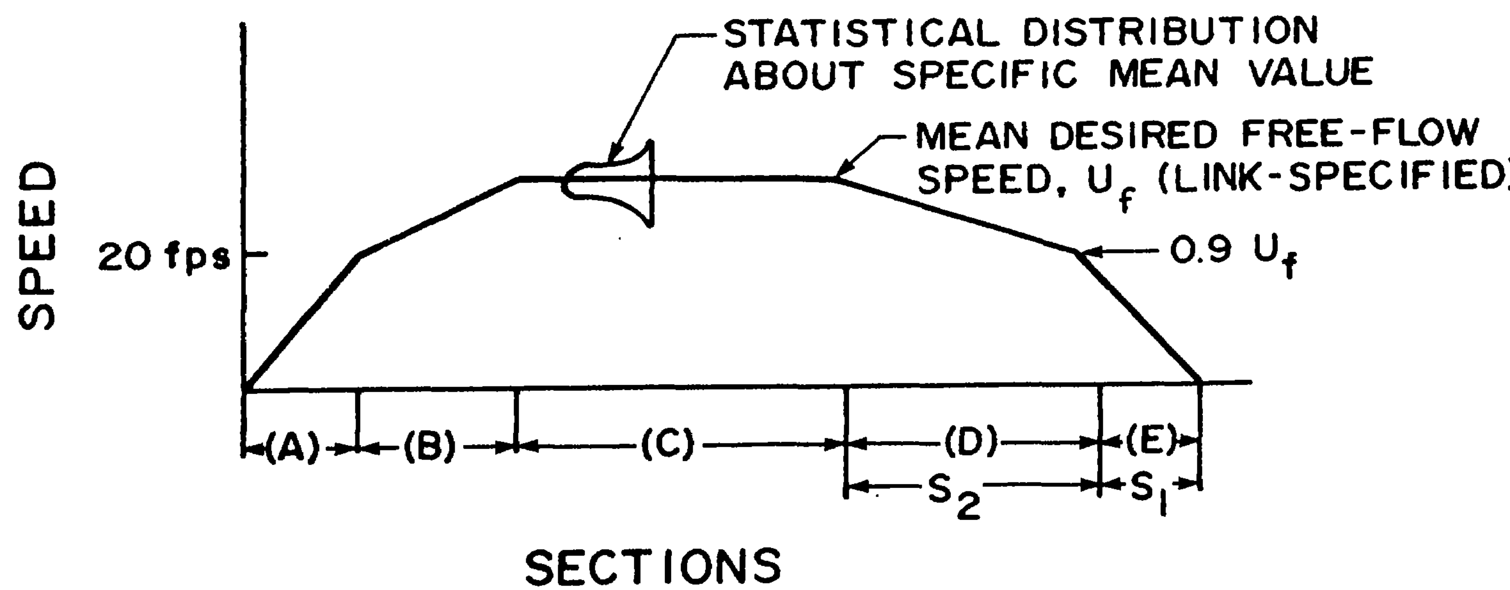

Figure 9 UTCS-1 Vehicle Speed Profile

$$S_2 = \frac{U_f^2}{10} - \frac{U_f^2}{200} = .095\ U_f^2$$

E) <u>Deceleration to Stop</u>: A deceleration of -7 ft/sec² is applied until the vehicle stops. The distance traveled during this period is:

$$S_1 = \frac{(0.9U_f)^2}{2a} = 0.058\ U_f^2$$

The total distance traversed during deceleration is

$$S = S_1 + S_2 = 0.153U_f^2 \simeq U_f^2/7$$

(7 is the nearest integer.)

The speed profile depicted above is applied only to those vehicles whose motion is not influenced by a lead vehicle that is also in motion. But when a trailing vehicle is influenced by a leading vehicle, a stimulus-response model must be invoked. A new model was developed for this purpose based on principles used in many existing car following models, but with changes to improve stability and avoid ''collisions.'' The model, which applies openly to a vehicle following another vehicle within 200 feet is stated as follows:

$$a_f = \frac{7(S - S_f - V_{f_i} - L\ ) + (\frac{1}{3})(V^2 - 2V_{F_i}^2\ )}{V_{f_i} + 3}$$

where: a_F = acceleration of follower at the end of the time slice

S = distance along link

V = speed

L = vehicle length, including three feet clearance

= subscript indicating lead vehicle

f = subscript indicating following vehicle at the end of the time step, or at the beginning if further subscripted by i

If the trailing vehicle's desired speed is reached, further processing is limited to a constant speed model until that vehicle ''catches up'' with the leader which is being stopped.

If a vehicle reaches the periphery of the network (or is assigned to a sink node) it is processed out at this point.

Input New Vehicles, Exterior

This routine scans all links to determine whether a new vehicle should be emitted into the system, effective at the end of the time step. If a vehicle is emitted, an identifying number is assigned and the following characteristics are randomly assigned:

- Driver characteristics
- Vehicle classification (car, truck or bus—buses are processed differently in the simulation)
- Lane assignment
- Action at next node (e.g., turn or through)

Checks are made to determine whether space exists for the desired lane/turn assignments and the vehicle is flagged for lane change(s) if appropriate.

''Arrivals'' at input links are based on a uniform distribution.

Input New Vehicle, Interior

These vehicles are generated at ''source'' nodes within the network. The logic is very similar to that described above. The documentation is not clear as to whether there must be a gap available to accept the vehicle.

Signal Status Update

All traffic signals are updated at this step. All nodes are scanned and at the signalized nodes. The current phase timer is decremented. When this timer reaches zero, the next phase is activated and the timer is reset. For fixed-time, this process is trivial. For actuated control, a routine is called which, for the appropriate type of controller, determines whether current conditions warrant updating the signals. If updating is not required, the controller acts similarly to a fixed-time unit. If updating is required (say a call is received on a semi-actuated approach), signals on all approaches are updated, and appropriate timers (e.g., extensions, minimum, etc.) are adjusted.

Statistics

In addition to simply updating all statistics in the simulation, several other important tasks are performed at this point. These are summarized as follows:

- Insure that all status parameters are consistent, correct if not
- Reset vehicle process codes
- Update all ''event'' actions
- Detect ''spillback'' of queues
- Block or unblock lanes
- Update pedestrian blockages

- Update dwelling buses

The ''events'' referred to above may be short or long-term. These are user specified and may be input to simulate blockages such as accidents, standing or parked cars or other extraordinary perturbations in the network. The exact nature is not specified, but the difference is that short-term events occur randomly for a variable amount of time, and long-term events are preprogrammed (i.e., input directly) and occur on schedule for a specified amount of time.

OUTPUT REPORTS

There are five basic printed output generated by NETSIM. Some of these are automatic and others must be requested by the user. Results may also be stored on tape for future use, such as evaluation by the post processor. The outputs are discussed below.

Input Data Report

General input data are summarized in a formatted report which can be checked for accuracy.

Standard Statistical Report

A summary of important statistics or measures of effectiveness is given at the end of each sub-interval. The cumulative performance on each link and the entire network are printed. The user may also request this report at any time in the simulation (e.g., every ''n'' minutes).

Note that total delay includes both delay due to stops and ''delay'' due to speeds reduced below some specified target level.

Intermediate Outputs

To obtain additional results to augment the above report, the user may request intermediate results at any point in the simulation. These reports are useful in detailed analyses of varying traffic conditions and/or selective (perhaps problem) locations.

Fuel Consumption and Vehicle Emission Report

A summary of fuel consumption and vehicle emissions for each link and the network as a whole is obtained for each run. The fuel consumption data is reported for three types of vehicles; (1) passenger car, (2) truck, and (3) bus, based upon vehicle characteristics coded on the Volume Card (autos and trucks) and Bus Flow Card. Fuel consumption is reported in both gallons and miles per gallon for each vehicle type. Vehicle emission is reported in grams per mile for autos only and includes hydrocarbons, nitrous oxide and carbon monoxides.

Supplementary Outputs

A variety of optional outputs may be obtained for detailed analysis or input data. The following may be tabulated at the user's request:

- The origin-destination pattern of all vehicles
- Types and locations of all detectors
- All ''rare events''
- Bus Performance

Additionally, comprehensive error messages are output to assist the user in ''debugging'' the data or locating inconsistencies in the network description. Special problems are also identified through these messages.

Diagnostic Messages

As noted earlier, there are extremely extensive diagnostic checks and feedback messages available in NETSIM. Errors may occur in several ways. If errors are detected in reading the input data, NETSIM will point out the error and disallow execution of the simulation, but error checking will continue to determine whether further errors exist in the data. The approximate breakdown of the documented errors (i.e., fixup taken, but careful review should be made for possible error) is as follows:

PROGRAM STEP	ERROR	WARNING
Preprocessor	112	1
Simulation	2	5

ADDITIONAL FEATURES

As with most large-scale microscopic simulation models, NETSIM has a multiplicity of features and user options. Virtually any feasible geometric configuration, traffic control system, traffic management strategy and demand configuration can be modeled. The type of network may vary from a single intersection, up to a complex grid network.

Currently, the FHWA is developing an Integrated Traffic Data System (ITDS), a product which allows for the storage of generic traffic data base that interfaces with user-friendly, menu-driven programs for the creation of input data sets for various traffic models. ITDS will take full advantage of state-of-the-art microcomputer hardware and software enabling the preprocessing of the data to create input files off-line, job submission and output retrieval, and the automatic use of optimization programs' output as input to other models. For example, to retrieve optimal timing plans output by an optimization model such as TRANSYT-7F, store those data, and retrieve them for input to NETSIM.

APPLICATIONS AND LIMITATIONS

NETSIM is particularly applicable to the analysis of large-scale complex traffic networks, optionally with coordinated systems (whether master controlled or dynamically controlled).

NETSIM is an operational, analysis and evaluation model. Its sole function is to approximate real-world conditions that are input by the user. It performs no design itself. Thus, in single runs, any of the infinitely variable input conditions may be considered by the user. Several of these may be evaluated by the user to determine which is ''best,'' thus the evaluation function is, to a certain extent, a design tool, but it must be emphasized that the ''best'' solution is only among those alternatives tested. There is no assurance that the ''best'' solution is an optimal solution.

Analysis is the role of microscopic simulation models. The results are generally more reliable than those obtained from macroscopic models, since the natural stochastic variation of traffic demand and behavior are considered. Although any simulation model is a simplification of the real-world, NETSIM is sufficientiy flexible to handle a highly sophisticated system of intersections, including on-line signal control system (with additional user programming).

Also inherent to microscopic simulation models are the disadvantages of costly calibration, extensive input requirements and the requirements for a high level of expertise in using them. These are all true of NETSIM; although the developers have written the model with the user in mind to the degree possible. For example, most parameters are furnished in the model, but these may be changed if the user has local data which would better calibrate the model.

Several specific limitations of the model are discussed below.

1. Physical constraints are 99 nodes, 160 links and 1600 vehicles in the system at any time. These can be increased easily, but a substantial increase in computer time will result to run larger networks.

