A Computer Graphics Analysis of a Freeway Merge Control System

1976

Thomas Alan Risher
University of Central Florida

Find similar works at: https://stars.library.ucf.edu/rtd

University of Central Florida Libraries http://library.ucf.edu

Part of the Transportation Engineering Commons

STARS Citation
https://stars.library.ucf.edu/rtd/251

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.
A COMPUTER GRAPHICS ANALYSIS OF A FREEWAY MERGE CONTROL SYSTEM

BY

THOMAS ALAN RISHER
B.S.E., Florida Technological University, 1975

RESEARCH REPORT
Submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering in the Graduate Studies Program of the College of Engineering Florida Technological University

Orlando, Florida
1976
ABSTRACT

In 1975, C. S. Bauer completed a doctoral dissertation at the University of Florida which treated the Green Band Merging Control System on I-75 in Tampa, Florida. In this work, Bauer suggested the possibility for the use of computer graphics as a tool for analysis of the bands generated by the Green Band Control System Simulation developed in his dissertation. The use of computer generated movies of the bands displayed to ramp drivers by the system allows the comparison of various band control strategies without the need for field implementation and testing. With the goal of producing such films in mind, the research topic discussed in this paper was undertaken.

The report introduces the reader to some of the basic aspects of computer graphics and presents specialized computer software and interface hardware for producing automated computer graphics movies from a Tektronix 4010 storage display.

A brief discussion of the Tampa System and its associated simulation program is presented, and representative frames from the movies of the Tampa System produced in the research are discussed. Suggestions for additional
work that could be undertaken in this research area con-
clude the report.

Christen S. Bauer
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF ILLUSTRATIONS</td>
<td>iv</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>Chapter</td>
<td></td>
</tr>
<tr>
<td>I  COMPUTER GRAPHICS</td>
<td>4</td>
</tr>
<tr>
<td>II THE TAMPA GREEN BAND MERGE CONTROL SYSTEM</td>
<td>10</td>
</tr>
<tr>
<td>III PRODUCTION OF THE COMPUTER GENERATED GREEN BAND MOVIES</td>
<td>18</td>
</tr>
<tr>
<td>IV CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH</td>
<td>45</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>52</td>
</tr>
<tr>
<td>LIST OF REFERENCES</td>
<td>85</td>
</tr>
</tbody>
</table>
LIST OF ILLUSTRATIONS

1. Locations of Sensors Used by the On-line Control System .................. 13
2. Subroutine Structure for Green Band Control Program Simulator .............. 17
3. The Computer Generated Green Band Movie Process .......................... 19
4. Flow Chart of the XVEH Subroutine ........................................ 22
5. Flow Chart of the FRMOUT Subroutine ....................................... 29
6. Flow Chart of the Drawing Program ........................................... 31
7. Equipment Set Up for Computer Generated Green Band Movie Production ........ 38
8. Block Diagram of the ASRM .................................................... 40
9. The Automatic Shutter Release Mechanism .................................... 41
10. Solenoid and Microswitch of the ASRM ....................................... 43
11. Frame 626 of a Green Band Movie ........................................... 46
12. Frame 642 of a Green Band Movie ........................................... 47
13. Frame 643 of a Green Band Movie ........................................... 48
INTRODUCTION

The digital computer has proven to be a great aid in the solution of computational problems which arise in the field of Engineering. Its ability to process information at a much higher rate, than is possible for human interpretation, calls for methods to be developed which will increase the speed of information transfer and allow dynamic interaction between the operator and the computer. With the use of computer graphics, many pages of conventional numerical output, which could require hours to interpret, can be converted into graphical presentations. This transformation can often reduce human interpretation time to minutes or even seconds.

Computer graphics can aid the simulation analyst when the graphics are presented on a real-time basis showing the progress of the system being simulated. Filming of the simulation graphic output can provide a record of performance of the simulator and can be used as a basis of comparison when changes are made within the simulation program. The films can then be analyzed without repeated running of the program and consequent use of computer time.

Computer graphics can be particularly helpful in
the interpretation of simulation outputs in Traffic Engineering (Sandys and Schlaefli 1969). Traffic flows can be seen and flow changes, due to alterations in control sequence or traffic density, can be viewed as they would occur in the real system. The viewing of the simulation output at the real system speed allows the analyst to gain a better understanding of the system and its response to input or control changes.

A problem of particular interest to the Traffic Engineer is the merging of ramp vehicles into the flow of freeway traffic. Freeway Merge Control Systems are designed to assist the ramp vehicle operator in his merging task. A smooth joining of the ramp vehicle traffic increases ramp flow rates and decreases the introduction of disturbances into the freeway traffic flow. In order to function properly and efficiently, the merge system must be both easy to understand and reliable. The ramp vehicle driver must have confidence in the system or he will not use it to its fullest potential. Any inconsistency in the merge system field display will cause distrust and consequent non-use. This is the problem confronted by the Tampa Green Band Control System presently in use at the Ashley Street entrance to Interstate 75 in Tampa, Florida (Bauer 1975).

The control logic for the band display causes instability of the bands used to indicate acceptable gaps in
freeway traffic into which a ramp vehicle could merge. Changes in control logic of the actual system would involve many hours of editing and system down time. To provide a tool for studying a wide variety of operating options, a simulation of the merge system was written (Bauer 1975). The addition of a graphic display and an associated technique for filming such displays are the subjects of this report.
CHAPTER I

COMPUTER GRAPHICS

Of man's senses, sight is the most important in the transfer of information concerning the world around him (McCormick 1970). Much of the visual information man receives is alphanumeric in nature (McCormick 1970). In engineering, the alphanumeric information presented to man is most likely a coded representation of some physical characteristic of a system or component, such as the length of a structural member or speed of a vehicle. The transfer of the information requires the engineer to decode the alphanumeric data to a form in which he can visualize the relationship of the structural member to the entire structure or the speed of the vehicle to other variables within the system. This decoding involves the use of time and introduces the possibility of errors in the engineers perception of the system. Computer graphics provides a means of assisting the decoding process, resulting in a time savings and a possible reduction in the probability of error.

Computer graphics can be thought of as the merging of man's most ancient means of communication, graphics; and his most modern tool for design and analysis, the
digital computer. It involves the definition, storage, manipulation, interrogation, and presentation of graphical data by the computer. Passive computer graphics allows no direct operator control of the pictorial output. Graphic plots on a line printer are examples of passive computer graphics. In contrast, however, interactive computer graphics provides an opportunity for real time operator interaction with the pictorial output and can involve both graphical input as well as output (Rodgers and Adams 1976).

The heart of many computer graphics systems is the cathode ray tube (CRT). The CRT provides pictorial or visual information from electrical signals. It is not within the scope of this paper to describe the principles involved in the functioning of a CRT, but a brief discussion of some of the operating characteristics would be relevant. Basically there are three types of CRTs used in computer graphics: the raster scan CRT, the refresh CRT, and the storage tube CRT (Rodgers and Adams 1976).

The raster scan CRT functions on the same principles as a television. The picture is composed of a series of dots which are scanned horizontally in two rasters. As in television systems, the purpose of using two rasters is to reduce flickering of the picture. The use of the raster scan CRT necessitates conversion of line and character information to a form compatible with the raster scan technique (Rodgers and Adams 1976). Scan conversion is a
difficult process and the storage and manipulation of the data after conversion presents further difficulties. The manipulation of the data to change a line position is difficult due to the storing of the points representing the line throughout the data (Newman and Sproull 1973).

A refresh CRT differs from the raster scan CRT in that the lines and points are drawn as lines and points and not as points in a horizontal scanning of the entire tube face. As the name implies, the refresh CRT requires refreshing of the display due to the short persistence time of the phosphors used on the CRT face. The refreshing rate is limited by the ability of the supportive hardware to process the instructions involved in presenting the picture. As the number of lines comprising a picture increases the processing of the instruction list for displaying the picture requires more time, and eventually the picture will begin to flicker since the refreshing rate has dropped below the human perceptive rate which is approximately 25 images per second. However, there is an advantage in the method used to represent a line in the refresh CRT. Since each line is stored as a drawing instruction, the manipulation of the individual lines involves the modification of one or more instructions and not a large number of individual data points representing dots as in the raster scan CRT (Rodgers and Adams 1976).

The storage tube CRT, in contrast to the other
7

types of CRTs, uses a phosphor with a long persistence. Once a line has been displayed it will remain unless erased by an electrical signal which erases the entire display. The erasing technique restricts the display of dynamic movement of the picture components as it requires the complete redrawing of the display when any modifications to the picture are to be made (Rodgers and Adams 1976). The erasing process is also distracting in that a flash is generated on the face of the CRT (McCormick 1970). Since no refreshing is required, the storage tube CRT is flicker free. Elimination of refreshing hardware along with the use of conventional storage techniques for the picture data allows the storage tube CRT graphics terminal to be of relatively low cost when compared to the two systems discussed previously (Rodgers and Adams 1976).

The CRTs described can be used to construct a wide variety of computer graphics systems. A storage tube CRT can provide the basis for a low cost system with little or no interactive capability; while a refresh CRT could be used in an expensive system to provide dynamic interactive capabilities. The supportive computer graphics software also ranges widely in complexity, from simple point and line plotting routines to very complex routines involving three dimensions, perspectives, and hidden line elimination. The interested reader can find example mathematical techniques and algorithms in Newman and Sproull
(1973) and Rodgers and Adams (1976).

The literature provides many examples of the application of computer graphics in the transportation area. Sandys and Schlaefle (1969) present an interactive computer graphics system which provides traffic parameters, such as vehicle delay and traffic volume, generated by a simulation of a portion of the San Francisco, California urban street network. They also developed a transportation oriented computer language to aid the operator in interacting with the system.

Computer graphics was utilized by Courage (1973) to aid in the analysis of network signal progression. The system developed relieves the traffic engineer of the cumbersome job of manually integrating signal system timing diagrams and allows him to better understand the signal timing plans he is developing.

Transportation Research News (1975) published an article describing an interesting application of computer graphics. The graphics terminal displays the output of a program used to predict the behavior of an automobile traversing irregular terrain or colliding with a roadside obstacles. A three dimensional picture is presented as the vehicle accomplishes its maneuvers. The aim of the system is to eventually reduce the injuries and damages incurred in single vehicle accidents. The program was developed by Calspan Corporation of Buffalo, New York, under the
direction of Raymond R. McHenry.

Rogers and Bruce (1976) give an example of interactive computer graphics use in airport master planning. The method developed relates four basic types of airport system variables. These variables are aircraft demand and pattern, air traffic control standards, runway configuration, and average delay criterion. The system allows the analyst to designate three of the variables as independent with the other becoming the dependent variable. The equations relating these variables are then solved, and a graphic presentation of the results is presented to the analyst.

