A MODEL FOR TRAFFIC SIMULATION AND CONTROL*

By

Roger S. Walker and B. F. Womack
Department of Electrical Engineering

Technical Memorandum No. 15
January 10, 1970

INFORMATION SYSTEMS RESEARCH LABORATORY

ELECTRONICS RESEARCH CENTER
THE UNIVERSITY OF TEXAS AT AUSTIN
Austin, Texas 78712

*This research was supported in part by the Joint Services Electronics Program under Research Grant AFOSR 69-1792 and The Center for Highway Research.

ABSTRACT

The need for better traffic control techniques has significantly increased during the past few years because of the rapidly increasing numbers of vehicles in American cities. A modeling method for traffic simulation is presented which promises to be a valuable tool for providing this needed control. This traffic simulation method is somewhat different from the usual queueing theory approach in that traffic simulation is realized by stepping vehicles through the traffic network system in accordance with prespecified driver response criteria. Modern control theory was used in implementing this method, thus rendering it adaptable to standard control systems theory for traffic control applications. A computer program was developed for this model in which systems programming techniques were employed to minimize memory and computer requirements.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Widespread interest in better traffic control techniques has resulted in the past few years because of the rapidly increasing numbers of vehicles in American cities. A few cities including Austin, Texas have installed prototype computer controlled traffic signal systems. The control techniques employed by these cities are generally similar in that a predetermined traffic signal timing pattern is selected for use according to the time of day or in accordance with previously observed traffic conditions. Feedback control is not used as the basis for extending or terminating signal indications in real-time.

This study describes a modeling technique which promises to be useful for traffic simulation and on-line signal control. For this method, simulation is accomplished by stepping individual vehicles through a traffic network system in accordance with driver responses to changing traffic and environmental conditions. Since the simulation can be used to predict traffic conditions, on-line simulation can then possibly be used as criteria for the selections of traffic signal control patterns or other control techniques.

1.1 Purpose

It is the purpose of this study to develop a modeling method which can be used for implementing the step-through simulation technique.* Such a method and a computer program employing Boolean Algebra and

---

* The idea of using the step-through approach for traffic simulation studies for this thesis was suggested by Dr. C. E. Lee, Professor of Civil Engineering and Director of the Center for Highway Research, The University of Texas at Austin, Austin, Texas.

modern system programming techniques are presented. Traffic flow through a hypothetical intersection controlled by four-way stop signs is used to illustrate the method and to indicate the relative efficiency of the computer program. Results from a few randomly generated vehicles passing through the intersection are discussed optimistically.

The next logical procedure before using the step-through technique in traffic control systems should be comparing simulation results of this technique with actual traffic data. Such comparisons should be performed on the various most common intersection types. Upon the successful completion of such studies, the modeling method could be used for traffic simulation and control.

The modeling method developed for the traffic simulation and control problem is not confined to this problem only, but could be used in a variety of other simulation and control situations. One such example is its use in missile range control count-down simulation for range scheduling purposes.[1] The method also appears to have possibilities of being the basis of a problem oriented simulation language[2] or its use with such a language already in existence.

1.2 A Brief Summary of Current Simulation Techniques

The majority of traffic simulation models proposed up to now have been developed around queueing theory and statistics. Descriptions of many of these models are available in the literature. Reference 3 provides a good summary of some of the more pertinent studies. Reference 4 provides a description of the uses of queueing theory and cybernetics in feedback traffic control systems. These studies have considered traffic simulation for the various types of traffic intersections as well as for freeway ramp and car following studies.

Two simulation studies (References 5 and 6) were investigations in which the standard queueing theory approach was not employed. In these, each individual car was considered and the various characteristic information computed for it as it proceeded through the intersection. Reference 5 considered the two-way stop sign single intersection. Reference 6 provided car following simulation. In the programs used for the model, however, program complexity, length and computer time become undesirable for its use in solving the multiple intersection network problem. The model developed in this thesis was set up so that this step-through procedure could be easily implemented for various and multiple intersection types with minimum memory and computer time requirements.

CHAPTER II

MODEL DEVELOPMENT

A model is a representation of some form of reality, devised in an attempt to explain some aspect of the behavior of this reality. Thus, in synthesizing a model we must first obtain or model as many thoughts as possible pertinent to the particular representation desired. From the limited complexity of that part of reality which we are attempting to understand, we are able only to select a finite number of these abstractions or concepts. Hence our model can never replace or explain all possible behaviors of reality since first, our thoughts are but abstractions from reality, and secondly, these abstractions are only a finite subset of a possible infinite number.