2. Freeway facilities cannot be modeled in NETSIM. A rough estimate of the effect of freeways on the street system is possible, by making the ramps ''sink/source'' nodes. The freeway effects must be estimated separately.

3. Similarly, rotary intersections and semi-major uncontrolled intersections cannot be modeled except with difficulty.

4. For agencies with limited access to large computers, NETSIM can be quite expensive to use, either in terms of dollars or computer time, depending on local operating policies.

5. It has been noted that real-time control systems can be simulated; however, the algorithms must be inserted by the user. NETSIM does not contain any package dynamic control systems. Surveillance capabilities do exist, however.

6. Preset vehicle trajectories through the network are not possible though they can be estimated by adding additional statements to the code. Vehicles are simply input to the network and turns are assigned randomly. This limits evaluation of one-way street

systems and does not allow for induced diversion from congested streets. The latter would have to be approximated by the user manually ''directing'' traffic via increased turns to effect the diversion or via utilization of the NETFLO model.

7. Input to the system are based on a uniform distribution. This is often not realistic. A patch deck is available from FHWA which corrects this difficulty.

In summary, NETSIM has several limitations inherent to any microscopic simulation model, as well as several limitations peculiar to this model. But on the whole, it is a powerful analysis tool for the traffic engineering agency that has the level of staff expertise and computer facilities to use the model. To overcome this latter requirement (i.e., resources), some state departments of transportation are assisting localities in using NETSIM on their facilities. Thus, mid-to-large-sized urban areas should not be discouraged from using this excellent traffic engineering tool.

NETSIM (MICROCOMPUTER VERSION)

PROGRAM DESCRIPTION - Netsim, a computer model that simulates microscopic traffic flow on urban streets, is one of the most powerful traffic engineering and research tools available today. The use of Netsim has been limited to users who have access to a mainframe computer. Often, the lack of opportunity or cost of using a large computer has made the use of Netsim impractical. Because of the growing number of microcomputers in use, a version of Netsim that runs on a microcomputer would make the program available to a larger number of users. Scott W. Sibley has developed such a version under a research fellowship with the Federal Highway Administration (FHWA), Office of Implementation. The microcomputer program is based on the current version of Netsim with actuated signal logic and is written in Fortran 77 for use on IBM-PC compatible microcomputers. This microcomputer version makes the use of Netsim feasible for an extended range of problems, including much smaller networks and situations in which a number of alternatives are to be examined.

The microcomputer version of Netsim is essentially the same as the mainframe computer version. The Netsim User Guide is applicable to this version, and most data sets from the mainframe version can be used with the microcomputer version with little or no modification.

The program has been modified to fit the limits and capabilities of a microcomputer. The file-handling features have been removed, and the user is given direct control over the input and output files. The mainframe version allows the user to specify several time intervals with different traffic characteristics in each the microcomputer version is limited to simulation of one time interval.

The microcomputer version of Netsim has the same size limitations as the mainframe version. The program can handle a network up to 160 links and 99 nodes and a maximum of 20 actuated traffic signals. Up to 30 bus routes can be included. The program handles a maximum of 1,600 vehicles in the network at one time. Most situations can be handled within these flexible limits.

The computation time of the microcomputer version depends on the size of the network and number of vehicles. This becomes a major factor for very large networks. The simulation may run ten hours or more for a 15-minute simulation on the largest possible network. A more typical 30-intersection network will take approximately 1-1/2 hours. The user only needs to be present during the first five minutes of this time for the input processing.

Performing the Simulation – The microcomputer version of Netsim is simple to use, consisting of three separate programs that combine to provide the complete Netsim package. To obtain useful results, the user should have some knowledge of the fundamentals of traffic engineering and an understanding of the situation being modeled. Some user interaction is required, but primarily the program runs itself using the input files provided by the user.

The first step, which is the same for the mainframe version, is to gather the required input information.

The second step, as with the mainframe version, is to prepare the input data set by using the Netsim coding sheets and the User Guide.

The third step is to run the preprocessor to check the input data set for completeness and consistency.

The fourth step is to run the simulator, which performs the simulation and accumulates the measures of effectiveness. This step involves the major portion of computer time, but the computer runs without assistance from the user. The speed of the simulator depends on the options selected and the reports requested, as well as on the size of the network and the number of vehicles.

During the simulator run, the optional fuel consumption and emissions program can be run for any additional analyses that may be desired, such as using different vehicle performance values. This program uses the vehicle trajectory data generated by the simulator, so the analysis is based on the same simulation. The user can alter the vehicle fuel consumption and emissions tables to see the effect of different vehicle characteristics on the network's performance. Again, the speed of this program depends on the number of vehicles in the network and the time period simulated. However, the program runs quickly, completing most problems in less than one hour.

As the fifth and final step, the user must analyze the output to determine if additional simulations are required.

Advantages of the Microcomputer Version – The microcomputer version of Netsim can be used for problems that would be too expensive or too time consuming on a mainframe computer, making the program ideal for analyzing several alternative solutions to a problem. Apart from the cost of the time required to gather data and prepare the input data set, there is virtually no cost for using the microcomputer version.

With the microcomputer version of Netsim the program can be run whenever a microcomputer is available there is no need to wait for access to a mainframe computer. This can mean substantial savings over renting time on a mainframe computer and makes it feasible to use Netsim with smaller simulation problems and with a larger number of alternative solutions. Also, the output is immediately available it does not have to be returned from a remote computer center. In addition, the program can be easily modified and rerun.

Computer Requirements - The microcomputer version of Netsim, designed to run on IBM-PC compatible machines, runs under either PC-DOS or MS-DOS, version 2.0 or higher. It requires at least 335 K of memory, exclusive of memory needed for the operating system. Two floppy disk drives are recommended, although one will suffice for most problems. A hard disk drive provides additional flexibility and increases the speed slightly, particularly with certain options such as printing the origin-destination table or saving vehicle trajectory data.

PLATOON PROGRESSION DIAGRAM (PPD)

PROGRAM DESCRIPTION - The PPD program produces Platoon Progression Diagrams, similar in function to traditional time-space diagrams. The Platoon Progression Diagram shows the density of traffic at all points in time and distance. It combines the best features of the time-space diagram and TRANSYT-7F flow profile plots (29).

A PPD plot, like a time-space diagram, illustrates the relationship between signal settings at adjacent intersections. However, PPD plots also provide information on the actual flow of traffic along a route, such as the formation of queues or the presence of secondary platoons, that cannot be discerned from a time-space diagram. A PPD plot illustrates traffic flow by shading on a time-space diagram, where darker areas indicate high flow rates or denser traffic. Lighter shading illustrates the dispersion, or spreading out of platoons as they move downstream.

The PPD program works as follows:

- Flow profile data for input to the PPD program is created by either:
  1) Running TRANSYT-7F on a microcomputer and storing the PPD data on a diskette or,
  2) Running TRANSYT-7F on a mainframe computer and down-loading the PPD data to the microcomputer.
- In the PPD program, TRANSYT-7F's platoon and queuing models are applied at 50-foot intervals along each link.
- The results are plotted in bit-image graphical form on a printer. The plots take the form of time-space diagrams that show traffic density levels.

The Platoon Progression Diagram (PPD) program and documentation were originally developed for the Apple microcomputer by K. G. Courgae and C. E. Wallace of Micro-Trans Associates, Inc., Gainesville, Florida.

The program was converted by the Federal Highway Administration (FHWA) for processing on the IBM/PC/XT (and compatible) microcomputers for the MS-DOS 2.0 (or later) operating systems and the Users Manual was revised to reflect IBM/PC/XT specifications.

The PPD program requires the following system components:

- IBM/PC or IBM/XT with one disk drive and the MS-DOS 2.0 (or later) operating system (a hard disk can be used in lieu of or in addition to the floppy disk drive),
- 128K memory,
- Video monochrome monitor or color monitor,
- Epson dot matrix printer (80 or 132 columns).

If running TRANSYT-7F on a mainframe computer, a communication interface is needed with terminal emulator software for down-loading the PPD data to the microcomputer.

The printer is a very critical element. The industry is well standardized for text printing, but not for dot (bit-image) graphics printing. Each printer has its own graphics personality which must be accommodated by the program. The current version of the PPD program accommodates only the Epson MX80 and FX100 with GRAFTRAX.

Early Epson model printers did not have a graphics capability. A ROM (GRAFTRAX) retrofit must be purchased for these printers. More recent models were delivered with this option as a standard feature and should be compatible with the PPD program.

INPUT - The PPD program makes use of special outputs from TRANSYT-7F (Release 4) which are generated by coding a single TRANSYT-7F input field. The special outputs consist of both the timing and geometric data normally associated with a TRANSYT-7F time-space diagram (TSD), plus the flow data associated with TRANSYT-7F flow profiles for those links included in the TSD.

These outputs are provided by a routine called PPD (for Platoon Profile Dump) that was included in TRANSYT-7F for such advanced analysis applications as the platoon progression diagram.

The request for PPD outputs is associated with the TSD, specifically the TRANSYT-7F Card Type 60. Neither the TSD Title Card nor Card Type 61 are affected by the request for PPD outputs, although both the Run Title and TSD Title are included in the PPD text file.

OUTPUT - When running TRANSYT-7F on a mainframe computer, the PPD data is written to Unit IDOUT (default = 8). Depending on your system, you have to provide for this output either as a permanent data set, a time sharing system file, or any recoverable storage medium.

After the TRANSYT-7F program has been run, the data file must then be down-loaded to the microcomputer for input to the PPD program. This can be done easily using a modem and communications interface software to link the micro to the mainframe computer. The data should be stored in a file on either a floppy diskette or a hard disk and the file name must conform with IBM-Basic file name specifications (for example, TESTPPD.DAT).

If running TRANSYT-7F on a microcomputer, the program will ask if PPD plots are wanted and then prompt for a file name. The file can be on either a floppy or a hard disk. As discussed above, the file name must conform to IBM-Basic file name specifications.

Having loaded the data, the program is ready to plot the Platoon Progression Diagrams on the printer. The program will display on the screen the direction and link number currently being analyzed. A dynamic message will indicate which operation is being performed (computing, storing or building queues).

The TRANSYT-7F platoon dispersion model is applied at 50-foot intervals along each link to compute flow rates during each step of the cycle. Queues are superimposed progressing backwards from the stopline by an input-output model. The matrix for each link is saved on a scratch file on the default disk. The file name is in the form ~D.L.~ where D represents the direction (1 or 2) and L is the link number. The links are processed in the same order as they appear in the data file generated by TRANSYT-7F.

The next processing step is to take the matrix values for each link and translate them into fonts (bit-image dots) which cover an area 2 printer columns wide (for each step) and 4 printer rows long (for each 50 foot distance increment). The graphic code numbers (0-9) are translated here into specific dot patterns in each 2 x 4 dot field.

For a two-way street, both directions are printed side by side on the printer. For a one-way street, the plot is expanded to fill across the page.