The future of computer graphics use in transportation analysis looks bright. Ruiter and Sussman (1971) describe an imaginary General Agency for Transportation Studies in which they visualize extensive use of computer graphics in the analysis of transportation systems. A 1974 report on a seminar held at the Battelle Seattle Research Center, Seattle, Washington, predicts growth in the use of computer graphics due to trends in the reduction of computer graphics equipment costs, increased use in secondary schools as well as universities, availability of micro computers for special purpose uses, and the development of large bases for access by graphics terminals (Schneider 1974).
CHAPTER II

THE TAMPA GREEN BAND MERGE CONTROL SYSTEM

The intended purpose of the Tampa Green Band Control System is to aid the driver of an entrance ramp vehicle in the merging task. Entrance ramps to freeways have been found to have a high degree of congestion and accident rates (Bauer 1975). Contributing factors to the high rate of accidents include:

1. Poor visibility due to bad geometric designs or environmental conditions.
2. The necessity for the ramp driver to monitor a number of moving vehicles on the freeway (mostly upstream relative to his position) and other ramp vehicles moving ahead of him, thus dividing his attention in two opposing directions.
3. The ramp driver has difficulty in accurately estimating the speeds of vehicles on the freeway.
4. The ramp driver must also monitor conditions in any queue of vehicles which may be stopped in the merge area waiting for gaps in the stream of freeway traffic.
5. The existence of short acceleration lanes at many existing freeway entrance ramps significantly increases the number of accidents on these ramps, as acceleration lanes of less than 200-foot length have been shown to have an accident rate approximately 60% higher than those with a length of 700 feet or greater (Bauer 1975).

The Ashley Street entrance to I-75 in Tampa,
Florida, is an example of a ramp having both a design geometry which does not meet current standards and a short acceleration lane. The geometry interferes with the ramp drivers view of the upstream freeway traffic, and the acceleration lane is only 414 feet long. The freeway and merge area are elevated and reconstruction to correct the bad geometry is limited by expense and land space. The only feasible alternative to reconstruction is to provide a ramp merging control system. The Green Band Control System gives the driver of the ramp vehicle information on gap availability in the freeway traffic, the required speed to enter a gap, and his relative positioning within the gap (Bauer 1975).

The system utilizes a Raytheon Model 704 Computer to process data obtained by various sensors implanted in the freeway and ramp. The sensors are inductive-loop type and are positioned in grooves cut in the road surface and covered with a heavy-duty sealant.

Four modes of operation are available to the Control System: (1) initialization mode, (2) stopped-metering mode, (3) stopped-gap acceptance mode, and (4) moving merge mode. The initialization mode, as the name implies, allows initialization of the system. Measurements of freeway velocity and volume are taken on-line in a continuous sampling process to determine the mode in which the system will operate. The stopped-metering mode releases
ramp vehicles from a queue on the ramp at predetermined fixed-time intervals with no green bands displayed and is the mode used for the highest freeway traffic density. Stopped-gap acceptance mode also releases ramp vehicles from a queue but only when an acceptable gap has been found. The green band display is utilized in this mode. The moving merge mode does not force queuing of ramp vehicles, but rather attempts to guide them into gaps in the freeway traffic. Moving merge mode is used for the lowest traffic densities when ramp queuing would not normally be expected to occur.

Figure 1 illustrates the locations of the sensors along the freeway and entrance ramp. The functions of the sensors as described by Bauer (1975) are:

1. Double-loop sensors F7 through F1 supply information used to establish freeway vehicle trajectories upstream of the merge area. Sensor F1 is used to generate three-minute freeway velocity averages for use in the system mode control logic.

2. Single-loop sensor R12 was originally installed to provide a capability for ramp queue detection in the stopped modes of system operation. This function was later assigned to single-loop sensor R8 to allow a more rapid system response capability to the onset of ramp congestion.

3. Single loops denoted by R6B and R6A, respectively, serve as vehicle check-in and check-out detectors at the ramp traffic signal location in the stopped modes of system operation.

4. Single-loop sensor R3 is used to control the operation of a blank-out YIELD sign located at the end of the ramp display.
Fig. 1. Locations of sensors used by the on-line control system.
This sign is illuminated if a ramp vehicle crosses R3 without an accompanying green band on the ramp display opposite this location.

5. Single-loop sensor FOA, which is the first loop of the double-loop freeway velocity sensor FOA/B, is used to generate continuous three-minute counts of freeway and ramp merging volumes downstream of the acceleration lane. This information is used by the system mode control logic in the Tampa software.

6. Acceleration lane presence sensors M1 through M8 are used to inhibit green band activity when occupied for a time greater than a preset threshold value.

Other physical components of the system include hardwired communication lines which were specially installed to carry the sensor data to the computer. The green band display units consist of 4 foot green fluorescent tube bulbs housed in aluminum channels. The bulbs are used to make up the 608 foot display.

Of the four system operating modes, the one of special interest to this report is the moving merge mode. Under this mode, the computer uses data from the freeway sensors to estimate the time of arrival (ETA) of the freeway vehicle at the merge point. The merge point is a point used by the computer as an expected point of merging of the ramp vehicle into the freeway traffic. A study was conducted by Bauer (1975) to arrive at a value of 204 feet past the nose of the ramp.

The computer then uses the ETAs to compute the gaps in the freeway traffic and display adequate gaps on
the green band display. Descriptions of the algorithms used may be found in Bauer (1975). The process is complex due to the geometry of the ramp and safety considerations which require that the bands must travel at 30 miles per hour through the initial section of the ramp and then accelerate to the nominal 45 miles per hour merge speed (Bauer 1975).

An operational problem with band instability exists in the moving merge mode. The variation of a vehicle ETA prediction across the series of sensors can cause the edges of the displayed band to move erratically along the display. The specific direction of this undesirable band movement depends on the type of band edge, leading or trailing, and the direction of the observed ETA change. Bias in the sensors themselves or an effect from the topography of the freeway can contribute to the band edge instability. The computer uses the activation times of the sensors to compute vehicle velocities and in turn ETA estimates. Thus, if there are differences in the sensor sensitivities, the estimated times of arrival will vary. Also, the freeway is not straight and level in the control area, and the acceleration and deceleration of vehicles on the slight hills may add to the band edge instability problem. This instability can significantly reduce user confidence in the system. To study this problem, a simulation model of the on-line system was developed. The
use of simulation is supported by several factors. Constant changing of the control logic for experimental reasons is very time consuming and could result in the display of bands which are not safe for use by the ramp driver; therefore, it would be necessary to close the ramp to prevent possible accidents. Also, the use of simulation allows different concepts for changes in the control logic to be compared with fixed data input (Bauer 1975).

The input data for the simulation are generated by a special program, TOGAP, which analyzes recorded data from the on-line system (Bauer 1975). The data provided include: (1) the ETA at the merge point, (2) the time at the vehicles crossing of the leading edge of the second loop of the sensor, (3) the vehicle velocity, (4) the vehicle length in seconds, and (5) the vehicle number from a tracking algorithm not used on the on-line system.

The simulation program models the actions of the on-line system in generating the green band displays. Figure 2 shows the subroutine structure of the simulation. The figure includes the two subroutines, XVEH and FRMOUT, which were added to the simulation during the completion of the project. Descriptions of the original routines may be found in Bauer (1975), while the two new subroutines are described in this report.
Fig. 2. Subroutine structure for Green Band Control Program Simulator.
CHAPTER III

PRODUCTION OF THE COMPUTER GENERATED GREEN BAND MOVIES

The work described in this report can be divided into three relatively independent tasks. The first task involved changing the simulation model to obtain the data required to draw a movie frame. The work involved making changes in the existing simulation code and the addition of new subroutines.

The second task was to develop a program to process the frame data and draw the pictures on the graphics terminal. Since a time-sharing terminal was to be used, the program execution had to be controlled by the operator to allow frame selection and photographing.

The third task was the execution of the drawing program and the photographing of the movie frames. Since the purpose of the project was to allow the comparison of green band control techniques, it was necessary to make several movies using the various band control techniques.

Figure 3 is a diagram of the process involved in the making of a computer generated Green Band Control Program Simulation movie. The processed sensor data serves as input to the green band simulation model which
Fig. 3. The computer generated green band movie process.
creates the frame data. The drawing program then uses the frame data as input to draw each frame of the movie on the graphics terminal, where it is photographed on a single frame basis using a 16 millimeter movie camera. The film is developed and the movie process is complete. The movies are then available for analyzing the effects of changes in the Green Band Simulation model.

The remainder of the chapter provides detailed description of the work accomplished in the completion of the tasks.

Generation of the Frame Data

The data required to draw a picture of the state of the freeway and the band display include the positions and types of the vehicles along the freeway, the status of the green band display, and the current simulation time. To acquire the data at a time interval corresponding to the frame exposure rate of the film required modification of the existing Green Band Simulation program.

The modification involved the addition of an interrupt to the simulation main program, changes in the GBUD subroutine, and the addition of two subroutines.

The interruption of the simulation to allow the output of the current state was provided by a cycle counter and the use of IF statements to test the number of cycles since the last frame data output. Each operating cycle in the simulation represents 2 milliseconds of the
green band system time history. The number of cycles between interrupts for a frame output is dependent on the film speed to be used. The following equation provides the value for the number of cycles between interrupts for any film speed.

$$CBI = \left(\frac{1}{FS}\right)/0.002$$

CBI is the number of cycles between interrupts, rounded to the nearest integer, and FS is the film speed in frames per second. For rates of 18 and 24 frames per second, the number of cycles are 28 and 21 respectively.

The changes in the GBUD subroutine involve the assignment of numerical values to represent the status of the green band and the division of the green band into 4 foot segments to represent the 4 foot fluorescent lights used in the actual system. Each 4 foot segment was assigned either a value of one or two corresponding to an off or on status respectively. The division of the band into 4 foot lengths was accomplished by changing appropriate constants involved in the GBUD subroutine statements.

The first of the two subroutines added to the simulation, XVEH, uses two methods to provide the current locations of the freeway vehicles. Figure 4 is a simplified flow chart of the XVEH subroutine. The RAWDAT lists created in the ISRINI subroutine are examined on a sensor.
Go to next sensor list

Fig. 4. Flow chart of the XVEH subroutine.
Fig. 4. Flow chart of the XVEH subroutine (continued).
Fig. 4. Flow chart of the XVEH subroutine (continued).
by sensor basis starting from the one closest to the ramp, F1, to the fourth sensor, F4. Lists for sensors F5 through F7 are not examined since they are physically located before the region of the freeway displayed on the graphics terminal. The display begins 57 feet downstream of sensor F4 and extends 1,023 feet along the freeway.

The XVEH subroutine begins by initializing vehicle index counters and setting all vehicle flags to zero. The vehicle flags are used to indicate whether a vehicle of a particular number has been processed during each call of XVEH. Since the sensors are examined from the closest location to the merge point to the farthest, the flags prevent a vehicle location from being recorded from more than the most recent sensor entry.

The routine sets an index variable to one, indicating sensor F1, and then tests if sensor lists for F1 through F4 have been examined. When all lists have been examined, the routine returns control to the main program; otherwise, another index variable is set equal to one to start the examination of the sensor activation entries. The first entry is examined to determine if it has occurred before or at the present simulation time. If the entry has not yet occurred, the routine increments the sensor index variable and begins to examine the next sensor's entries, provided all sensors have not been examined. If the entry has occurred, the vehicle number causing the
entry is determined.

The vehicle flag is checked and if it is set, indicating the vehicle has been handled at a previous sensor, the routine advances to the next entry. If not set, the length of the vehicle, in feet, is calculated as the product of the measured vehicle velocity and its length in seconds. The vehicle is then classified according to its length as a truck, a car, or a motorcycle.