PROGRAM HIGHLIGHTS AND SHORTCOMINGS - There are several simple rules that must be used when coding the TRANSYT-7F network if using the PPD output capability. (Note: if the user does not use PPD, these rules have absolutely no effect and can be ignored completely.) There are also several suggested guidelines.

RULES -

1) The PPD routine requires that the link numbering scheme recommended in the TRANSYT-7F User's Manual be used.
2) Only through links may be included in a PPD plot. If using shared stopline links, only primary through links can be plotted and the flows shown on the plot will be only for the primary shared stopline link.
3) The route to be plotted may be all two-way or all one-way in either direction, but no mixture of one and two-ways are allowed in a single plot. If necessary, split such sections into separate TSD/PPD's.

There are additional guidelines that should be observed to obtain the best quality of PPD plot, although these are not strict rules like the ones above.

GUIDELINES -

1) Use the maximum number of steps per cycle possible to get the best resolution of the plot.
2) Be particularly careful to ensure that the total flows coded in field 5 of Card Types 28 are accurate for all links to be included in a PPD plot, because all volumes have an effect on the plot.
3) The PPD routine normalizes the data to the nominal number of lanes, L, from the coded saturation flow rate for the link as follows:

No. Lanes	If Sat. Flow Exceeds
1	0 vphg
2	2000 vphg
3	4000 vphg
4	5400 vphg
5	6750 vphg

For particularly low saturation flow rates (say 950 vphg/lane) the PPD program would not estimate the correct number of lanes. Minor adjustments to meet the above thresholds are probably acceptable.

A PPD data file may contain data for more than one plot if multiple time-space diagrams were included in the TRANSYT-7F.

PRETRANSYT

PROGRAM DESCRIPTION - Barton-Aschman Associates, Inc. has developed a package of traffic engineering and related programs which provides quick, accurate and consistent analysis results of traffic and transportation-related engineering problems. These programs are known collectively as TEAPAC, an acronym for Traffic Engineering Application PACkage. Over 20 integrated programs are included in TEAPAC, all of which use the same basic set of commands to direct the actions of the program. The programs also share the same characteristics with regard to the format of commands, their conventions, and their storage format in data files. PRETRANSYT is one of the programs in the TEAPAC package (30).

PRETRANSYT is a preprocessor to be used in conjunction with the TRANSYT program. TRANSYT requires a rigid input stream of specially numbered card types and fixed-format input fields. On the other hand, PRETRANSYT takes data in a free format directed by commands, and builds the fixed format TRANSYT input, thus eliminating user error in coding card types and preparing fixed format input.

This interactive analysis tool will generate the network structure for the signal system being analyzed through the use of the NETWORK command, thus the user does not have to specify all of the movements throughout the network. Further, the phasing and timings for each signal is specified in straightforward traffic engineering terms, and can be changed quickly when testing alternatives. After an analysis by TRANSYT is completed, PRETRANSYT has the capability of plotting a time-space diagram for any portion of the system. Because of the speed of the computer and the ease of input, an entire day's variation in traffic conditions can be analyzed quickly and accurately. This, in combination with the capacity analysis output of

SIGNAL, provides all the information needed to implement the optimum timings in a network.

The maximum number of intersections allowed in one run has been set to 45 nodes, and the number of characters allowed in the intersection names is limited to 16.

The TRANSYT methodology simulates the total delay and stops to vehicles traveling within the network, and seeks to set the signal timings to attain a minimum combination of delay and stops. The procedure provides a concise and well-founded operational design for a signal system, rather than those which have previously been performed using less accurate manual estimates. The program can be used to simulate existing signal system operations as well as to optimize operations. The analysis pays special attention to heavy turning movements within the system, as well as exclusive turning phases, a refinement which is poorly covered by traditional linear progression analysis. The user has the option of having the program optimizing signal splits or offsets, or both of these parameters.

There are few useful commands within PRETRANSYT. These are as follow:

1. ASK Command: It requests parameter values from the user for each of the listed commands.
2. DATA Command: List the parameter values for specified commands.
3. BUILD Command: Construct fixed format TRANSYT input files.
4. HELP Command: Print the command names, parameter descriptions, and default values for each command listed.
5. PLOT Command: Plots a time-space diagram for the nodes specified.
6. RESET Command: Reset parameters of specified commands to default values.

7. SUMMARISE Command: Print a formatted summary of all PARAMETERS values.

SUMMARY OF PARAMETER VALUES - The network progression parameter summary generated by the SUMMARISE command is a compilation of the data pertinent to analyzing the network progression, recognizing all the pertinent information required by the TRANSYT program. The parameters are discussed below in the order in which they appear on the report.

System Parameters

Cycle. The cycle length, in seconds, for the time period being analyzed.

Simulation. The number of simulation steps in the cycle length, the length of time, in minutes, being simulated, and a third parameter which allows the definition of a global stop penalty for the performance index.

Lost Times. The amount of lost time at the beginning and end of each phase.

Optimize. The parameters that are being optimized such as splits, and/or offsets, or a user-specified list.

Node List. The list of node numbers that are being considered.

Link List. The list of links, in any arbitrary order when building TRANSYT input for Version 7 and above. The linklist can also be used to force the program to simulate all links at an intersection, regardless of whether they feed the internal links of the system.

Intersection Parameters

Intersection. The node number and name of the intersection.

Network. The description of the location of intersection in the network being analyzed. This includes the direction, distance and travel speed from the upstream node, and the movements supplying traffic at that node.

Volumes. The design hour volumes at the intersection for each of the 12 turning movements at the intersection.

Widths. The widths available for each of the 12 movements at the intersection.

Saturation Flows. The capacity at Level of Service E in vehicles per hour of green for each traffic stream.

Sequence. A two-digit code representing the phasing of the traffic signal.

Lead/Lags. For multi-phase signal operation, indicates whether turning phases lead ahead of or lag behind the through phase.

Permissives. Denotes whether left-turns are allowed on a green ball signal indication following an exclusive phase (arrow).

Phase Movements. Identifies which movements are allowed on each signal phase, as defined by the SEQUENCE command.

Green Times. The green time, in seconds, for each signal phase.

Yellow Times. The yellow time, in seconds, for each signal phase.

Offset. The system progressive offset for the phasing, and the phase number to which this offset applies.

Output. The type, amount of output desired, and the third parameter which allows the generation of modified TRANSYT 7 input to be used with the TRANSYT 7F (Release 2) Program. The default value of this option is 6 for TRANSYT 6 input a 7M may be used to produce modified 7 input.

SPECIAL FEATURES AND SHORTCOMINGS - PRETRANSYT has the following special features:

1. Output can be generated for TRANSYT 6./7./7M/7F2/7F3.

2. It is very handy in assigning links to each node.

3. It has a good error checking capability.

4. It has default values for most data.

5. It has a short Help for every command.

6. Time-Space diagram can be plotted from the offsets obtained from TRANSYT program or inputted by the user.

7. It has the capability of recognizing single and double cycles.

8. The SUMMARISE command will produce a neat and helpful data summary.

Among the shortcomings the following can be listed:

1. The documentation is fairly old. It is extremely hard and time consuming and practically impossible to master this software with existing documentation.

2. The output generated for TRANSYT 7F3 is hardly used anymore instead the new version of TRANSYT 7F4 is used by majority of traffic engineers.

3. The program structure is not made to be user-friendly.

4. While building the output for TRANSYT, if the program detects any error, it terminates the session and the user has to start from the beginning.

5. The program will do the general data codings. In order to do plotting and other special cards, the user has to use an external editor.

6. It requires the user to be familiar with input coding of TRANSYT.

SIGOP III

SIGOP III is a macroscopic signal timing design and analysis model. It contains two primary submodels: 1) a traffic flow submodel, and 2) an optimization submodel which minimizes a user specified ''disutility'' function (31).

SIGOP III uses the underlying principles of the TRANSYT model, and was based upon the following objectives.

1. Develop a new, improved optimization procedure
2. Improve effective utilization of the model
3. Enable explicit representations of the traffic environment, including exclusive turning bays
4. Consider the effect of extensive queueing to prevent ''spillover'' into upstream intersections
5. Explicitly consider multi-phase control
6. Include useful features of other models

SIGOP III originated from the SIGOP model, but most of the difficulties with the earlier model have been overcome. Several SIGOP features, notably the time-space plot capability, have been retained in SIGOP III. It considers delay, stops, and a term for queue ''spillover'' in calculating the optimization objective function.

SIGOP III is a powerful analysis and design tool. Preset conditions, such as existing conditions, may be analyzed in terms of a number of useful traffic engineering measures. The signal timing may be optimized for cycle length, splits and offsets to minimize the ''disutility'' function. Comparisons of results of several candidate configurations enables the engineer to evaluate the relative effectiveness of the alternative designs.

SIGOP III was developed by KLD Associates, Inc. for the Office of Research, Federal Highway Administration (FHWA). The model is maintained by the Implementation Division of FHWA, thus the utility and useful life of the model should be both current and reliable.

The input parameters are greatly improved over the original SIGOP. Data requirements for SIGOP III are relatively less than TRANSYT and NETSIM but more than PASSER II(84).

MODEL DESCRIPTION

SIGOP III is an acronym for Traffic SIGnal OPtimization Model, version III. The program is written in FORTRAN IV and has successfully run on both CDC 6600, IBM 360 and 370, and Amdahl 470 computer system. The current version contains 34 subroutines and 23 common blocks. The FORTRAN program is approximately 7,900 lines in length of which approximately 76 percent are definition and executable statements. The program requires approximately 300k bytes of core storage on an IBM 360 computer, but an overlay structure reduces the space requirement to 200k bytes.

Execution time is variable and depends upon the number of intersections (nodes) and the number of cycle length iterations. The computing time varies approximately linearly with the number of nodes and cycle iterations. Thus, even large networks can be optimized in a relatively short time, and computer time is comparable to recent versions of TRANSYT.

The study network can presently consist of a maximum of 80 nodes and 230 links, however, the developers have given instructions for expanding the capacity of the program.

SIGOP III is a macroscopic, deterministic, simulation and optimization model with a periodic time scan over the solution space (e.g., cycle lengths, offsets and splits). The optimization technique employs a gradient methodology to scan the feasible solution surface to be confident that the system-wide global optimum solution is found. The model uses an application of a technique referred to as the ''Method of Successive Approximations'' that shortens the solution times.

The model deals exclusively with mixed-flow traffic on a signalized arterial network. Multiple approaches (e.g., diagonal streets) are permissible and signal timing is assumed to be fixed-time, but with multiple phasing.

The model contains four main program segments which are: 1) an executive module, 2) an initialization module, 3) a traffic submodel, and 4) the optimization submodel. The program structure is shown in Figure 10.

INPUT REQUIREMENTS

There are 13 types of input cards needed to run SIGOP III. A standardized input format has been designed for the benefit of the users. Alphabetic information is input to name the network and streets. Most numeric data are input in standard four-column integer fields.

The network structure is input by identifying each node. Links are identified by link-end node numbers, thus a link running from Node 1 to Node 2 would be ''named'' Link (1,2). Turning bays are handled explicitly, rather than via separate links. The flow through the network is further identified by inputting the downstream node number receiving through traffic from each link. Only internal links carry traffic. External links (identified by having an external node number of 800 or greater) serve only as input sources, or exit sinks, and no travel occurs in these links.

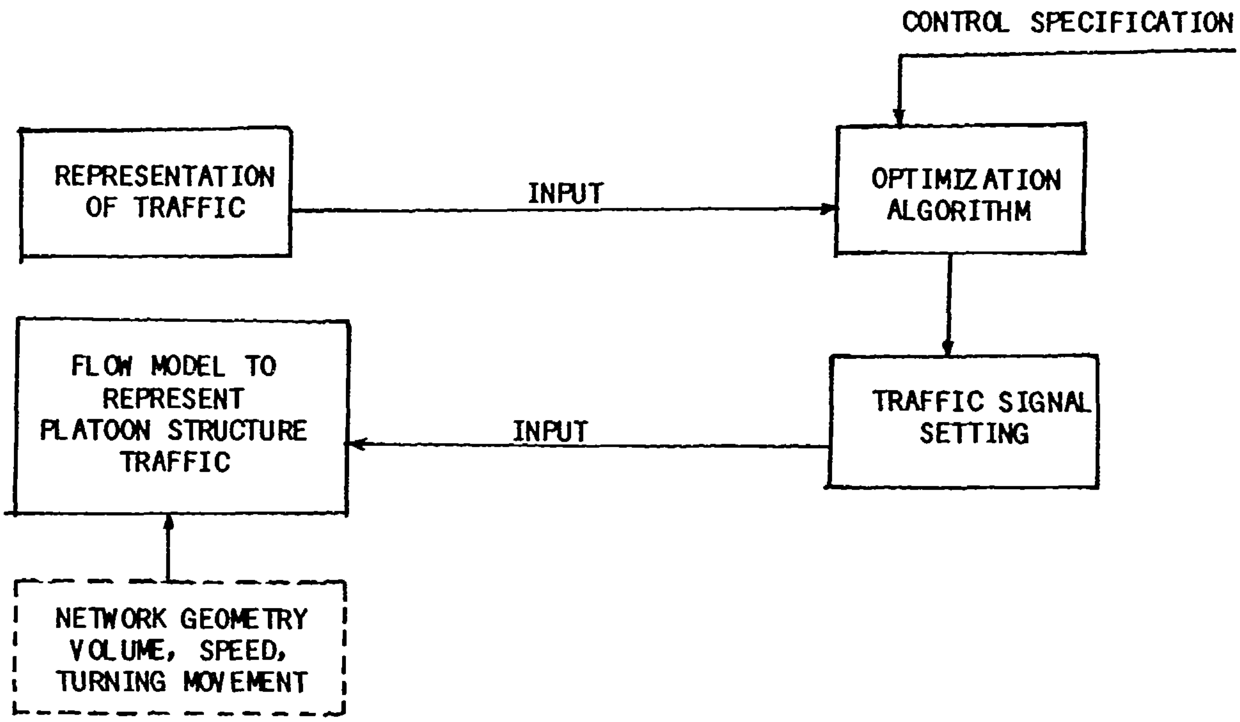

Figure 10 SIGOP III Program Structure

Neither of the 800 level nodes included in the network. Queuing and delay do, however, occur on external input links.

Mid-block sources or sinks may be included to reflect the affect of parking lots, shopping centers, etc.

Links that share common, or paralleled, movements may be coupled together and, thus, move on simultaneous phases.

Volumes must be specified, both in terms of input and output. A ''primary'' volume is the through input from upstream. ''Secondary'' volume is that from other upstream movements, such as turns from cross streets (excluding sink/source flows). Output volumes are expressly input as to turning movement, where the through output volume is calculated, as the sum of all inputs, less the sum of output right and left turns. Since the input/output flows are specified per link, volumes need not ''balance'' from node to node. This is convenient since data collection techniques are rarely sufficiently accurate that volumes do balance.

Signal patterns are input in a fairly easy manner. The steps are given below:

1) Diagram the phase patterns (for example the four phase scheme is shown in Figure 11) and identify the movement diagram for the link in question (say from left, or eastbound, in the 4-phase example).
2) Determine which phases carry the through movement from this link (e.g. phase 2 and 3).
3) Enter Table 11 to determine the code for this link that satisfies the phases determined in step 2 (e.g., code = 23). This value is entered in the first of three fields on the Link Card that is provided for patterns.

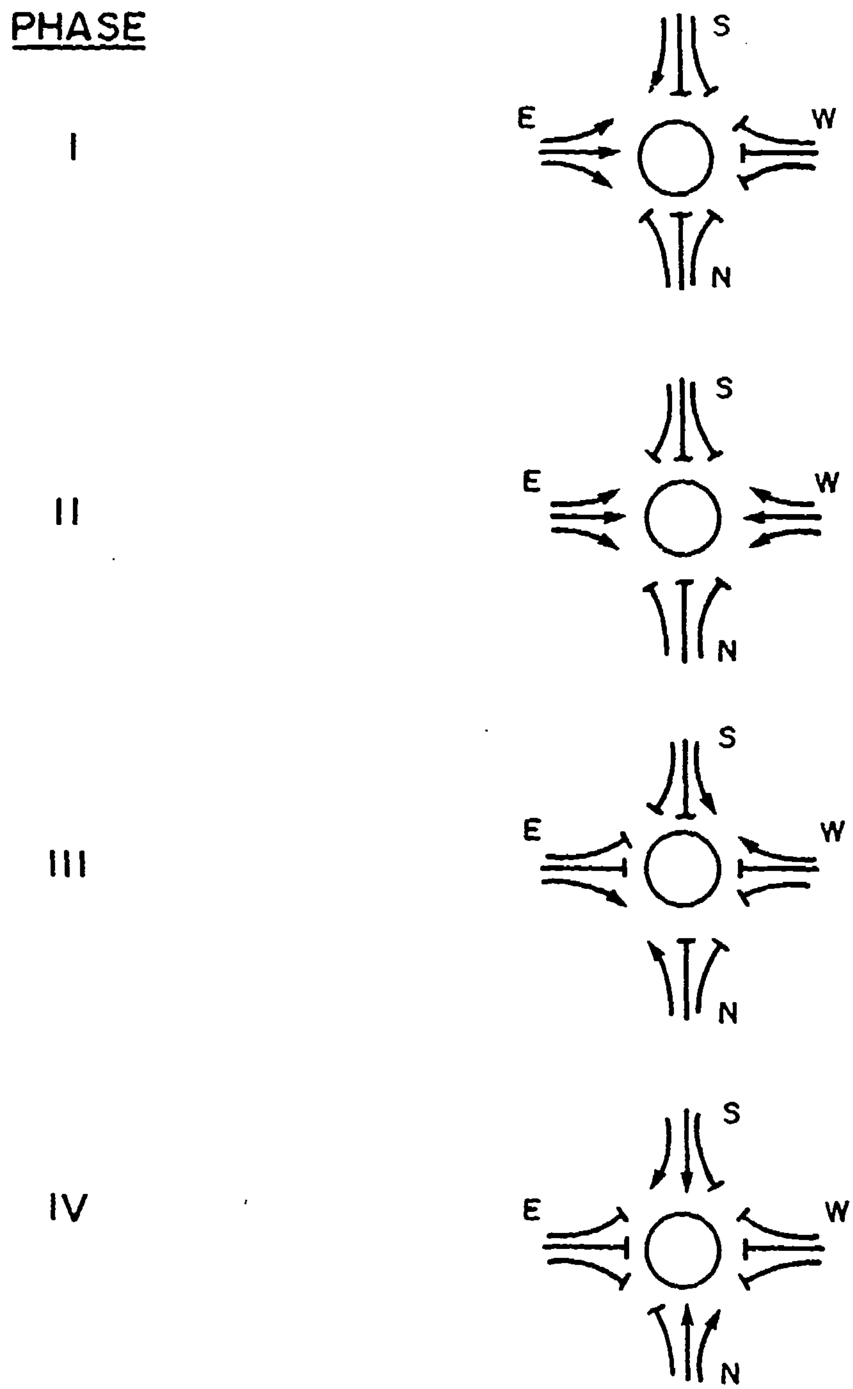

Figure 11 Illustrative Node Diagram Depicting Signal Phasing and Traffic Movements Serviced: Four Signal Phases

Table 11 - Input Phase Codes for Link Card of SIGOP III

Code	Phase(s) Servicing Indicated Movement			
	I	II	III	IV
1	X			
2		X		
3			X	
4				X
12	X	X		
13	X		X	
14	X			X
23		X	X	
24		X		X
34			X	X
41	X	X	X	
42	X	X		X
43	X		X	X
44	X	X	X	X
45		X	X	X
50	Movement is not services			

4) Repeat steps 2 and 3 for left-turns (e.g., code = 12) and right turns (e.g., code = 50). These codes are entered in the remaining two fields of the pattern section on the Link Card.

5) Repeat steps 1 through 4 for all remaining approaches at this node, and subsequently for all nodes.

Note that for less than three phases per approach, the code is identical to the phase numbers continuing in order; thus, the user should quickly become familiar with the coding scheme.

The major disadvantage of this approach is the limitation to a 4-phase cycle. Many controllers operate on five or six phases even in fixed time operations. The advantage is that it provides an easy to understand encoding scheme.

Capacities of movements are input in terms of the numbers of lanes, start-up lost time and minimum discharge headways. The latter value, for a given link, is the reciprocal of the maximum vehicle service rate; thus users who normally work with capacities (vphg) can easily convert to minimum discharge headway.

Signal offsets and splits may be input to analyze preset (e.g., existing) conditions. Furthermore, if the user desires, selective offsets and splits may be input which cannot be changed by the optimization model. This feature may be used, for example, when optimizing a very large system, by segmenting the network into groups of 80 or fewer nodes. The ''border'' street(s) could be optimized in one segment, then fixed in the adjacent segment.

In summary, the inputs to SIGOP III are functionally similar to TRANSYT. Both programs have some advantages over the other in terms of ease of coding, although SIGOP III appears to have a slight advantage in this regard. One minor problem with the SIGOP III coding scheme is the necessity to indicate, on one card, the identification of the following card. The codes for ''next card'' vary among cards; thus, the user must always be attentive to the ''current'' card.

OPERATIONAL SUMMARY

As noted before, SIGOP III is a macroscopic, deterministic model with a traffic submodel and an optimization submodel.

The network is formulated as a system of nodes with unidirectional links between nodes, as required. External links have pseudo nodes, indicated by node numbers greater than 799. An example of a network is shown in Figure 12.

Traffic appears on the external input links and is assumed to arrive uniformly at the input stop line. Within the system traffic is assumed to travel in platoons which disperse over downstream links according to 1) the time of release upstream, 2) the distance traveled and 3) the free speed.