If the entry being examined is an F1 sensor entry, its location is calculated using method one. The measured velocity at the sensor is assumed to be continued through the sensor region, and the location of the vehicle is found by the following equation:

\[
\text{IXLOC} = (\text{RAWDAT}(I, \text{LCNT}, 3)) \times (\text{TIME} - \text{RAWDAT}(I, \text{LCNT}, 2)) + \text{XSNS}(I) + 0.5
\]

The first term is the product of the vehicle velocity and the time since the entry, the second term is the location of the sensor, and the constant one-half is used for round off purposes. The result of using method one is the apparent jerking of the vehicle when its next sensor activation occurs. If the vehicle has slowed down between sensors, the vehicle will appear to go beyond the next sensor location before the next sensor is activated. When the activation does occur, the vehicle will be jerked back to the sensor location. In the case of a vehicle
increasing its speed between sensors the vehicle will jump forward when the next sensor activation occurs. The vehicle location is tested to determine if it is within the region displayed in the picture. If it is, the location is recorded according to vehicle type, the vehicle flag is set and the next entry is examined. If it is not, the vehicle flag is set without recording the location and the routine advances to the next entry.

If the entry had not been from sensor F1 the routine begins to search the sensor's entries for the entry caused by the vehicle under consideration. The search begins at the first list entry and continues to the end of the list if the proper entry is not found. If no entry is found, the vehicle is assumed to have cut out of the right lane and is plotted using method one until it reaches the midpoint of the sensor region. If an entry is found, the location is calculated using both methods.

Method two uses linear interpolation of the simulation time and the sensor entry times to determine the vehicle position. This method causes the vehicles to arrive at the next sensor at the proper time, and a smooth motion of the vehicles is achieved. The vehicle location is calculated using the following equation:

\[
IXLOC = \left(\frac{TIME - RAWDAT(I, LCNT, 2)}{(RAWDAT(I - 1, NCNT, 2) - RAWDAT(I, LCNT, 2)) \times 200}
+ XSNS(I) + 0.5\right)
\]
The first term is the linear interpolation term, and the second and third terms are the same as used in method one. The distance between sensors is 200 feet. This distance is multiplied by a fraction, less than one, which is determined by the amount of time since the sensor entry occurred.

After the vehicle location has been calculated using both methods, the results are recorded, the vehicle flag is set, and the routine advances to the next entry. When the lists for all four sensors have been examined, control is returned to the main program.

The second subroutine, FRMOUT, writes the required frame data into an on-line file. The flow chart for FRMOUT is shown in figure 5. FRMOUT is called immediately after the XVEH subroutine. The output data consist of the simulation time, the green band status, and the vehicle locations and is written such that each record is eighty characters in length. The data are subsequently used by the drawing program to draw the movie on a frame by frame basis at the CRT terminal. The next section describes the terminal hardware and the program used to draw the movie frames.

The Drawing Process

The drawing of each frame of the movie was accomplished with a Tektronix Model 4010 Graphics Terminal and the supportive Tektronix PLOT-10 Graphics Software.
Fig. 5. Flow chart of the FRMOUT subroutine.
The 4010 is a storage tube CRT terminal with a typewriter keyboard for operator input. There are also operator controls for movement of the drawing cursor, but they were not used in this application of the terminal.

The PLOT-10 software is a set of subroutines available for use by the computer from the University of South Florida's Computer Library. The subroutines are user oriented and allow the drawing of pictures through programmer control of the movement of the CRT writing beam and its on-off status. When the beam is on, a line is drawn on the CRT face. When off, the beam is moved without drawing a line. Among the other subroutines available are routines which initialize the terminal, control switching to alphanumerical mode, and end a drawing session. Descriptions of the subroutines and examples of their use are given in the Terminal Control System User's Manual (1972) which is available at the Florida Technological University Computer Center.

The drawing program consists of a main program and a series of subroutines. The main program allows the operator to control the drawing mode and to input variable values. The subroutines call the PLOT-10 routines to draw various portions of the movie frame. Figure 6 is a flow chart for the drawing program.

The main program initializes indexing variables and the terminal. The initialization of the terminal sets
Fig. 6. Flow chart of the drawing program.
Fig. 6. Flow chart of the drawing program (continued).
Fig. 6. Flow chart of the drawing program (continued).
the transmission rate of the terminal in characters per second. It also erases the screen, causes entry into alphanumerical mode, sets margins, and carries out various graphics functions required before any drawing can be done (Terminal Control System User's Manual 1972). The program then instructs the operator to enter the drawing mode to be used.

The program allows two drawing modes; (1) movie mode and (2) search mode. The differences in the two modes will be described as the explanation of the program continues. If the operator selects the movie mode, he is prompted to enter a number which sets the number of frames to be drawn before another mode selection can be made. The program then requests the operator to enter the film code or name to be displayed in the movie frames and to enter the starting frame for the drawing session. The program reads from the frame data to the starting frame. When the starting frame is reached, the program calls the subroutines which draw the picture and branches to another portion of the program determined by the drawing mode selected. In either case, the terminal bell, actually an electronic oscillator, is sounded to indicate the frame drawing is complete.

If the movie mode was selected, the program delays advancing to the next frame. The delay is achieved by causing short movements of the beam, in an off status,
along the bottom of the display. The delay allows the frame to be photographed before the display is erased. The program then tests whether the number of frames required until the next mode selection have been drawn. If not, the program advances one frame and calls the drawing subroutines. If the number of frames selected have been drawn, the program interrupts to allow the re-entering of the drawing mode. The operator can again select which mode he wishes to use. If he selects the movie mode, the program will request the number of frames to be drawn before the next interrupt and the drawing process will resume.

At the start of the drawing session, or at any movie mode interrupt for mode input, the operator can command the program to enter the search mode. In the search mode, the operator must enter an instruction after the drawing of each frame is completed. The operator has five options. He can: (1) hit return, (2) enter '1' and hit return, (3) enter '2' and hit return, (4) enter '3' and hit return, (5) enter '4' and hit return. The hitting of the return key without another input will cause the program to advance and draw the next frame. Entering a '1' and the return key will direct the program to redraw the present frame. Entering a '2' and the return key allows the operator to advance any number of frames. The ability to advance in this manner allows the operator to
search the frame data; thus, the name of the mode is derived. The program prompts the operator to enter the number of frames he wishes to advance, advances to the proper frame, and draws the picture. Entering a '3' and the return key instructs the program to end the drawing session. This is the normal method for ending the program execution. The program execution can be stopped at any time by striking the break key. One depression of the break key will stop the program but allows continuation by striking the return key. If the operator enters 'end' before striking the return key the program execution will be terminated. Entering a '4' and the return key causes the program to enter the movie mode and the operator is prompted to enter the number of frames to be drawn before the next interrupt.

There are six subroutines used for drawing the various portions of the movie frame. Three of the subroutines draw the various types of vehicles. TRKDRW is called to draw vehicles which have been classified as trucks. CARDRW is called to draw cars, and CYLDRW is called to draw motorcycles. Each of the three subroutines calls a special subroutine which draws the wheels for the vehicles. The two remaining subroutines BCKRND and GRBDRW are used to draw the background and green band display respectively. The BCKRND subroutine draws and labels the freeway, ramp, sensor, and merge point locations. It also
writes the film code, the frame number, and the simulation time. The GRBDRW subroutine processes the green band data and in accordance to the band status, draws a line, representing the light being on, or simply moves over the light location without drawing a line. The subroutine also marks the ends of the green band display to facilitate subsequent data reduction when viewing the films.

**Making the Movies**

The physical equipment used to make the movies include the terminal, the camera, and the automatic shutter actuator. Figure 7 is a picture of the equipment set up. When making a movie, the terminal face plate is removed to increase the brightness of the display, and the camera is mounted on a sturdy tripod to insure camera stability during the period of filming.

The camera used is a Bolex H16. It is a spring-driven, 16 millimeter, motion picture camera with a single frame capability. Next day development capability at a local television film department prompted the selection of Kodak VNF 7240 color film. The exposure for single frame photography is controlled by the F-stop which was set at 1.1 for making the movies. A cable release and adapter were attached to the camera to reduce the chance of camera movement when the shutter was released. Any camera movement would cause the movie to jitter when shown at the
Fig. 7. Equipment set up for computer generated green band movie production.
normal frame rate. The cable release can be actuated by the operator or automatically.

Figure 8 is a block diagram of the automatic shutter release mechanism (ASRM). The major components of the ASRM are a sound switch, a silicon-controlled rectifier, a relay driver, a relay, a solenoid, and a power switch for the sound switch.

When the terminal bell sounds, the sound switch triggers the silicon-controlled rectifier (SCR). The SCR in turn triggers the relay driver which actuates the relay. The relay completes the circuit for the solenoid power, and the solenoid pushes the cable release to photograph the displayed movie frame. Once triggered, the sound switch remains on, and thus it was necessary to place a switch between the sound switch power supply and the sound switch itself. The switch turns off the sound switch power when the solenoid is actuated, thus closing the loop to provide automated operation without operator intervention.

Figure 9 is a picture of the ASRM. The unit was constructed by interfacing the various components. The sound switch operates from a 1.5 VDC battery and when triggered provides a 0.7 volt output. Since this voltage is not adequate to drive a solenoid large enough to trigger the camera, it was used to trigger the SCR. The SCR is used as a switch to provide a ground condition for the
Fig. 8. Block diagram of the ASRM.
Fig. 9. The automatic shutter release mechanism.
relay driver. The relay driver circuitry was originally used to drive a small 12 VDC solenoid and was salvaged from an auto tape player. A 12 VDC relay was substituted for the solenoid. The relay provides the switching capability to control a solenoid operating on 110 VAC line voltage. The solenoid pushes the cable release and near the end of its stroke turns off the 1.5 VDC power to the sound switch by mechanical actuation of a microswitch. Figure 10 shows the solenoid and the microswitch used to turn off the sound switch. Turning off the sound switch causes the SCR, the relay driver, the relay, and finally the solenoid to turn off. The solenoid returns to its unactivated position which allows the microswitch to turn the sound switch back on. The unit is then ready for the next activation by the terminal bell. The ASRM is also provided with a switch to disable the sound switch. This allows the system to be turned off while positioning adjustments are made and prevents undesired triggering of the unit.

After the operator has set up the equipment, he logs onto the terminal. He then types in the command 'exec gbx', without the quote marks. The GBX routine is a series of terminal commands that allocate the data sets required to execute the program and then calls the drawing program. The drawing program itself is stored in the form of a load module which has already been compiled.
Fig. 10. Solenoid and microswitch of the ASRM.
When called the execution of the drawing program described in the previous section begins.

The operator enters the search mode and draws the first frame. The camera is then focused and the starting frame is photographed manually. The operator can then turn on the ASRM and instruct the drawing program to enter the movie mode. After inputing the number of frames to be drawn before the next mode interrupt, the program begins to automatically draw and photograph the movie frames.

The exposed film is developed and, when played back at 24 frames per second, shows the dynamic movement of the vehicles and the green bands on a real time basis as generated by the Green Band Control Program Simulation.
CHAPTER IV

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

Two methods are used to calculate the vehicle locations. The first method maintains the measured velocity between the sensors and results in an apparent jerking motion of the vehicles. The second method uses linear interpolation to achieve smooth vehicular motion.