Upon the onset of green, and after the initial start-up and acceleration lost times expire, any existing queue is assumed to discharge at the saturation flow rate. The traffic moves in a coherent platoon along the link, but dispersing (i.e. lengthening) as it progresses. Robertsons' platoon dispersion technique is used. Delay, stops and queuing can thus be computed, given the predictable arrival and departure profiles of the traffic. The methodology for these computations is discussed in the next section.

The above describes the traffic submodel briefly. The traffic model can be exercised for each link, given the signal timing of the upstream and downstream nodes. The optimization process thus searches for a set of

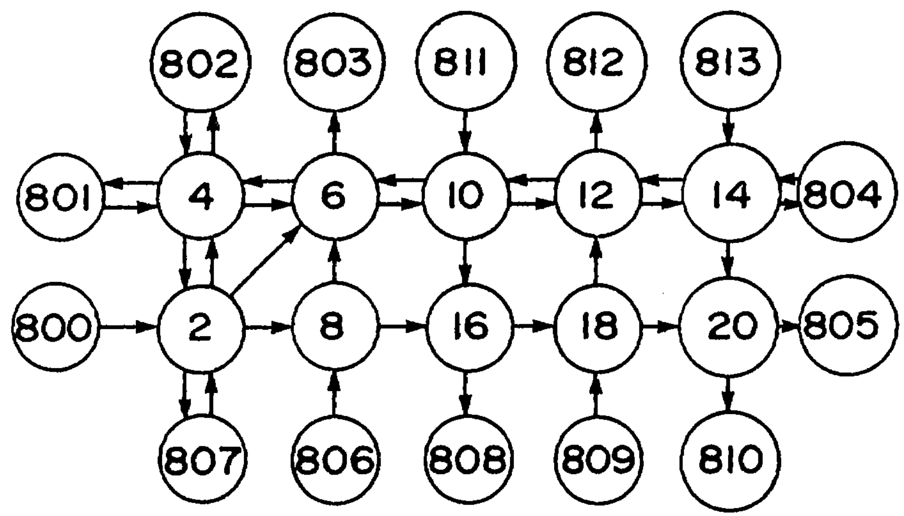

Figure 12 Typical SIGOP III Network

signal timings (offsets and splits) that minimizes the ''disutility function'' (defined later). By switching the signal timing according to a rule, the effect on traffic flow is recalculated and the disutility is compared with the previous value. If improvement (reduction) results, the model continues to ''search'' until disimprovement is encountered. By repeatedly evaluating changes in the disutility, due to new signal settings, an optimal condition, or design, can be determined. The optimization technique is also discussed further in the next section.

COMPUTATIONAL ALGORITHMS

From the foregoing discussion, it might appear that the computations of the SIGOP III model, while numerous, are somewhat trivial. This is not the case. Several sophisticated techniques are employed in both the traffic submodel and the optimization submodel. Indeed, the calculation of splits is of interest as well. Once the signal timing has been completed for any given iteration, the traffic submodel is entered to obtain the measures of effectiveness (MOE) followed by the optimization process. The computational algorithms in each of these steps are described below, in turn.

Signal Timing

Signal timings input to the traffic submodel are cycle length, splits and offsets. The cycle length range is a user input as is the increment of cycle length. Thus cycle length is constant for each iteration analyzed. Offsets are affected in the optimization process and are discussed later.

Unlike TRANSYT, which allows all splits to vary (subject to the minimum green constraint) to achieve the lowest value of the objective function, SIGOP III calculates minimum green requirements using Webster's method:

$$g_K = \frac{Y_k}{Y}(C-L),$$

where: g_k = green time required to service traffic on approach k

Y_k = critical volume/capacity ratio for approach k

$Y = \sum Y_k$

C = cycle length, and

L = total lost time per cycle

Then, if the sum of these green times is less than the cycle length (e.g., $\sum g_k < C$), the remaining ''slack'' time is allocated to the major movements only in the optimization.

Traffic Flow and Measures of Effectiveness (MOE)

The MOE noted previously are delay, stops and queue length. The developers of SIGOP III have conducted extensive investigations to relate the offset/split relationship of adjacent signals to traffic flow on the link. The entire process is too complex to relate here, and interested readers are referred to the ''SIGOP III: Program to calculate optimal, Cycle-Based Signal Timing Patterns. Vol. 2: SIGOP II Formulation and Software Documentation,'' FHWA Report, September 1976. Critical to all the calculations is the assumption concerning platoon dispersion. Robertsons' method was found to be satisfactory, but to eliminate the recursion relationship from the computations (and thus save computing time), a series of studies were performed to replace Robertson's recursion formula with a direct estimate of the additional time required to service a platoon of, say, length N beyond the time the platoon is discharging at the saturation flow rate. Thus, the total green time required to service the bulk of a platoon (e.g., allowing the relatively small number of vehicles, having long headways, at the tail of the platoon to be ''clipped'' off) was derived as:

$$\bar{V} = T_P - T_Q = a_1 + a_2N + a_3N^2 + L\ (a_4 + a_5N + a_6N^2)$$

where: $\bar{V}$ = additional time to service the platoon relative to the saturation service rate

T_P = time for a platoon of length $N - N_c$ to pass a point located L feet downstream of the signal

T_Q = time required to service a platoon discharging at the saturation rate (e.g., T_Q = Nh, where h is the saturation headway in sec/veh)

a_i = constants of regression

N = total number of vehicles in the platoon

L = distance to the downstream point, or the next stopline

According to the developers, close comparisons resulted from this technique when compared to Robertson's approach.

This approach eliminates the step-wise simulation used in TRANSYT; thus MOE must be calculated deterministically. The current version of SIGOP III calculates delay similar to Webster's method, namely for light flow:

$$D = \frac{C(1-\lambda)^2}{2(1-\lambda X)} + \frac{X^2}{2q(1-X)}$$

where: D = average delay in sec/veh

C = cycle length

λ = proportion of the cycle that is effectively green

q = flow rate

X = degree of saturation

For moderate to heavy flow the revised equation is:

$$D = \frac{C(1-\lambda)^2}{2(1-\lambda X)} + \frac{IH(\mu)X}{2q(1-X)}$$

where: I = variance of the number of arrivals per each cycle divided by the average number of arrivals per cycle

$H(\mu)$ = a complex function of μ, that shears Webster's curve through the region where the q/s ratio is close to or exceeds 1.0,

where: $\mu = (sg-gC)/Isg$

g = effective green time and

s = saturation flow in veh/sec; and

all other variables are as previously defined

Stops are computed for many different conditions and the user is referred to SIGOP II Formulation and Software Documentation.

The SIGOP III documentation is not clear as to how queue length is explicitly determined. The necessary ingredients are, however, available from the queue profiles. The maximum length is controlled by the user in the parameters input to the disutility function.

Finally, SIGOP III has the facility to automatically examine double cycling of signals if the degree of saturation does not exceed a threshold, also input by the user. This is a very convenient method of examining double cycling.

Optimization Submodel

Most traffic signal optimization models employ some sort of iterative methodology to arrive at the optimal design. SIGOP III employs a unique approach in its optimization process.

First, the objective function (disutility function) is defined as follows:

$$\min \sum J_{ij} = \sum \left[D_{ij} + kS_{ij} + \frac{\delta[D_Q(Q_{max} - Z_{ij})^2]}{R^2} \right]$$

where:

J_{ij} = disutility on link ij during one cycle

D_{ij} = delay on link ij per cycle, veh-sec

S_{ij} = stops on link ij per cycle, veh-stops

k = user specified equivalence factor for stops

D_Q = user specified equivalence factor in veh-sec

$(Q_{max})_{ij}$ = estimated maximum queue length on link ij, in feet

R = user specified value of residual storage desired on all links beyond $(Q_{max})_{ij}$ to prevent spillback arising from short-term fluctuations in volume, in feet

Z_{ij} = the distance from the downstream stopline back to the previous intersection, or L_{ij} − R, where, L_{ij} = link length; and

= a binary index which is zero (0) if $Q_{max} \leq Z_{ij}$ or one (1) if $Q_{max} > Z_{ij}$

The third term, controlled by the index, is not involved unless the maximum queue treatments to spill back into two upstream intersection.

The user controls the objective function through his inputs of the values k, D_a, and R.

The optimization process is algorithmic, rather than analytical,thus it is described below in steps.

Step 1 - Initial Settings - At the beginning of a run it is necessary to arrive at an initial set of signal timings that will subsequently be revised in the optimization process. The procedure used in SIGOP III produces a ''good'' set of timings for the network (i.e., including offsets).

a. Transform the network into a series of nodes separated by ''links'' whose ''length'' is proportional to the two-way volume on the ''link'' (The developers refer to these as arcs).

b. Using a technique developed by Kruskal determine the maximum ''path'' through the network. That is, construct a ''tree'' which includes all nodes and is the maximum ''length'' of all trees possible. Store the node sequence of this tree.

c. For the actual network, use Webster's equation to determine splits.

d. For each node in turn, construct a mini-network where the current node is the central node and it is connected to any adjacent nodes which have already been processed in this manner.

e. Treating the current central node of the mini-network, exercise the optimization procedure on the current mini-network to adjust the signal timing of the current (central) node to produce the minimum disutility in the mini-network.

f. Repeat steps c through e along the spanning tree determined at step b until all nodes have been treated.

This process produces a ''good'' initial timing since the spanning process emphasizes the heaviest traveled links. It thus reduces the number of iterations in the network optimization.

Step 2 - Calculate Traffic Performance and Disutility - The initial settings enter the traffic model and the performance measures and the disutility function are computed as discussed earlier. This process is repeated after each change of signal timing from the optimization process.

Step 3 - Gradient Search - Making use of the known aspects of the relationship between traffic operations and control, namely that platoons arriving primarily during the green will result in the lowest delay and stops. The developers of SIGOP III established predictable relationships between offsets and splits. The assumption is made first that the primary platoon always enters the link shortly after the beginning of green at the upstream intersection and the secondary platoon enters shortly after the onset of red. By projecting the platoons downstream, an ideal offset is easily determined for each link. A practical range of offsets is also readily calculated, since the length of the platoons (in seconds) is known. Within this relatively narrow range of alternative offsets (and splits on the major links), the optimization submodel does a gradient search over all possible values of offset and split to achieve the minimum network disutiity.

The last process is functionally similar to the hill-climb process used in TRANSYT; however, the preparation for entering the search is so highly developed by that point, the computer time required to conduct the search is greatly reduced.

The developers do caution, however, that there is no guarantee that the true global optimum will always be achieved. (Note: this is true of all ''optimization'' models with the possible exception of MAXBAND.)

OUTPUT REPORTS

There are three general types of outputs provided by SIGOP III, several of which are comprised of more than one table or plot. The major output are discussed separately below.

Input Data Report

The input to SIGOP III are reported back to the user in a series of formatted tables. A Link Data Input Report reflects the geometric and traffic data. Data in this report come from the Identification, Network and Link Cards.

Shorter reports indicate the inputs on the Minimum Phase Duration, Coupled Approaches, the various plot, Fixed offset and Split and Signal Timing Cards.

Optimal Signal Settings

The signal settings determined by SIGOP III (or input by the user if no optimization was to be performed) are output in the formatted table. The order of phase is not that to be implemented, but rather, phases should be implemented in the order I, II, III, and IV, as applicable. The affected links, offset and splits are output for each phase. This permits the relatively easy conversion to yield points after correcting for clearance and interval length.

Performance Analysis

This report shows the disutility value for each iteration of the model and reports the optimal. Next are given the performance values for each link and the network total. The MOE are:

- Volume (vph)
- Average Speed (mph)
- Delay (sec/veh)
- Stops (per minute)
- Capacity (vphg)
- Degree of Saturation (%)
- Maximum Queue (veh)
- Fuel Consumption (gal/hour)
- Total Emissions (lb/hu.)
  - Hydrocarbons
  - Carbon Monoxide
  - Nitric Oxide

Finally, the user specified time-space plots are issued.

Diagnostic Messages

SIGOP III performs extensive checks on the input data to identify obvious errors. During execution of the model other errors may be detected, such as excessive saturation. There are a total of 52 error messages in the library. Some of these also advise the user on a course of action, if applicable. In all cases, the messages, cause and corrective action required are well documented.

ADDITIONAL FEATURES

As already noted, SIGOP III can handle multiphase signals (up to four phases) and can automatically investigate the advantage of double cycling signals that have a low degree of saturation (thus extensive delay, stops and queue length).

SIGOP III can be used purely as an analysis tool to evaluate alternative timing plans derived from sources other than the SIGOP III optimization or to examine alternative patterns. Naturally, the user must code and run each alternative and evaluate the results manually.

Up to five runs may be executed per cycle length with no limit on the number of cycles optimized. This enables the user to investigate the effect of changing trend in traffic demand. Although limited to 80 signals and 230 links, the documentation describes how to expand the capacity of the program.

APPLICATIONS AND LIMITATIONS

SIGOP III is a powerful design and analysis tool for the engineer concerned with coordinated signal systems. Functionally, both SIGOP III and TRANSYT are quite similar, both with unique properties not available in the other. For example, inclusion of maximum queue length in the objective function is an important advantage in SIGOP III.

There are several items that would be considered as limitations in SIGOP III. These are listed below.

1. The limitation to four phases in the cycle cannot adequately serve some users. Up to six phases are not uncommon in many systems.
2. There is no provision for bus links in a SIGOP III analysis.

3. Permissive and unprotected turns are not addressed explicitly by SIGOP III. While this is true of other models, the user is often able to ''model'' such conditions by restricting the capacities of such movements. This is not possible in SIGOP III. However, permissive and unprotected turns are accounted for within the model.
4. The model does not explicitly deal with minor intersections (e.g., stop sign control).

In summary, SIGOP III has, as do all traffic models, several limitations and disadvantages. Nonetheless, the complexity of the optimization technique makes this model somewhat faster in terms of running time. The multiple cycle length capability is clearly an asset, which can save the designer a considerable amount of time that would ordinarily be spent in generating numerous jobs.

TW-SIGOP-III (MICROCOMPUTER VERSION OF SIGOP III)

The SIGOP III program is designed to generate fixed-time traffic signal timing patterns that minimize a specified form of motorist ''disutility.'' Disutility is directly expressed in terms of those Measures of Effectiveness of primary concern to the traffic engineer: vehicle delay, stops, and excessive queue length--all of which are computed by the program in the course of the optimization procedure.

TransWare has developed a microcomputer version of this program (TW-SIGOP-III).

TW-SIGOP-III requires 278K of RAM for the non-overlayed version and only 216K of RAM for the overlayed version exclusive of your operating system.

Bob Ellington of Federal Highway Administration's Office of Implementation has expressed his willingness to provide support for the program users.

SIGRID

PROGRAM DESCRIPTION - SIGRID is a grid signal timing optimization program. SIGRID optimizes signal offsets for specified conditions of cycle length, system geometry and phase splits, based on a ''desired'' difference in offsets between adjacent intersections. The ''desired'' differences in offsets can either be input directly by the traffic engineer or calculated by the program, based on distances between signals, assumed speeds and correction factors. Because of differing traffic volumes, travel times, turning movements and queue length, the ''desired'' differences in offsets will vary for each specified average traffic flow situation (32).

The system wide ''desired'' differences in offsets are rarely achievable in practice because of geometric conflicts. Discrepancies between the ''desired'' differences in offsets and the actual differences in offsets result in a tendency or proponsity to produce delay at signalized intersections. Other factors, notably marginal friction and overloading of the street system, contribute to the total delay encountered in the system. As these other factors are not controllable by the preprogrammed signal timing and are highly variable, they are used implicitly by the traffic engineer in determining the ''desired'' situation for the setting of the traffic signals.

The program determines the actual offset timing pattern which most closely approaches the ''desired'' differences in offsets. This is achieved by a mathematical procedure which minimizes the weighted differences between ''desired'' and actual differences in offsets, for each set of inputs. This approach relieves the traffic engineer of the repetitive and often inconclusive series of calculations and adjustments required for manual signal timing studies of networks, which allows him to use his professional judgment in specifying criteria for calculating and weighting the importance of the ''desired'' differences in offsets in determining optimum signal timing patterns.

The engineer also has the option of ''freezing'' the offset differences of certain links and having the program determine optimum timing of all other links in relation to the ''frozen'' links. If a set of ''frozen'' offsets produces conflicting demands, the program will make adjustments.

A special technique employed to lock various routes into any particular SIGRID run to produce combined band width and minimum delay solutions is also possible. This technique uses a combination of user specified link time differences, volume factors and importance weighting to produce the desired results. In general this procedure requires an iterative analysis using SIGART and TIMESPACE II initially and entering band data produced from that effort into the SIGRID program. The summary of time differences listed at the end of each TIMESPACE II report is used directly as inputs to SIGRID in this situation simplifying input preparation effort.

The output of the program includes a weighted average propensity to produce delay in the entire system. This weighted average propensity is a rating of system efficiency for each combination of offsets, phase splits and volume distribution studied.

INPUT - SIGRID data is loaded from card image inputs created by an editor or data input program. All numeric data is right justified.

The input for SIGRID include network descriptive data and performance target values. This program calculates the optimum signal offsets to minimize delay and produces delay propensity parameters that indicate the calculated difference between the existing network offset plan and the optimized offset plan.

The arrangement of the input files are as follow:

- TITLE CARD (card type A, 1 per data set) - It is used as identification of the street system.
- CYCLE CARD (card type b, 1 per cycle) - It is entered one cycle length per card. The program will produce a separate optimization for each different cycle card.
- A BLANK CARD IMAGE (a line of input with no data).
- SIGNAL DATA CARDS (card type C, 1 per signalized intersection) - This will identify the signal, gives the splits, and provides delay beginning of green.
- A BLANK CARD IMAGE (a line of input with no data).
- LINK DATA CARDS (card type D, 1 per link) - This card includes the upstream intersection of the link, downstream intersection of the link, the direction, volume, desired difference in offset, and Importance factor to favor certain street sections over other ones regardless of relative traffic volumes.
- A BLANK CARD IMAGE (a line of input with no data).
- PLATOON LENGTH FACTOR CARD (card type E, one required) - A system-wide value (in percent) that instructs the program to assume a specified percentage of green time utilization.
- A BLANK CARD IMAGE (a line of input with no data).

* NAMEPLOT CARD (card type F, 1 per plot) - This is used to identify each TIMESPACE diagram.
* SEQUENCE CARD (card type G, 1 per plot) - This is the sequence of intersections encountered when traveling on the street in question.

* The nameplot cards and the sequence cards must alternate with one of each card type for each arterial to be plotted.

OUTPUT - The output starts with pertinent network descriptive information. A data table is then produced which summarizes the program results. Link data is listed in the first three columns. An offset difference tabulation is then listed showing desired, original and new (selected) difference values. Delay propensity factors are then listed for both original and new conditions. Volume and link importance factors are then tabulated. The last element in the report is a weighted average delay propensity factor for the original system and for the new system.

The next portion of the output deals with the actual timespace diagrams. The top line includes the cycle length and the title of the plot. The horizontal and vertical scale of the plot are shown on the right. These scales are accurate for 14 inch wide paper. Four fields of information are shown along the left side of the plot. Under ''OFF'' is the offset information for each intersection. Under ''INT'' is a four character alphanumeric identifier of the intersection. Under ''FEET'' is the accumulated distance traveled down the diagram. Under ''DWN'' is the percent of green plus amber time available at that intersection while traveling down the page. The column on the far right of the page labelled ''UP'' is the percent of green plus amber time allocated at that intersection while moving up the page.

On the diagram itself there are several symbols to identify different signal situations. The ''='' sign indicates the traffic stops in both directions going up and down the page. The ''-'' sign indicates that traffic moving down the page is required to stop. This is used for traffic moving up the page using a protected left turn, etc. The ''+'' sign indicates that traffic moving up the page is required to stop.

The ''.'' sign is a visual aid used to delineate the cycle length on the horizontal scale and 1,000 foot increments on the vertical scale.

The bottom part of the output is a summary of time differences for beginning and end times of greens along the arterial. This information may be used during field installation to verify proper controller settings.

PROGRAM HIGHLIGHTS AND SHORTCOMINGS - A maximum of 40 signals may be included in a SIGRID run. Timespace diagrams may be plotted for a maximum of 20 routes. Each route may consist of up to 15 intersections. The routes may be configured by the Traffic Engineer to traverse any set of links in the system. Routes can be plotted around corners or along special roadway sections such as truck and/or bus routes.

The grid signal timing optimization program SIGRID uses a data set that has an identical format to TIMESPACE III. The consistencies of formats of the two programs was established to facilitate their combined use. In addition to the SIGRID input data file, TIMESPACE III uses as input data an offset file to produce the desired diagrams.

The utility program NAMELIST.COM can be used to genertate a file containing intersection names called NAME.DAT to be printed on the timespace diagrams.

The user has the option of ''freezing'' the offset differences of certain links and having the program determine optimum timing of all other links in relation to the ''frozen'' links. If a set of ''frozen'' offsets produces conflicting demands, the program will make adjustments.

Among the advantages are: 1) the program will produce a separate optimization for each different cycle length, 2) it is capable of favoring certain street sections over others regardless of relative traffic volumes, 3) timespace diagram takes into account the lead and lag green times, 4) output is neat and in a useful form.

Among the disadvantages are: 1) it requires an external editor, 2) timespace diagram does not give a bandwidth, 3) the user's manual needs some verifications, 4) a few terminologies need to be defined.