Figures 11 through 13 show the effects of the two methods of calculating the vehicle location. Each vehicle along the freeway is displayed twice in the frame. The top vehicle shows the location of the vehicle as calculated by linear interpolation and moves smoothly along the freeway. The bottom vehicle is the vehicle which appears to jerk.

Figure 11 shows two vehicles (with two representations of each vehicle) approaching the sensor F4 location. Note that the top representation of both vehicles is ahead of the corresponding bottom vehicle. Since both top vehicles are pulling ahead, they must have increased their velocities after crossing over sensor F5. The top vehicles will arrive at sensor F4 when the actual sensor entry occurred; while vehicles, whose positions are projected ahead by maintaining the sensor F5 velocities, will
Fig. 11. Frame 626 of a green band movie.
Fig. 12. Frame 642 of a green band movie.
Fig. 13. Frame 643 of a green band movie.
be jerked forward to the sensor location. Figure 12 shows the top vehicle about to cross the sensor. The next frame, figure 13, shows the bottom vehicle has jumped forward to the sensor.

The crossing of sensor F4 by the lead vehicle has given the Green Band Control Program Simulation an update for the ETA of the vehicle. Since the vehicle velocity has increased, its ETA has decreased as the vehicle will arrive sooner. The change in the ETA has caused the trailing edge of the first band to move forward along the display, as can be seen in figures 12 and 13. The band edge movement occurred within a period of 42 milliseconds and is an example of an undesired instability occurring with the on-line system.

Several movies, using various simulated band control strategies, have been made in the manner presented in this report. A detailed analysis of the movies will be carried out under a project of which this work was only a part. A band control strategy will be selected for implementation in the on-line system using the movies as an aid in the selection.

The movies have been invaluable in visually correlating the unstable band edge movements to the vehicle actions along the freeway. In the first film produced, it was noted that the trailing edge of the bands would allow a ramp vehicle to merge very close to a freeway vehicle.
A later simulation run showed the generation of a band whose trailing edge overlapped the leading edge of the next band. An investigation of the simulation found a sign change, carried out by the on-line system, was not accounted for in the simulation. The incorrect sign caused the trailing edge of all bands to be displaced toward the beginning of the display. Therefore, the films have not only proven to be an aid in analysis of the band edge instability; but have contributed to the validation of the simulation's representation of the on-line system.

The requirement to redraw the entire picture for each frame, imposed by the use of a storage tube CRT, greatly increases the time required to produce a movie. With proper synchronization, a refresh CRT would allow the photographing of the simulation in real time. Presently it requires eight to nine hours to produce a thirty second movie. A reduction by a factor of 1,000 is possible with the use of a refresh CRT.

The modification of the Green Band Control Program Simulation to function interactively on the time-sharing terminal would allow the analyst to change various parameters and run the simulation to immediately see the effects.

Further improvement could be made by storing the simulation in a time-sharing data set. The analyst could easily edit the program to make any control modifications.
which require changes to the program coding and then run the simulation under the analyst's control.

The report has discussed one application in which Computer Graphics can aid the engineer. The specific application presented here shows an example from the field of Transportation Engineering. With the increase in availability of low cost hardware and user-oriented software packages, growth in the use of Computer Graphics is inevitable. It is left to the engineer to recognize possible applications of Computer Graphics and to utilize this valuable analytic tool effectively.
APPENDIX

FORTRAN LISTINGS OF THE DRAWING PROGRAM AND THE GREEN BAND CONTROL PROGRAM SIMULATION
COMMON/NSTRT/ LSTART
DIMENSION IXG8(153), IYGR(153), IXRL(20), IYRL(20), IFIELD(152),
   IXRR(16), IYRR(16), NTKLOC(20), NCRLOC(20), NCYLOC(20)
INTEGER FILMNM(2)
LSTART = 1
NFOP = 0

INITIALIZE THE TERMINAL.
CALL INITT(30)

READ THE DRAWING MODE.
WRITE(6,10)
10 FORMAT (1X,'THE MOVIE MODE DRAWS N FRAMES BETWEEN OPERATOR INPUTS,
   *,1X,'THE PROGRAM CAN BE STOPPED WHILE IN THE MOVIE MODE BY THE BR
   *EAK KEY, *1X,'typing END, AND THEN HIT RETURN.*1X,'IF THE MOVIE M
   *ODE IS TO BE ENTERED NOW ENTER A 1, IF NOT HIT RETURN.*)
READ (5,20) IOP
20 FORMAT (1I1)
   IF (IOP.NE.1) IOP = 2

READ THE VALUE OF N, IF APPLICABLE.
   IF (IOP.EQ.2) GO TO 50
WRITE (6,30)
30 FORMAT (1X,'ENTER THE NUMBER OF FRAMES TO BE DRAWN BEFORE THE NEXT
   *OPERATOR INPUT, THE FORMAT IS 1I4*)
READ (5,40) NN
40 FORMAT (1I4)
READ THE FILM CODE.
50 WRITE (6,60)
60 FORMAT (1X,'ENTER THE FILM CODE, THE FORMAT IS 2A4.')
70 FORMAT (2A4)

READ THE STARTING FRAME NUMBER.
80 WRITE (6,80)
90 FORMAT (1X,'ENTER THE STARTING FRAME NUMBER, THE FORMAT IS 114.')

READ TO THE STARTING FRAME FROM THE FRAME DATA.
NFR = NFRAME
100 DO 130 NF = 1,NFR
110 READ (3,100) TIME,(IFIELD(I),I = 1,72)
120 READ (3,110) (IFIELD(I),I = 73,152)
130 CONTINUE

DRAW THE FRAME.
140 CALL ERASE
150 CALL BCKRND(FILMNM,TIME,NFRAME)
160 CALL GRBDR(1 FIELD)
170 LEND = LSTART + 6
180 DO 180 NTRK = LSTART,LEND
190 IF (NTKLOC(NTRK),EQ.,9999) GO TO 170
200 CALL TRKDRW(NTKLOC(NTRK))
210 CONTINUE
220 LEND = LSTART + 9
230 DO 230 NCAR = LSTART,LEND
240 IF (NCRLOC(NCAR),EQ.,9999) GO TO 190
250 CALL CARDRW(NCRLOC(NCAR))
260 CONTINUE
190 LEND = LSTART + 2
DO 200 NCYL = LSTART, LEND
IF (NCYLOC(NCYL) .EQ. 9999) GO TO 210
CALL CYLDRW(NCYLOC(NCYL))
200 CONTINUE
210 IF (LSTART .EQ. 11) GO TO 220
LSTART = 1
GO TO 150
220 LSTART = 1
GO TO THE PROPER MODE SEGMENT OF THE PROGRAM.
GO TO (230, 280), IOP

THE MOVIE MODE SECTION OF THE PROGRAM ALLOWS THE DRAWING OF N FRAMES WITHOUT OPERATOR INTERVENTION. IT CAN BE STOPPED BY THE BREAK KEY, TYPING END, AND THEN HIT THE RETURN KEY.

INCREMENT THE FRAMES SINCE LAST OPERATOR INPUT COUNTER.
230 NFOP = NFOP + 1
CALL BELL
CALL BELL
DELAY THE PROGRAM TO ALLOW THE PICTURE TO BE TAKEN.
CALL MOVABS(0, 0)
DO 240 MOVIT = 1, 100, 4
240 CALL MOVABS(MOVIT, 0)

INTERRUPT IF N FRAMES HAVE BEEN DRAWN.
IF (NFOP .EQ. NN) GO TO 260
INCREMENT THE FRAME COUNTER AND ADVANCE TO NEXT FRAME.
250 NFRAME = NFRAME + 1
NFR = 1
GO TO 90
GO THROUGH MOVIE OPTION INTERRUPT.
260 CALL ANMODE
CALL ERASE
CALL HOME
WRITE (6,270) NFOP
270 FORMAT (1X,114,2X,"FRAMES HAVE BEEN DRAWN SINCE THE LAST OPERATOR"
* INPUT."/1X,"IF YOU WISH TO CONTINUE THE MOVIE MODE ENTER A 1, IF
* NOT HIT RETURN.")
READ(5,20) IOP
IF (IOP.NE.1) IOP = 2
NFOP = 0
275 IF (IOP.EQ.1) WRITE (6,30) NN
IF (IOP.EQ.1) READ (5,40) NN
GO TO 250
THE SEARCH MODE IS THE DEFAULT IF THE MOVIE MODE IS NOT
SELECTED. IT ALLOWS THE OPERATOR TO REPEAT A FRAME, DRAW
THE NEXT FRAME, ADVANCE MORE THAN ONE FRAME, OR END THE SESSION.
AN OPERATOR INPUT IS REQUIRED AFTER EACH FRAME IS COMPLETED. THE
INPUT DETERMINES WHAT IS TO BE DONE NEXT. THE CODES ARE:

<table>
<thead>
<tr>
<th>INPUT</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETURN</td>
<td>NEXT FRAME IS DRAWN</td>
</tr>
<tr>
<td>1</td>
<td>FRAME IS REPEATED</td>
</tr>
<tr>
<td>2</td>
<td>ADVANCE ONE OR MORE FRAMES</td>
</tr>
<tr>
<td>3</td>
<td>END THE SESSION</td>
</tr>
<tr>
<td>4</td>
<td>ENTER THE MOVIE MODE</td>
</tr>
</tbody>
</table>

SOUND THE TERMINAL BELL.
280 CALL BELL
CALL BELL
CALL HOME
CALL ANMODE
WITHOUT ERASING THE PICTURE, READ THE SEARCH INSTRUCTION.
READ (5,20) INST

CARRY OUT THE INSTRUCTION.
IF (INST.EQ.1) GO TO 290
IF (INST.EQ.2) GO TO 300
IF (INST.EQ.3) GO TO 320
IF (INST.EQ.4) GO TO 340
GO TO 250
REPEAT THE FRAME.
290 GO TO 140

ADVANCE ONE OR MORE FRAMES.
300 CALL ERASE
     WRITE (6,310)
310 FORMAT (1X,'ENTER THE NUMBER OF FRAMES TO ADVANCE, THE FORMAT IS 1
     *14,1')
     READ (5,40) NFADV
     NFRAME = NFRAME + NFADV
     NFR = NFADV
     GO TO 90

END THE DRAWING SESSION.
320 CALL ERASE
     WRITE (6,330)
330 FORMAT (1X,'THE DRAWING SESSION HAS ENDED.')
     CALL FINIT(525,125)

ENTER THE MOVIE OPTION MODE FROM THE SEARCH MODE.
340 IOP = 1
     CALL ERASE
     WRITE (6,350)
350 FORMAT (1X,'THE MOVIE MODE HAS BEEN ENTERED')
     GO TO 275
END

SUBROUTINE BCKRND(FILMNM,TIME,NFRAME)

THE BCKRND SUBROUTINE DRAWS THE BACKGROUND OF THE PICTURE
CONSISTING OF THE FOLLOWING:
(1) THE FREEWAY
(2) THE RAMP
(3) THE SENSOR AND MERGE POINT LOCATIONS
(4) ALL ALPHANUMERIC OUTPUT INCLUDING THE FRAME NUMBER,
     TIME, FILM CODE, AND LABELING.