TRANSYT-7F

MODEL DESCRIPTION

The TRANSYT model is a macroscopic, deterministic, time scan optimization model. It is used for optimizing the signalization on arterials and grid networks. The program is written in FORTRAN IV for more universal use. The TRANSYT-7F model operates on an IBM 370, CDC 7700, VAX and HONEYWELL computers. On the IBM 370 the core requirements for TRANSYT-7F is 278k. The program contains 7650 lines of code with approximately 10 percent used for comments (33).

The physical characteristics of a system considered by TRANSYT-7F is a coordinated network of up to 50 intersections (nodes) with up to 250 directional links. Only signalized intersections are normally modeled, but facilities exist for modeling sign controlled intersections and ''bottleneck'' locations. Signal control is fixed-time, two to seven phase (including pedestrian movements) and fixed sequential phasing. Stoplines may be ''shared'' by several movements and priority lanes may be designated for buses.

Signal timings are printed in a format that is directly implemented in the field for pretimed controllers and time-space diagrams may be printed for selected routes.

INPUT REQUIREMENTS

There are 14 major types of input cards for TRANSYT-7F, some of which have single cards, others multiple cards. The basic inputs fall into four functional categories, namely, data which:

a. Are common to the entire network (e.g., cycle length)

b. Control the optimization process

c. Specify signal timing and

d. Specify traffic data

Input cards are numbered by card type and are input with all node-specific data grouped by intersection. Standard coding sheets are available to assist the user in preparing input cards.

OPERATIONAL SUMMARY

TRANSYT-7F is a macroscopic, deterministic optimization model with independent time scan. It has a moderately structured organization with a master program which calls other subroutines as the analysis progresses.

Input cards are read and checked for apparent accuracy and if errors are detected the erroneous card is printed out with the detected error underlined, and a message is printed. TRANSYT-7F may calculate initial splits if these were not supplied by the user. Thus after satisfactorily reading the input data and, if necessary, computing the initial splits, the program execution begins. The execution of TRANSYT-7F is controlled by the optimization model. The process of obtaining the optimum solution is called the ''Hill-Climbing'' procedure.

Hill-climbing is accomplished by varying off-sets and splits in small, medium or large steps and calculating the resulting traffic effects. To accomplish the latter, it is necessary to determine the behavior of traffic within a link. These are based on the manipulation of the following:

a. The ''IN'' pattern is the periodic flow rate of traffic that arrives at the stopline (downstream) if the traffic was not impeded by the signal.

b. The ''OUT'' pattern is the periodic traffic flow rate leaving a link.

c. The ''GO'' pattern is the periodic traffic flow rate that leaves the stopline if there was enough traffic to saturate the green.

The word ''pattern'' refers to the fact that TRANSYT-7F does not deal with individual vehicles, but rather platoons in histogram form.

The inflows of one link are obtained from the outflows of the upstreams link(s). These flow characteristics are computed for each link for each iteration and the delays are calculated, as discussed in the next section.

With this background, the full process may now be described. The first step is to calculate the performance index (PI) for the initial timings. Then the offset of one signal is altered by the number of time units (steps) input on Card 4 and recalculate the PI. If the PI is reduced, the offset is changed successively in the same direction until a minimum PI is reached. If the first alteration increased the PI, the search is made in the opposite ''direction.''

Each signal is adjusted in a similar manner in the order specified on Card Type 2 until the network minimum PI is reached. This process is repeated for each hill-climb value on Card Type 4 (or in the default list). This is the offset optimization.

TRANSYT-7F also optimizes splits. It does this by altering the start of each phase and recalculating the PI as before.

It is obvious that the length of a run will be largely dependent on how many iterations of the model are required. Another factor is that if the number of steps used to alter the particular timing is too small, the solution may be ''trapped'' into a local optimum which is not global.

If a ''quick optimization'' is desired, the user may specify a hill-climb sequence which may not result in the ''best'' PI, but will be reasonably good. Another option allows optimization to include only those links directly connected to the present link, rather than the entire network.

COMPUTATIONAL ALGORITHMS

The major algorithms in TRANSYT-7F are the objective function and the calculations of traffic characteristics. The objective function is called the ''performance index,'' or ''PI,'' and it is defined as follows:

$$\text{Minimize PI} = \sum_{i}^{n} [w(D)_i \; d_i + Kw(S)_i \; s_i]$$

where: d_i = delay on the ith link of network (veh-hr/hr)

s_i = average number of stops per second on link i,

k = the weighting factor for stops entered on card type 1, and

w = weighting factors for delay (D) and stops (S) for link i.

This objective function is minimized by an iterative search procedure where the signal timings are changed and the resulting flow and travel characteristics are recalculated.

The link patterns discussed in a previous section are found as follows, for the ith link at time step t:

$$IN_{iT} = \sum_{j} F_{ij} \; (P_{ij} * OUT_{jT})$$

where: F_{ij} = the smoothing process from link j to i

P_{ij} = the proportion of OUTj which feeds link i, and

OUT_{jT} = the OUT patterns of link $_{jT}$ at time t.

The number of vehicles (m_t) held at the stopline during time interval t is found by:

$$m_T = \mathrm{Max}[(m_{t-1} + q_T - S_T) \text{ or } (0)]$$

where: q_t = the number of vehicles arriving in interval t, given by the IN pattern, and

s_t = the number of vehicles allowed to leave in interval t, given by the OUT patterns.

The number of vehicles leaving in interval t is:

$$m_{t-1} + q_t - m_T$$

and these figures are used to derive the OUT pattern.

The average delay is calculated in two parts which are added together. The first is the average queue lengthover the cycle (times the cycle length) and the second is the delay due to random variations of arrivals and saturation. The second component for each link is found by,

$$d_{rs} = \left[\left(\frac{B_n}{B_d}\right)^2 + \frac{X^2}{B_d}\right]^{1/2} - \frac{B_n}{B_d}$$

where: d_{rs} = random and saturation delay

B_n = 2(1-X) = ZX

B_d = $4Z - Z^2$

Z = (2x/v*60/T)

X = degree of saturation

v = volume on the link and

T = simulation time

Since TRANSYT-7F assumes that traffic disperses as it travels downstream, the smoothing function (F) used in above equation is used to more realistically represent this dispersion of vehicles. F is calculated by,

$$F = \frac{1}{1+\alpha\beta t}$$

where: α = smoothing parameter (usually assumed to be 0.35 but it may be varied), and

β = a coefficient which ''shifts'' the effective travel time (set to 0.8), and

t = link travel time

The number of stops is simply equal to the number of vehicles delayed. Since some delays may only be slow downs and not full stops, the calculation of stops may be adjusted by entering the appropriate parameters on card type 5. The recommended values found to be valid in England are as follows:

Seconds of Delay:	1	2	3	4	5	6	7	8	9	10	> 10
% of Stops:	20	50	65	76	83	88	93	95	97	99	100

OUTPUT REPORTS

There are five basic outputs available from a successful TRANSYT-7F run (i.e., no errors detected).

Input Data Report

The input data are echoed in essentially the same format they were input, with column headings to identify each data item.

Traffic Performance Tables

Traffic performance estimates are produced for each set of timings, normally initial and/or final. The initial and final tables have identical format but are labeled differently. Below the title, the link data are given, along with several measures of effectiveness (MOE) and green periods (in seconds). The link MOE's are subtotaled by node to enable rapid identification of critical intersections. Starred (*) links are bus links. The ''system'' MOE's are shown at the bottom of the table.

Flow Profile Plots (Optional)

Figure 13 shows a typical flow pattern plot produced by TRANSYT 7F. The flow patterns use symbols to enable the user to ''see'' what is happening over the signal cycle. The following symbols are used:

a. Flow that queues at the stopline, normally on red (I)

b. Flow leaving the stopline on green which clears the queue (S)

c. Arrivals on green that may or may not be delayed, as explained below (O)

The symbol (S) represents queue discharge and is generally at the saturation flow rate. The symbol (O) represents arrivals and when the ''O's'' appear

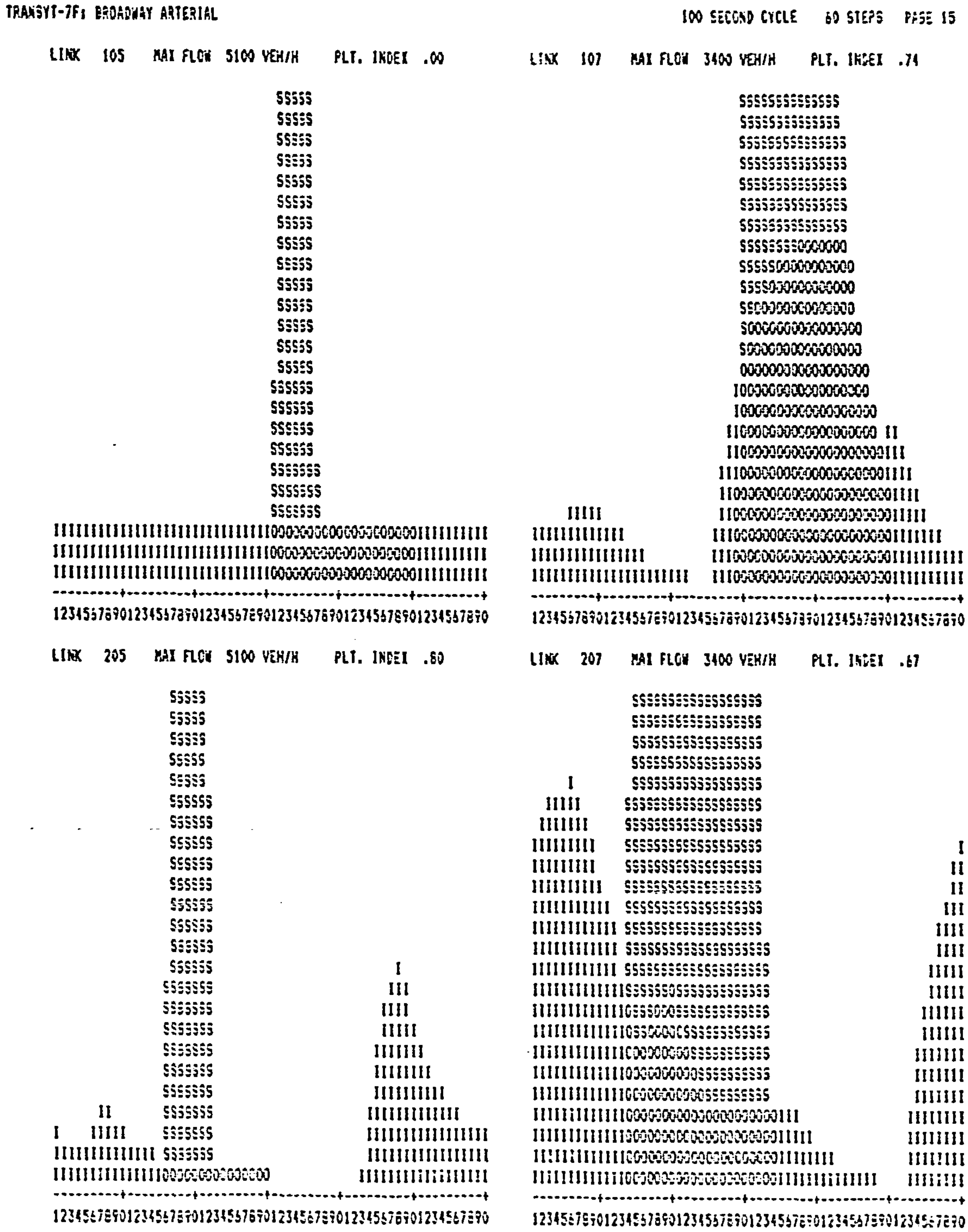

Figure 13 Flow Pattern of TRANSYT 7F

below the (S), it indicates those vehicles which join the back of the queue. When the ''O's'' appear without the ''S's'' above them, these are undelayed arrivals/departures.