DIMENSION IXRL(20),IYRL(20),IXRR(16),IYRR(16)
INTEGER FILMNM(2)
DATA IXRL/683,645,607,570,535,504,476,451,431,419,412,413,446,413,*368,398,434,454,466,473/
DATA IXRR/492,437,432,433,441,456,476,504,536,570,608,640,680,758, *844,1023/
DATA IYRR/68,321,359,399,439,476,509,534,563,586,605,622,635,652, *661,662/
CALL MOVABS(0,742) CALL DRWABS(1023,742) CALL MOVA85(757,672) CALL DRWA85(757,678) CALL MOVABS(0,678) CALL MOVA85(757,672) DO 10 I=1,15
10 CALL DRWABS(IXRL(I),IYRL(I)) CALL MOVABS(354,274) DO 20 I=16,20
20 CALL DRWABS(IXRL(I),IYRL(I)) CALL MOVABS(500,0) DO 30 I=1,16
30 CALL DRWABS(IXRR(I),IYRR(I)) CALL MOVABS(543,742) CALL DSHABS(543,678,12) CALL MOVABS(343,742) CALL DSHABS(343,678,12) CALL MOVABS(143,742) CALL DSHABS(143,678,12) CALL MOVABS(960,742) CALL DSHABS(960,662,12) CALL HOME CALL ANMODE WRITE(6,40) (FILMNM(I),I=1,2),NFRAME,TIME
40 FORMAT (/10X,"F3",12X,"F2",13X,"F1",28X,"MP",///1X,2A4,5X,1I8,5X +1F9.3) RETURN END
SUBROUTINE GRADRW(IFIELD)

THE GRADRW SUBROUTINE DRAWS THE GREEN BAND DISPLAY. IF THE LIGHT IS ON A LINE IS DRAWN IF NOT THE BEAM IS MOVED WITHOUT DRAWING.

DIMENSION IXGA(153), IYGB(153), IFIELD(152)
DO 3 LITN=1,152
NBAND=LITN+1
ITLEST=IFIELD(LITN)
GO TO (1,2), ITLEST
1 CALL MOVABS(IXGB(NBAND),IYGB(NBAND))
GO TO 3
2 CALL DRWABS(IXGB(NBAND),IYGB(NBAND))
CONTINUE
3 CALL MOVABS(420,236)
CALL DRWABS(430,240)
CALL MOVABS(724,668)
CALL DRWABS(724,678)
RETURN
END

SUBROUTINE TRKDRW(IXLOC)

THIS ROUTINE DRAWS TRUCKS AT IXLOC.
COMMON/NSTRT/ LSTART
IY = 686
IF (LSTART.EQ.11) IY = 714
IF (IXLOC.LT.50) GOTO 10
IF (IXLOC.GT.973) GOTO 10
CALL MOVAS(IXLOC,IY)
CALL DRWREL(-50,0)
CALL DRWREL(0,18)
CALL DRWREL(32,0)
CALL DRWREL(0,-18)
CALL MOVREL(16,0)
CALL WHEEL
CALL MOVREL(-28,0)
CALL WHEEL
CALL MOVREL(-2,0)
CALL WHEEL
CALL MOVREL(26,14)
CALL DRWREL(8,0)
CALL DRWREL(4,-8)
CALL DRWREL(6,0)
CALL DRWREL(0,-6)
10 RETURN
END

SUBROUTINE CARDRW(IXLOC)

THIS ROUTINE DRAWS CARS AT IXLOC.
COMMON/NSTRT/ LSTART
IY = 686
IF (LSTART.EQ.11) IY = 714
IF (IXLOC.LT.20) GO TO 10
IF (IXLOC.GT.1003) GO TO 10
CALL MOVABS (IXLOC,IY)
CALL DRWREL(-20,0)
CALL DRWREL(0,3)
CALL DRWREL(3,3)
CALL DRWREL(4,1)
CALL DRWREL(4,0)
CALL DRWREL(2,-3)
CALL DRWREL(7,-1)
CALL DRWREL(0,-3)
CALL MOVREL(-2,0)
CALL WHEEL
CALL MOVREL(-8,0)
CALL WHEEL
10 RETURN
END

SUBROUTINE CYLDRW(IXLOC)
THIS ROUTINE DRAWS MOTORCYCLES AT IXLOC.
COMMON/NSTRT/LSTART
IY = 686
IF (LSTART.EQ.11) IY = 714
IF (IXLOC.LT.13) GO TO 10
IF (IXLOC.GT.1010) GO TO 10
CALL MOVABS (IXLOC,IY)
CALL WHEEL
CALL DRWREL(-5,0)
CALL WHEEL
CALL DRWREL(0,1)
CALL DRWREL(1,1)
CALL DRWREL(2,0)
CALL DRWREL(1,1)
CALL DRWREL(5,0)
CALL DRWREL(1,-1)
CALL DRWREL(2,0)
CALL DRWREL(0,-1)
RETURN
END

SUBROUTINE WHEEL

THIS SUBROUTINE DRAWS WHEELS FOR THE VEHICLES.
CALL DRWREL(0,-1)
CALL DRWREL(-1,-1)
CALL DRWREL(-2,0)
CALL DRWREL(-1,1)
CALL DRWREL(0,1)
RETURN
END

**************************************
* LISTING OF THE GREEN BAND CONTROL *
* PROGRAM SIMULATION                  *
**************************************

SIMULATED GREEN BAND EXECUTIVE ROUTINE

ISR CALLED 500 TIMES/SEC
HVP CALLED 100 TIMES/SEC; 1ST CALL ON SECOND INTERRUPT
GBUD CALLED 100 TIMES/SEC; 1ST CALL ON THIRD INTERRUPT
XVEH AND FRMOUT CALLED TO CORRESPOND TO THE FILM SPEED
COMMON/GBT/NBANDS,TRAND(15),VBAND(15),XLE(15),XTE(15),
BT1(15),BT2(15),ALLG,VMRG
COMMON/RCNTRL/ IOUT
COMMON/RUN/ TSTART,TSTOP
COMMON/SFLAG/ ACTF(7)
COMMON/TIMER/ICLOCK,TIME,TNEXT

FOLLOWING VARIABLE CONTROLS DETAILED OUTPUT OPTION...
IOUT = 1
IOUT = 0

FOLLOWING VARIABLE CONTROLS THE RATE OF FRAME DATA OUTPUT
NOUT=28
NOUT=21

ESTABLISH RUN TIME CONDITIONS...
TSTART = 0.850
TSTOP = 2.0

INITIALIZE TIME VARIABLES
TIME = TSTART
ICLOCK = TIME*500.

INITIALIZE VARIABLES FOR ISR AND GBUD Routines...
NBANDS = 0
ALLG = 0.
VMRG = 66.

CALL ISRINI
CALL HVPNINI

SET UP CALLING SEQUENCE COUNTERS FOR SUBROUTINES
NHVP = 3
NGRS = 2
NGBUD = 1
NDISPL=0

MAIN PROCESSING SEQUENCE...
100 CALL ISR
NHVP = NHVP + 1
IF(NHVP.EQ.5) CALL HVP
IF(NHVP.EQ.5) NHVP = 0

NGBS = NGBS + 1
IF(NGBS.EQ.5) CALL GBS
IF(NGBS.EQ.5) NGBS = 0

NGBUD = NGBUD + 1
IF(NGBUD.EQ.5) CALL GBUD
IF(NGBUD.EQ.5) NGBUD = 0

NDISPL=NDISPL+1
IF (NDISPL.EQ.NOUT) CALL XVEH
IF (NDISPL.EQ.NOUT) CALL FRMOUT
IF (NDISPL.EQ.NOUT) NDISPL = 0

UPDATE CLOCK (COUNT OF 2 MILLISECOND INTERRUPT CYCLES)
ICLOCK = ICLOCK + 1
CLOCK = CLOCK + 1
TIME = CLOCK/500.0
IF(TIME.LT.TSTOP) GO TO 100
STOP
END

SUBROUTINE ISRINI
COMMON/INPUT/ RAWDAT(7,100,5),STNEXT(7),INEXT(7)
COMMON/NEXT/ISENS,T1,ICAR,VEL,XLEN,ETA
COMMON/RUN/ TSTART,TSTOP
COMMON/SFLAG/ ACTF(7)
COMMON/TIMER/ICLOCK,TIME,TNEXT
DIMENSION ISENS(7)
DIMENSION ADJUST(7)
DATA ADJUST/0.0,1.839,1.299,-1.55,-4.231,2.93,1.536/

CLEAR THE RAWDAT ARRAY
DO 100 I = 1,7
DO 100 J = 1,100
DO 100 K = 1,5
100 RAWDAT(I,J,K) = 0.0
C CLEAR VEHICLE COUNTERS FOR INPUT ARRAY...
DO 1 I = 1,7
1 INSENS(I) = 0
C BRING IN ENOUGH FREEWAY DATA FROM CARDS TO SATISFY RUN TIMING
PARAMETERS ESTABLISHED IN MAIN...
2 READ(5,200) INSENS,T2,ICAR,VEL,XLEN,ETA
IF (T2.LT.TSTART) GO TO 2
200 FORMAT(9X,I12,F12.0,F12.0)
TIME READ IS T2, ADJUST TO T1 VALUE...
T1 = T2 - (XLEN+6.)/VEL
C CONVERT VEHICLE LENGTH IN FEET TO LENGTH IN TIME...
XLEN = XLEN/VEL
C ADJUST THE VEHICLE VELOCITY.
VEL = VEL + ADJUST(INSENS)
WRITE(6,101) TIME,INSENS,T1,ICAR,VEL,XLEN,ETA
101 FORMAT(1X,'TIME='',F9.3,'',ISRINI READING...',I3,F9.3,I4,3F9.3)
C COUNT NUMBER OF VEHICLES READ AT EACH SENSOR STATION...
INSENS(INSENS) = INSENS(INSENS) + 1
IF(INSENS(INSENS).GT.100) STOP
C LOAD RAW DATA ARRAY FROM PUNCH TO GAP OUTPUT...
RAWDAT(8-ISENS,INSENS(INSENS),1) = ETA
RAWDAT(8-ISENS,INSENS(INSENS),2) = T1
RAWDAT(8-ISENS,INSENS(INSENS),3) = VEL
RAWDAT(8-ISENS,INSENS(INSENS),4) = XLEN
RAWDAT(8-ISENS,INSENS(INSENS),5) = ICAR
C IF(T1.LT.(TSTOP+15.)) GO TO 2
DATA IS READY, SET UP RUN CONTROL FLAGS AND TIMERS...
DO 10 I = 1,7

SURROUTINE ISR
COMMON/INPUT/ RAWDAT(7,100,5),STNEXT(7),INEXT(7)
COMMON/NEXT/ISENS,T1,ICAR,VEL,XLEN,ETA
COMMON/SFLAG/, ACTF(7)
COMMON/TIMER/ICLOCK,TIME,TNEXT

THIS SUBROUTINE SIMULATES THE SCANNING OF FIELD SENSOR DATA IN THE ON-LINE SYSTEM BY USING HISTORICAL DATA OBTAINED FROM REDUCED DATA GENERATED BY THE TOGAP SYSTEM.

NOTE... THE PROGRAM DOES NOT LOOK AHEAD IN TIME. THE VARIABLE TNEXT INDICATES WHEN THE NEXT FIELD SENSOR DATA WILL BE READY. UNTIL THAT TIME, ISR REPORTS NO NEW DATA.

SCAN SENSORS, TURN ON 'DATA READY' FLAGS IF APPROPRIATE...
DO 1000 I = 1,7
IF (TIME.GE.STNEXT(I)) ACTF(I) = -1.0
1000 CONTINUE

SEARCH FOR SMALLEST OF THE NEXT SENSOR ACTIVATION TIMES...
NOTE... THIS VALUE IS NOT CURRENTLY USED IN THE SIMULATION BUT COULD BE USED IN A SIMULATION STRATEGY BASED ON EVENT SCANNING RATHER THAN THE CURRENT MODE OF TIME SCANNING.

TNEXT = 1.0E60
DO 2000 I = 1,7
IF (STNEXT(I).LT.TNEXT) TNEXT=STNEXT(I)
2000 CONTINUE
SUBROUTINE HVPINI
COMMON/BINS/ SETA(9,7),ST1(9,7),SVEL(9,7),SLEN(9,7),
* SNC(9,7),NLIST(7)
COMMON/FWYDAT/NVEH,FETA(63),FT1(63),FVEL(63),FXLEN(63),NC(63)

THIS ROUTINE CLEARS THE HIGHWAY VEHICLE DATA LIST COUNTS TO
ZERO VALUES TO SET UP THE INITIAL HIGHWAY VEHICLE PROCESSING
CONDITIONS.

DO 50 I = 1,7
50 NLIST(I) = 0
NVEH = 0
RETURN
END

SUBROUTINE HVP
COMMON/BINS/ SETA(9,7),ST1(9,7),SVEL(9,7),SLEN(9,7),
* SNC(9,7),NLIST(7)
COMMON/FWYDAT/NVEH,FETA(63),FT1(63),FVEL(63),FXLEN(63),NC(63)
COMMON/NEXT/ISENS,T1,ICAR,VEL,XLEN,ETA
COMMON/OCNTRL/ IOUT
COMMON/SFLAG/ ACTF(7)
COMMON/TIMER/ICLOCK,TIME,TNEXT
COMMON /EXPT/ GBDATA

NOTE ••• THOLD CONTROLS FWY VEHICLE RETENTION TIME IN DATA TABLES...
DATA THOLD / 20.48 /

THIS ROUTINE SIMULATES THE ACTIONS OF THE HVP ROUTINE IN
THE ON-LINE GREEN BAND CONTROL PROGRAM. NOTE THAT THE
VARIABLES IT USES TO UPDATE THE FREEWAY LISTS ARE ALREADY COMPUTED AND SUPPLIED BY ISR. IN THE ON-LINE SYSTEM, THESE VALUES MUST BE COMPUTED BY THE HVP FROM SENSOR EVENT TIME DATA.

PROCESS CURRENT FWY DATA LISTS TO REMOVE ANY VEHICLES PAST DUE AT THE MERGE POINT...

DO 200 ILIST = 1,7 JNV = NLIST(ILIST) IF(JNV.EQ.0) GO TO 200 NSQ = 0
DO 100 JVEH = 1,JNV
MARK OVERTIME VEHICLES WITH NEGATIVE ETA VALUE...
IF(SETA(JVEH,ILIST).GE.TIME) GO TO 100
SETA(JVEH,ILIST) = -99.
NSQ = NSQ + 1
100 CONTINUE
ROUTINE 'SQUEEZE' REMOVES ALL VEHICLES IN CURRENT LIST WITH NEGATIVE ETA VALUES...
IF(NSQ.GT.0) CALL SQUEEZE(ILIST)
200 CONTINUE
DO LIST SCAN FOR NEW DATA UPDATES...
GDATA = -1.
GDATA = 1.
DO 400 I = 1,7
IS THERE NEW DATA READY FOR THIS LIST?
IF(ACTF(I).GE.0.) GO TO 400
DATA IS AVAILABLE FOR LIST I, IS THE LIST CURRENTLY EMPTY?
401 IF(NLIST(I).GT.0) GO TO 405
THE CURRENT LIST IS EMPTY, SO INSERT THE NEW DATA AND CONTINUE PROCESSING WITH THE NEXT LIST...
CALL INSERT(I)
GO TO 400
THE CURRENT LIST HAS AT LEAST ONE ENTRY.
CHECK VEHICLE RETENTION TIME FOR THE TOP VEHICLE IN LIST (E.G., IS THE VEHICLE OVERDUE AT THE NEXT DOWNSTREAM SENSOR?)

405 IF((ST1(I,I)*THOLD).GT.TIME). GO TO 410

FIRST VEHICLE IS OVERDUE, DELETE IT AND PUSH LIST UP...
SETA(I,I) = -99.
CALL SQUEZE(I)
PUT NEW DATA IN UPDATED LIST...
CALL INSERT(I)

FIRST VEHICLE IN THE LIST IS NOT OVERDUE, CHECK TO SEE IF THE LIST IS FULL...
410 IF(NLIST(I).EQ.9) GO TO 420

LIST IS NOT FULL, STORE THE NEW DATA...
CALL INSERT(I)
GO TO 400

LIST IS NOW FULL, DELETE FIRST VEHICLE...
420 SETA(I,I) = -99.
CALL SQUEZE(I)
LIST NOW HAS 8 ENTRIES, STORE THE NEW DATA IN THE LAST POSITION...
CALL INSERT(I)
400 CONTINUE

NOW FINISHED WITH INDIVIDUAL SENSOR LIST PROCESSING, SO COPY THE INDIVIDUAL SENSOR BINS INTO THE MASTER ARRAY USED BY THE SIMULATED GRS ROUTINE...
IGO = 1
DO 500 ILIST = 1,NV
      NV = NLIST(ILIST)
      IF(NV.EQ.0) GO TO 500
      DO 501 ICOPY = 1,NV
            FETA(IGO) = SETA(ICOPY,ILIST)
            FT1(IGO) = ST1(ICOPY,ILIST)
            FVEL(IGO) = SVEL(ICOPY,ILIST)
            FXLEN(IGO) = SLEN(ICOPY,ILIST)
            NC(IGO) = SNC(ICOPY,ILIST)
            IGO = IGO + 1
   501 CONTINUE
   500 CONTINUE
COMPUTE NUMBER OF VEHICLES IN CURRENT LIST...
NVEH = IGO - 1
IF(IOUT.EQ.1)  CALL BINOUT(1)
RETURN
END

SUBROUTINE INSERT(ILIST)
COMMON/BINS/  SETA(9,7),ST1(9,7),SVEL(9,7),SLEN(9,7),
*  SNC(9,7),NLIST(7)
COMMON/INPUT/  RAWDAT(7,100,5),STNEXT(7),INEXT(7)
COMMON/NEXT/ISENS,T,ICAR,VEL,XLEN,ETA
COMMON/SFLAG/ACTF(7)
DATA HMIN/0.8/

THIS ROUTINE PLACES NEW FREEWAY VEHICLE DATA IN THE CURRENT DATA
LIST REPRESENTING THE SENSOR 'BIN' AT LOCATION ILIST IN THE ON-LINE
PROGRAM.

IF CURRENT LIST EMPTY, GO TO SIMPLE INSERTION CASE...
IF(NLIST(ILIST).EQ.O) GO TO 200

CURRENT LIST IS NOT EMPTY, PUT NEW DATA AT END...
SETA(NLIST(ILIST)+1,ILIST) = RAWDAT(ILIST,INEXT(ILIST),1)
ST1(NLIST(ILIST)+1,ILIST) = RAWDAT(ILIST,INEXT(ILIST),2)
SVEL(NLIST(ILIST)+1,ILIST) = RAWDAT(ILIST,INEXT(ILIST),3)
SLEN(NLIST(ILIST)+1,ILIST) = RAWDAT(ILIST,INEXT(ILIST),4)
SNC(NLIST(ILIST)+1,ILIST) = RAWDAT(ILIST,INEXT(ILIST),5)

NOW DO MINIMUM FWY VEHICLE ETA TEST ... E.G., IF COMPUTED MP ETA
IS LESS THAN THAT OF THE PREVIOUS VEHICLE, SET THE NEW ETA EQUAL
TO THE OLD VALUE + HMIN.

IF(SETA(NLIST(ILIST)+1,ILIST).LT.SEITA(NLIST(ILIST),ILIST))
  *  SETA(NLIST(ILIST)+1,ILIST) = SETA(NLIST(ILIST),ILIST) + HMIN
BUMP LIST COUNT...
NLST(ILIST) = NLST(ILIST) + 1
GO TO 200

LIST WAS EMPTY, INSERT NEW DATA AT FIRST POSITION...

SETA(1,ILIST) = RAWDAT(ILIST,INEXT(ILIST),1)
ST1 (1,ILIST) = RAWDAT(ILIST,INEXT(ILIST),2)
SVEL(1,ILIST) = RAWDAT(ILIST,INEXT(ILIST),3)
SLEN (1,ILIST) = RAWDAT(ILIST,INEXT(ILIST),4)
SNC (1,ILIST) = RAWDAT(ILIST,INEXT(ILIST),5)

ADJUST LIST COUNT TO REFLECT SINGLE ENTRY...
NLST(ILIST) = 1

TURN OFF 'SENSOR ACTIVE' FLAG AND UPDATE POINTER VARIABLES TO FUTURE SENSOR ACTIVATIONS ON THE CHANNEL...

ACTF(ILIST) = +1.0
INEXT(ILIST) = INEXT(ILIST) + 1
STNEXT(ILIST) = RAWDAT(ILIST,INEXT(ILIST),2)

RETURN
END

SUBROUTINE SQUEZE(ILIST)
COMMON/BINS/  SETA(9,7),ST1(9,7),SVEL(9,7),SLEN(9,7),
*  SNC(9,7),NLST(7)

THIS ROUTINE COMPRESSES FREEWAY DATA LISTS BY MOVING TABLE ENTRIES UP IN THE SPECIFIED LIST TO REPLACE SENSOR REPORTS MARKED AS UNACCEPTABLE BY THE HVP.

THE 'DATA UNACCEPTABLE' CONDITION FOR THE SIMULATION SCHEME IS INDICATED BY A NEGATIVE VALUE FOR A VEHICLE ETA SET BY HVP.