The flows are overlayed so the distortion caused by red/green periods are easily observed. The horizontal scale is always constant and equal to the cycle length in steps. The vertical scale is always flow rate, but the scale depends on the maximum flow. The saturation flow always extends to the top of the respective plot (i.e., 24 lines).

These plots are intended to be used to verify field conditions by merely observing whether the intersection approaches actually perform as predicted.

Signal Timing Tables

TRANSYT-7F produces a unique output of signal settings. For pretimed controllers, these timings may be readily implemented in the field with no further manual manipulation, so long as the offsets do not fall within a clearance (or on another pin in the case of electro-mechanical controllers). Warning are issued in the event of either of these conflicts.

Time-Space Diagrams

TRANSYT-7F will print a time-space diagram for any selected route of up to 50 nodes. The route need not be linear, and many plots may be printed. The symbols of the diagram are as follows:

a. ''+'' green in the direction of increasing distance from the origin (down the page)

b. ''-'' green in the direction of decreasing distance (up the page)

c. ''blank'' green on the route in both directions

d. ''*'' red on the route

Although through bands are not explicitly plotted, the scaling allows convenient use of such tools as triangles and protractors to plot the bands.

ADDITIONAL FEATURES

TRANSYT-7F has a number of options, most of which are handled by control cards. It has already been noted that buses can be modeled separately by including bus links. These can either be separate lanes or shared lanes. In addition, pedestrians can be modeled by treating them as ''vehicles'' on separate links. Care must be taken to insure that pedestrians do not interchange with vehicles in the flow patterns. Pedestrian links should have zero stops penalty and delay weights if it is desired to exclude them from the PI and fuel calculations.

TRANSYT-7F can be used to design larger networks by subdividing the networks into sections that can be handled by the present program (i.e, 50 nodes and 250 links). The boundary nodes are fixed from section to section so that their timings are not changed in the subsequent analysis. In this manner, sections can be ''stacked'' such that they will always share one or more nodes whose timings will be optimized in one section then remain fixed in the subsequent section.

Additionally, bottlenecks and unsignalized intersections can be considered. At intersections governed by a fixed priority rule (e.g. stop sign on cross-street) the main route traffic incurs no delay. The inflow from the side road is given a ''GO'' pattern proportionate to its actual capacity which is a function of the main street traffic.

While TRANSYT-7F is the most current version of TRANSYT readily available in the U.S., TRRL has also written version 8 which improves upon the current version and is only available on a license basis.

APPLICATIONS AND LIMITATIONS

In addition to designing the optimal signalization of coordinated networks, TRANSYT-7F can analyze existing (or any preset) conditions by simply inputting card type 51 (Run Card).

TRANSYT-7F does not explicitly optimize the cycle length or phase sequences; however, these can be ''optimized'' by multiple runs with varying values of the cycle length input in Card Type 1 or phase sequences on Card type 2X. A manual approach similar to the hill climb technique explained earlier should be used (probably with the ''quick optimization'' procedure used in the initial trials and the normal optimization used for ''fine tuning'').

The shortcomings listed in the above paragraphs are clearly limitations present in this version; however, TRANSYT-7F is sufficiently realistic to design many network configurations, and can be extremely useful to the local traffic agency.

Other limiting assumptions are listed below:

a. Large amounts of data are needed.

b. All major intersections in the network are assumed to have traffic signals, although sign-controlled intersections and other mid-block bottlenecks can be modeled.

c. Traffic entering the network from the outside does so at a constant uniform rate on each approach. This is not unrealistic over a long period such as an hour.

d. The volumes and proportions of turns remain constant at each approach for the entire period of analysis.

e. Traffic dispersion is assumed to be uniform for the period of analysis.

f. Only pretimed signal controllers can be simulated and no capability of evaluating actuated signals.

The last three are probably the most serious limitations; although the platoon dispersion model is far more realistic than a simpler assumption of uniform platoons.

TRANSYT-7F (MICROCOMPUTER VERSION)

PROGRAM DESCRIPTION - This disk contains the program TRANSYT-7F Release 4, version 4, from FHWA. The executable program on this disk has been linked with the 8087 math coprocessor. The execution time for an optimization run with the 8087 chip is approximately 5 minutes per node (33).

Also, included on this disk is an executable copy of the Platoon Progression Diagram (PPD) program. When TRANSYT-7F is run with the PPD flag on (card type 60), it will produce a data file to be used with the PPD program. The PPD program will read this data file to produce Platoon Progression Diagrams, plots which show the density of platoon by dark to lighter shading on a time-space diagram.

The minimum hardware requirements are as follows:

- at least 256K RAM
- two 360K floppy disk drives, or one floppy and one hard disk
- DOS 2.0, 2.1, 3.0, or 3.1 operating system
- 132 column printer (Epson 80 or 132 column printer for PPD)
- 8087 or 80287 math coprocessor chip

The execution speed of TRANSYT-7F run is directly proportional to the size of the network. Therefore, if running a large network optimization or cycle evaluation, it may appear that nothing is happening.

REFERENCES

1. Roger Creighton and Associates, ''CAPCALC User's Manual,'' Roger Creighton and Associates, Delmar, New York.

2. Roger Creighton and Associates, ''CAPCALC 85 User's Manual,'' Roger Creighton and Associates, Delmar, New York.

3. MGA Associates, ''Comprehensive Analysis Program for a Single Signalized Intersection,'' Operator's Manual, MGA Associates, La Habra, CA.

4. Bather, Belrose, and Boje, Inc., ''CMA User's Manual,'' Bather, Belrose, and Boje, Inc., Minneapolis, Minnesota.

5. University of Florida Transportation Research Center, ''CMA/M User's Manual,'' McTrans Package, University of Florida, Gainesville, Florida, January 1983.

6. TRANSTEK, ''EZ-POSIT II.2 User's Manual,'' TRANSTEK, Transportation Center, University of Kansas, Lawrence, Kansas.

7. Bather, Belrose, and Boje, Inc., ''ICAP User's Manual,'' Bather, Belrose, and Boje, Inc., Minneapolis, Minnesota.

8. Bather, Belrose, and Boje, Inc., ''INTERCALC User's Manual,'' Bather, Belrose, and Boje, Inc., Minneapolis, Minnesota.

9. De Shazo, Starek Tang, Inc., ''Signalized Capacity Analysis User's Manual,'' De Shazo, Starek Tang, Inc., 330 Union Station, Dallas, Texas, 75202, 1986.

10. TRANSWARE, ''FREESIAP and SIAP User's Manual,'' TRANSWARE, P.O. Box 6896, Mclean, Virginia.

11. Binghamton Metropolitan Transportation Study, ''SICA User's Manual,'' BMTS, P.O. Box 1766, Binghamton, New York 13902.

12. Barton-Aschman Associates, ''SIGNAL User's Manual,'' Barton-Aschman Associates, Evanston, Illinois.

13. Barton-Aschman Associates, ''TED TEAPAC EDITOR,'' Barton-Aschman Associates, Evanston, Illinois.

14. Federal Highway Administration, ''SOAP 84 User's Manual,'' FHWA, Washington, D.C. 20590, January 1985.

15. University of Florida Transportation Research Center, ''SOAP/M User's Manual,'' McTrans Package, University of Florida, Gainesville, Florida, January, 1983.

16. University of Texas Transportation Research Center, ''TEXAS User's Manual,'' University of Texas, Austin, Texas, 1984.

17. TRANSTEK, ''EZ-PASSER II.X1 User's Guide,'' TRANSTEK, Lawrence, Kansas 66046.

18. Barton-Aschman Associates, ''NOSTOP User's Manual,'' Barton-Aschman Associates, Evanston, Illinois.

19. Bather, Belrose, and Boje, Inc., ''PASSER II-80 User's Manual,'' Bather, Belrose, and Boje, Inc., Minneapolis, Minnesota.

20. TEXAS Transportation Institute, ''Analysis of Reduced-Delay Optimization and Other Enhancement to PASSER II-80--PASSER II-84--FINAL REPORT,'' State Department of Highway and Public Transportation, Austin, Texas.

21. Bather, Belrose, and Boje, Inc., ''SIGART and TIMESPACE II User's Manual,'' Bather, Belrose, and Boje, Inc., Minneapolis, Minnesota.

22. University of Florida Transportation Research Center, ''SPAN User's Manual,'' McTrans Package, University of Florida, Gainesville, Florida, January 1983.

23. DKS Associates, ''TIMDIS User's Manual,'' Version 2.01, DKS Associates, Oakland, California, May, 1984.

24. Federal Highway Administration, ''Optimal Signal Timing for Arterial Signal Systems, Vol. 2. MAXBAND User's Manual,'' FHWA, Washington, D.C. 20590, December 1980.

25. Texas Transportation Institute, State Department of Highways and Public Transportation, ''A Report on the User's Manual for Diamond Interchange Singalization--PASSER III,'' State Department of Highway and Public Transportation, P.O. Box 5051, Austin, Texas 78763, August 1977.

26. TransWare, ''EIBAND,'' TransWare, McLean, Virginia.

27. TRANSTEK, ''EZ-TRANSYT User's Manual,'' TRANSTEK, Transporation Center, University of Kansas, Lawrence, Kansas.

28. Federal Highway Administration, ''NETSIM User's Manual,'' FHWA, Washington, D.C. 20590, January 1980.

29. Federal Highway Administration, ''Platoon Progression Diagram (PPD) for the IBM-PC/XT,'' Systems and Software Support Team, Office of Traffic Operations, FHWA, Washington, D.C., August 1985.

30. Barton-Aschman Associates, ''PRETRANSYT User's Manual,'' Barton-Aschman Associates, Evanston, Illinois.

31. Federal Highway Administration, ''SIGOP-III User's Manual,'' FHWA, Washington, D.C. 20590, July 1983.

32. Bather, Belrose, and Boje, Inc., ''SIGRID II User's Manual,'' Bather, Belrose, and Boje, Inc., Minneapolis, Minnesota.

33. Federal Highway Administration, ''TRANSYT-7F User's Manual,'' FHWA, Washington, D.C. 20590, February 1983.