IF(NLIST(ILIST).GT.1) GO TO 200

LIST HAD ONLY ONE ELEMENT, DELETE BY ZEROING COUNT.
NLST(ILIST) = 0
RETURN

FIND FIRST NEGATIVE ETA IN CURRENT LIST...
200 NNLIST = NLIST(ILIST)
DO 205 IBAD = 1,NNLIST
IF (SETA(IBAD,ILIST) .LT. 0.) GO TO 207
205 CONTINUE

LIST IS NOW O.K., RETURN TO HVP...
RETURN

SQUEEZE LIST UP FROM BAD DATA ROW...
207 IF (IBAD.EQ.NNLIST) GO TO 209
INMI = NNLIST - 1
DO 208 IMOVE = IBAD, INMI
SETA(IMOVE,ILIST) = SETA(IMOVE+1,ILIST)
ST1(IMOVE,ILIST) = ST1(IMOVE+1,ILIST)
SVEL(IMOVE,ILIST) = SVEL(IMOVE+1,ILIST)
SLEN(IMOVE,ILIST) = SLEN(IMOVE+1,ILIST)
SNC(IMOVE,ILIST) = SNC(IMOVE+1,ILIST)
208 CONTINUE
NLIST(ILIST) = NLIST(ILIST) - 1
IF (NLIST(ILIST) .EQ. 0) RETURN
GO TO 200
209 NLIST(ILIST) = NLIST(ILIST) - 1
RETURN
END

SUBROUTINE BINOUT(IOPT)
COMMON/BINS/ SETA(9,7),ST1(9,7),SVEL(9,7),SLEN(9,7),
SNC(9,7),NLIST(7)
COMMON/FWYDAT/NVEH,FETA(63),FT1(63),FVEL(63),FXLEN(63),NC(63)
COMMON/TIMER/ICLOCK,TIME,TNEXT

THIS ROUTINE PRINTS THE CURRENT CONTENTS OF THE SEVEN HIGHWAY
VEHICLE DATA BINS IN THE SIMULATED HVP ROUTINE.
IF (IOPT.EQ.1) WRITE (6,10) TIME,NVEH
10 FORMAT(/' HVP FINISHED AT TIME = ',F9.3,' WITH NVEH = ',I4)
DO 100 IS = 1,7
WRITE(6,90) IS,IS,NLIST(IS)
90 FORMAT(1X,'SENSOR BIN ',I2,' ... NLIST(',I2,')=',I2)
IF(NLIST(IS).EQ.0) GO TO 100
NV = NLIST(IS)
DO 95 JV = 1,NV
WRITE(6,96) JV,SETA(JV,IS),ST1(JV,IS),SVEL(JV,IS),
SLEN(JV,IS),SNC(JV,IS)
96 FORMAT(1X,'VEH. NO. = ',I2,' ETA,T1,VEL,LEN,CARID=','4F10.3,F5.0)
100 CONTINUE
RETURN
END

SUBROUTINE GBS
COMMON/BINS/ SETA(9,7),ST1(9,7),SVEL(9,7),SLEN(9,7),
SNC(9,7),NLIST(7)
COMMON/FWYDAT/NVEH,FETA(63),FT1(63),FVEL(63),FXLEN(63),NC(63)
COMMON/GBT/NBANDS,TBAND(15),VBAND(15),XLE(15),XTE(15),
BT1(15),BT2(15),ALLG,VMRG
COMMON/OCNTRL/ IOUT
COMMON/TIMER/ICLOCK,TIME,TNEXT
COMMON /EXPT/ GBDATA
REAL*4 THDY,LHDS
DATA THDY,LHDS =-1.2,0.4/
DATA ZERO /0.0/
THIS ROUTINE SIMULATES THE GBS ROUTINE OF THE ON-LINE CONTROL SYSTEM. FREEWAY VEHICLE DATA IS SEARCHED FOR ACCEPTABLE GAPS. BANDS REPRESENTING THESE GAPS ARE CREATED FOR OUTPUT BY THE GRUD ROUTINE.

PRINT FREEWAY VEHICLE DATA TABLES...
IF(IOUT.EQ.0) GO TO 8891
WRITE(6,8888) TIME,NVEH,NLIST
8888 FORMAT(1X,'TIME=',F9.3,', NVEH = ',I3,', NLIST= ',7I5)
IF(NVEH.EQ.0) GO TO 8891
DO 8889 IV = 1,NVEH
8889 WRITE(6,8890) IV,FETA(IV),FT1(IV),FVEL(IV),FXLEN(IV),NC(IV)
8890 FORMAT(1X,FHWY VEH NO. ',',I2, ',',F12.3,I5)
8891 CONTINUE
C ABORT THE RUN IF DATA TABLES ARE OVERFLOWING...
  IF(NVEH.GT.63) STOP
C TEST FOR UPSTREAM SECTION 'ALL CLEAR' CONDITION...
  IF(NVEH.GT.0) GO TO 100
  ALLG = 1.
  RETURN
C SET BAND COUNTER TO ZERO, CREATE NEW SET...
  CONTINUE
  IF (GBDATA.LT.0.) RETURN
  NBANDS = 0
C TURN OFF ALL GREEN FLAG TO INDICATE THERE ARE ACTIVE BANDS...
  ALLG = 0.
C CREATE FIRST BAND
  NVSUM = 0
  DO 10 NSENS = 1,7
    NVSUM = NVSUM + NLIST(NSENS)
  IF (NLIST(NSENS).EQ.0) GO TO 10
  GO TO 11
10 CONTINUE
11 NL = 1
  NT = 1
  NBANDS = NBANDS + 1
  XLE(NBANDS) = 844.
  CALL XLTE(FETA(1),ZERO,THDY,VBAND(NBANDS),XTE(NBANDS))
  TBAND(NBANDS) = TIME
  BT1(NBANDS) = -1.
  BT2(NBANDS) = FETA(1)
  IF(NVEH.EQ.1) GO TO 300
C DO INTERMEDIATE BAND PROCESSING...
  NT = NT + 1
  IF (NT.GT.NVEH) GO TO 300
  IF (NT.LE.NVSUM) GO TO 210

NSENS = NSENS + 1
NSUM = NVSUM + NLIST(NSENS)
IF (FETA(NT) .GT. (FETA(NL) + 0.25)) GO TO 210
GO TO 200
C CHECK FOR MIN ETA SPACING OF 2 SECONDS
NLISTCNSENS) )
IF (FETA(NT) .GT. (FETA(NL) + 2.0)) GO TO 250
NL = NT
GO TO 200
C COMPUTE MEASURED GAP
XMGAP = FETA(NT) - FETA(NL) - FXLEN(NL)
C COMPUTE REQUIRED GAP
CALL RAYGAP(NT, NL, RGAP)
IF (IOUT .EQ. 1) WRITE(6, 2000) TIME, NT, NL, XMGAP, RGAP
2000 FORMAT (1X, 'TIME=', F9.3, ', TVEH, LVEH, MGAP, RGAP...', 213, 2F12.3)
C IS THIS GAP ADEQUATE?
IF (XMGAP .GE. RGAP) GO TO 275
NL = NT
GO TO 200
C YES, CREATE NEW BAND...
NBANDS = NBANDS + 1
CALL XLTE(FETA(NL), FXLEN(NL), LHDY, VBAND(NBANDS), XLE(NBANDS))
CALL XLTE(FETA(NT), ZERO, THDY, VDUMMY, XTE(NBANDS))
TBAND(NBANDS) = TIME
BT1(NBANDS) = FETA(NL)
BT2(NBANDS) = FETA(NT)
NL = NT
GO TO 200
C DO PROCESSING FOR LAST BAND
NBANDS = NBANDS + 1
CALL XLTE(FETA(NVEH), FXLEN(NVEH), LHDY, VBAND(NBANDS), XLE(NBANDS))
XTE(NBANDS) = -9999
TBAND(NBANDS) = TIME
BT1(NBANDS) = FETA(NVEH)
BT2(NBANDS) = -1.0
C CONTINUE
IF (IOUT .EQ. 0) RETURN
WRITE(6, 1001) TIME, NBANDS
FORMAT(1X,'TIME=',F9.3,'GBS FINISHED WITH NBANDS=',I5)
IF(NBANDS.EQ.0) RETURN
DO 1002 J = 1,NBANDS
1002 WRITE(6,1003)J,TBAND(J),VBAND(J),XTE(J),XLE(J),BT1(J),BT2(J)
1003 FORMAT(1X,'BAND NO.',12,'...',6F12.3)
RETURN
END

SUBROUTINE XLTE(ZETA,ZLEN,ZLTHDY,VNOW,EDGE)
COMMON/GBT/NBANDS,TBAND(15),VBAND(15),XLE(15),XTE(15),
* BT1(15),BT2(15),ALLG,VMRG
COMMON/TIMER/ICLOCK,TIME,TNEXT
DATA ACC/3.0/

ROUTINE TO SIMULATE BAND POSITIONING LOGIC FOR TAMPA VARIABLE GREEN BAND SPEED CASE.

UFTMP = ZETA + ZLTHDY + ZLEN - TIME
WORK1 = (VMRG*VMRG - 1936.)/6.
PAM = (580. - WORK1)/VMRG
UFTAE = UFTMP - PAM
UFTSL = UFTAE - ((VMRG-44.)/ACC)
UFTDB = UFTSL - 6.

IF(UFTMP.LT.0.) GO TO 100
IF(UFTAE.LT.0.) GO TO 200
IF(UFTSL.LT.0.) GO TO 300

BAND EDGE IDENTIFIED AS BEING IN 30 MPH SECTION...
SIMULATION OF ON-LINE CODE AT LABEL SLOW...
VNOW = 44.
EDGE = -UFTDB + 44.
RETURN

SIMULATION OF ON-LINE CODE AT LABEL ATMP...
100 VNOW = VMRG
EDGE = 844.
RETURN
SIMULATION OF ON-LINE CODE AT LABEL FAST...

200 VNOW = VMRG
EDGE = 844. - (UFTMP*VMRG)
RETURN

SIMULATION OF ON-LINE CODE AT LABEL UFACC...

300 VNOW = 44. - (UFTSL*3.)
SET BAND 'STILL ACCELERATING' CONDITION...
VNOW = - VNOW
RETURN
END

SUBROUTINE RAYGAP(ITRAIL, ILEAD, RGAP)
COMMON/FWYDAT/NVEH,FETA(63),FT1(63),FVEL(63),FXLEN(63),NC(63)
COMMON/GBT/NBANDS,TBAND(15),V8AND(15),XLE(15),XTE(15),
* BT1(15),BT2(15),ALLG,VMRG
DATA HZERO /0.8/
DATA DEE /4.0/
DATA TLRAMP /0.3/

RAYGAP IMPLEMENTS THE ALGORITHM USED BY RAYTHEON TO COMPUTE
THE REQUIRED GAP SIZE FOR GREEN BAND GENERATION AS A FUNCTION
OF THE VELOCITIES OF THE LEADING AND TRAILING FREEWAY VEHICLES
DEFINING A GAP, AND THE VELOCITY OF THE RAMP VEHICLE BEING
CONSIDERED FOR THE GAP. SINCE THE GREEN BAND RAMP VEHICLES ARE
NOT TRACKED BY THE ON-LINE SYSTEM, THE CURRENT GREEN BAND SPEED
IS USED FOR THE RAMP VEHICLE SPEED.

RGAP = (H1+H1P+HZERO) + (H2+H2P+HZERO) + TLRAMP

VT = FVEL(ITRAIL)
VL = FVEL(ILEAD)

IF(VT.GT.VL.AND.VMRG.GT.VL) GO TO 100
IF(VT.GT.VL.AND.VL.GT.VMRG) GO TO 200
IF(VL.GT.VT.AND.VT.GT.VMRG) GO TO 300
IF(VL.GT.VT.AND.VMRG.GT.VT.AND.VMRG.GT.VL) GO TO 400
IF(VL.GT.VT.AND.VMRG.GT.VT.AND.VLR.GT.VMRG) GO TO 500
RGAP = 1.0E50
RETURN
C
100 H1 = (VMRG-VL)*(VMRG-VL)/(2.0*DEE*VL)
    H2 = (VT-VL)*(VT-VL)/(2.0*DEE*VL)
    GO TO 600
C
200 H1 = 0.
    H2 = ((VT-VL)*(VT-VL)/(2.0*DEE*VL)) + F2(VMRG,VL)
    GO TO 600
C
300 H1 = 0.
    H2 = F2(VMRG,VT)
    GO TO 600
C
400 H1 = (VMRG-VL)*(VMRG-VL)/(2.0*DEE*VL)
    H2 = 0.
    GO TO 600
C
500 H1 = 0.
    H2 = 0.
    GO TO 600
C
COMPUTE ETA PREDICTION ERROR ALLOWANCE... 600
H1P = 0.777*F3(VL)*FETA(ILEASE-FI1(ILEASE))
H2P = 0.777*F3(VT)*FETA(ISTRAIL-FI1(ISTRAIL))
C
RGAP = H1 + H1P + H2 + H2P + 2.0*HZERO + TLRAMP
RETURN
END

FUNCTION F2(V1,V2)
THIS FUNCTION COMPUTES THE TIME TO ACCELERATE FROM V1 TO V2
IN SECONDS. V1 AND V2 ARE TO BE SUPPLIED IN FT/SEC.
VZERO = 146.7
ALPHA = 5.
VDIFF = V2 - V1
TERM = (VZERO-V2)/(VZERO-V1)
FTIME = (VZERO/(ALPHA*V2))
FTIME = FTIME + (VZERO*V2)*ALOG(TERM)

ATIME = 0.11*VDIFF - 0.925
IF(VDIFF.LE.17.5) ATIME = 0.057*VDIFF
IF(VDIFF.GE.41.) ATIME = 0.16*VDIFF - 2.85

IF THE 'TRUE' FORMULA FOR F2 IS TO BE USED, SET F2 EQUAL TO
THE VARIABLE FTIME IN THE FOLLOWING ASSIGNMENT.
F2 = ATIME
RETURN
END

FUNCTION F3(VEL)
F3(VEL) COMPUTES A TERM IN THE EQUATION USED BY RAYTHEON
TO PREDICT FWY VEHICLE ETA UNCERTAINTY IN THE REQUIRED GAP
CALCULATIONS.
VEL IS SUPPLIED IN UNITS OF FT/SEC
F3 = 0.01 + ((88.-VEL)/73.5)
IF(VEL.LT.14.7) F3 = 0.10
IF(VEL.GT.88.0) F3 = 0.01
RETURN
END

SUBROUTINE GAUD
COMMON/GBT/NBANDS,TBAND(15),VBAND(15),XLE(15),XTE(15),

* COMMON/TIMER/ICLOCK,TIME,TNEXT
COMMON/LIGHTS/FIELD(160)
INTEGER*2 FIELD
REAL*4 ACCEL /3.0/
DATA DT /0.010/

C THIS ROUTINE SIMULATES THE GBUD ROUTINE IN THE ON-LINE SYSTEM.
DATA IN THE CURRENT GREEN BAND TABLE IS UPDATED AND DISPLAYED ON THE SIMULATED FIELD LIGHTS.

IF(ALLG.LT.1)  GO TO 100
DO 50 1=1,160
   50 FIELD(I)=2
GO TO 1000

C CLEAR BAND DISPLAY FOR NEW DATA...
DO 101 I=1,160
   101 FIELD(I)=1

C SCAN BAND TABLE
IF(NBANDS.EQ.0)  GO TO 1000
DO 105 K = 1,NBANDS
C DO NOT PROCESS BAND IF XTE IS FARTHER ALONG DISPLAY THAN XLE...
   IF(XTE(K).GE.XLE(K))  GO TO 105

C CONSTANT VELOCITY BAND?
IF(VBAND(K).GT.0.)  GO TO 150

C NO, STILL ACCELERATING...
   VB = ABS(VBAND(K)) + ACCEL*DT
   IF(VB.GE.VMRG)  VBAND(K) = VMRG
   IF(VB.LT.VMRG)  VBAND(K) = -VB

C UPDATE LEADING AND TRAILING EDGES OF BAND...
   DX = VB * DT
   XTE(K) = XTE(K) + DX
   XLE(K) = XLE(K) + DX
IGO=XTE(K)/4.0
ISTOP=XLE(K)/4.0

C
IF(IGO.GT.160) GO TO 105
IF(ISTOP.LE.0) GO TO 105
IF(ISTOP.GT.160) ISTOP=160
IF(IGO.LE.0) IGO=1
IF(IGO.GE.ISTOP) GO TO 105

DO 160 I = IGO,ISTOP
160 FIELD(I)=IGO,ISTOP
1000 CONTINUE

C
RETURN
END

SUBROUTINE XVEH
COMMON/INPUT/ RAWDAT(7,100),STNEXT(7),INEXT(7)
COMMON/XLOCAL/ NTKLOC(20),NCRLOC(20),NCYLOC(20)
COMMON/TIMER/ ICLOCK,TIME,TNEXT
INTEGER*2 NTKLOC,NCRLOC,NCYLOC,IFLAG(780)
DIMENSION XSNS(4)
DATA XSNS / 543.0,343.0,143.0,-57.0 /

THIS ROUTINE PLOTS THE VEHICLE LOCATIONS USING TWO METHODS. METHOD 1 MAINTAINS THE VELOCITIES MEASURED AT THE SENSORS UNTIL THE NEXT SENSOR ENTRY OCCURS. METHOD 2 USES LINEAR INTERPOLATION TO ACHIEVE SMOOTH VEHICLE MOTION BETWEEN THE SENSORS.

SET VEHICLE FLAGS TO ZERO.
DO 10 N = 1,780
10 IFLAG(N) = 0

INITIALIZE VEHICLE COUNTERS AND LOCATIONS.
NTRK = 0
NCAR = 0
NCYL = 0
DO 20 N = 1,20
START AT SENSOR F1 ENTRIES.
I = 1

RETURN WHEN LISTS FOR SENSORS F1 THROUGH F4 HAVE BEEN EXAMINED.
30 IF (I.GT.4) RETURN

SET LIST ENTRY COUNTER.
LCNT = 1

HAS THIS ENTRY OCCURRED?
40 IF (TIME.GE.RAWDAT(I,LCNT,2)) GO TO 50
NO, GO TO THE NEXT SENSOR LIST.
I = I + 1
GO TO 30

YES, WHAT IS THE VEHICLE NUMBER?
50 IVEH = RAWDAT(I,LCNT,5)

HAS THE VEHICLE BEEN PROCESSED?
IF (IFLAG(IVEH).EQ.1) GO TO 80
NO, CALCULATE THE VEHICLE LENGTH.
XLENTH = (RAWDAT(I,LCNT,4)) * (RAWDAT(I,LCNT,3))

CLASSIFY THE VEHICLE ACCORDING TO LENGTH.
ITYPE = 3
IF (XLENTH.GT.10.0) ITYPE = 2
IF (XLENTH.GT.24.0) ITYPE = 1
IF (I.EQ.1) GO TO 90

SEARCH FOR THE VEHICLES ENTRY IN THE NEXT SENSOR LIST.
DO 60 ICNT = 1,100
NCNT = ICNT
NVEH = RAWDAT(I-1,ICNT,5)
IS THIS THE PROPER ENTRY?
IF (IVEH.EQ.NVEH) GO TO 100
60 CONTINUE
NO ENTRY AT THE NEXT SENSOR. PLOT UNTIL THE MIDPOINT OF THE REGION USING METHOD 1.

```
C MIDPT = XSNS(I) + 100.0
IXLOC = (RAWDAT(I,LCNT,3)) * (TIME - RAWDAT(I,LCNT,2)) + XSNS(I) + 0.5
IF (IXLOC.GT.MIDPT) GO TO 150
KXLOC = IXLOC
GO TO 110
```

GO TO THE NEXT ENTRY.

```
80 LCNT = LCNT + 1
GO TO 40
```

AN F1 ENTRY. PLOT UNTIL THE VEHICLE IS OUT OF THE DISPLAYED REGION USING METHOD 1.

```
90 IXLOC = (RAWDAT(I,LCNT,3)) * (TIME - RAWDAT(I,LCNT,2)) + XSNS(I) + 0.5
IF (IXLOC.GT.1023) GO TO 150
KXLOC = IXLOC
GO TO 110
```

AN F2 THROUGH F4 ENTRY WITH AN ENTRY AT THE NEXT SENSOR. PLOT USING BOTH METHODS.

```
100 IXLOC = 0.5 * XSNS(I) + 200.0 * (TIME - RAWDAT(I,LCNT,2)) /
  (RAWDAT(I-1,NCNT,2) - RAWDAT(I,LCNT,2))
  + XSNS(I) + 0.5
KXLOC = (RAWDAT(I,LCNT,3)) * (TIME - RAWDAT(I,LCNT,2)) + XSNS(I) + 0.5
```

RECORD THE VEHICLE LOCATION ACCORDING TO THE VEHICLE TYPE.

```
110 GO TO (120,130,140), ITYPE
120 NTRK = NTRK + 1
NTKLOC(NTRK) = KXLOC
NTKLOC(NTRK + 10) = IXLOC
GO TO 150
130 NCAR = NCAR + 1
NCRLOC(NCAR) = KXLOC
NCRLOC(NCAR + 10) = IXLOC
GO TO 150
140 NCYL = NCYL + 1
NCYLOC(NCYL) = KXLOC
```
NCYLOC(NCYL+10) = IXLOC

SET THE VEHICLE FLAG.

150 IFLAG(IVEH) = 1
GO TO 80
END

SUBROUTINE FRMOUT
THIS SUBROUTINE WRITES THE DATA REQUIRED TO DRAW A PICTURE INTO A
PREVIOUSLY ALLOCATED DATA FILE. THE DATA INCLUDES:

(1) THE GREEN BAND STATUS
(2) THE VEHICLE LOCATIONS
(3) THE SIMULATION TIME

COMMON/XLOCAL/NTKLOC(20), NCRLOC(20), NCYLOC(20)
INTEGER*2 FIELD, NTKLOC, NCRLOC, NCYLOC
COMMON/TIMER/I_CLOCK, TIME, T_NEXT
COMMON/LIGHTS/FIELD(160)

1 FORMAT (1X,F8.3,72I1)
2 FORMAT (1X,20I4)
3 FORMAT (1X,20I4)
WRITE(3,1) TIME, (FIELD(I), I = 9,80)
WRITE(3,2) (FIELD(I), I = 81,160)
WRITE(3,3) (NTKLOC(I), I = 1,7), (NCRLOC(I), I = 1,10), (NCYLOC(I),
* I = 1,3)
WRITE(3,3) (NTKLOC(I), I = 11,17), (NCRLOC(I), I = 11,20), (NCYLOC(I),
* I = 11,13)
RETURN
END
LIST OF REFERENCES


Spiral auto jump entertains movie goers, may result in safer highways. 1975. Transportation Research News. 58: 10-12.