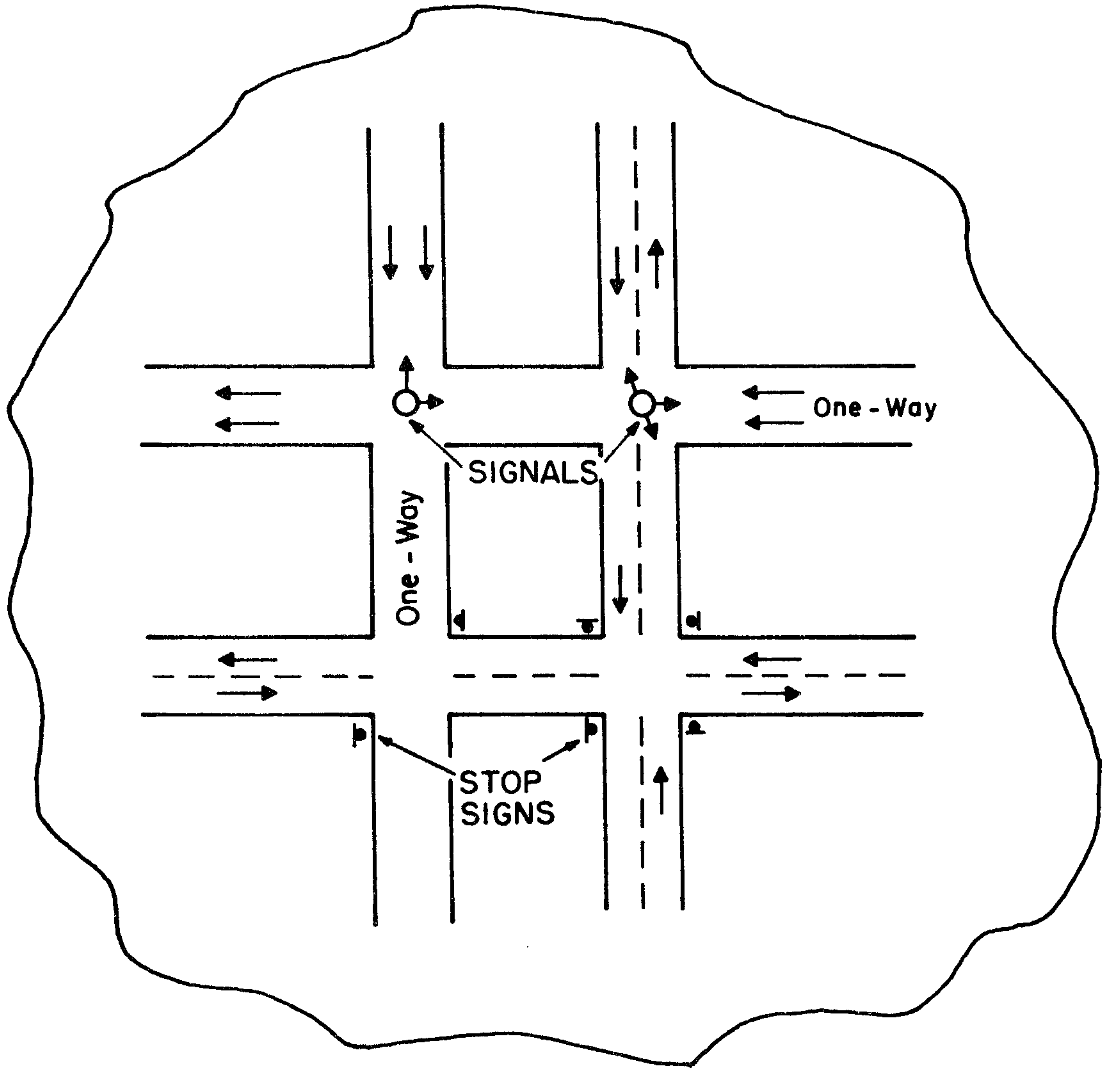
Before condemning model development, however, we must recall that it is currently the only technique available for simulating or predicting reality and has been used successfully throughout the gamut of disciplines. For this reason, it will be employed for predicting traffic behavior. One should, however, be continually aware of the limitations of models, the traffic network model described herein being no exception.

2.1 Traffic Network Model

As implied above, the usefulness of a model is dependent largely upon how well one has characterized the reality of interest. The development of traffic flow network simulators or models has taken various forms as noted in Chapter I. The technique employed in this thesis is somewhat different in that an attempt has been made to simulate or model the actions taken by the driver of each vehicle in the system as the vehicles move within a street network. This is accomplished by first prespecifying what the driver's actions will be in response to several sets of inputs that are available to him concerning the roadway and traffic situation. The driver actions are then determined for each particular combination and for each increment of time. Thus, by stepping each driver through the traffic network system, network traffic flow can be realized.

To formulate this technique the following approach was taken: first, a traffic network system was defined as a network of various types of traffic intersections (i.e., four-way stops, two-way stops, signal-controlled, etc.), each with one or more lanes per approach (see Fig. 2.1). Next, the set of n inputs required by the driver while stepping through the system and the set of m driver responses were defined as the driver input set and driver response set, respectively. The set of decision response functions was defined as that set of functions which associates with each of the possible 2^n evaluations of the input set a value from the driver response set; see Fig. 2.2. There are 2^n possible evaluations of the input set, as each input variable of the input set has only two possible values, i.e., yes or no. A second set of functions denoted as the intersection functions associates a particular set of driver inputs and driver responses to the physical characteristics of an intersection type. A set of intersection types completely describes a network system.

Some typical vehicle inputs might be: present speed, desired speed, and questions such as "Is the vehicle at the intersection," or "Is the vehicle stopped." Some typical driver responses are: stop vehicle, increase speed, decrease speed, turn right, turn left, etc. Since a particular driver response is determined by a set of yes/no driver input combinations, the response function may be expressed in a standard computer flow chart form. The network system could thus be expressed by all such intersection flow charts. Because of the way



TRAFFIC NETWORK SYSTEM

FIG. 2.1

VEHICLE CHARACTERISTICS $\Longrightarrow$ INTERSECTION CHARACTERISTICS $\Longrightarrow$ NETWORK CHARACTERISTICS

	DRIVER INPUTS	DRIVER RESPONSE	INTERSECTION TYPES	
	$X_1, \ldots X_N$	$F\ (X_1, \ldots X_N) \ldots F_M(X_1, \ldots X_N)$	$G_1\ (F_1, \ldots F_M) \ldots G_K(F_1, \ldots F_M)$	
2^n	$\vdots$	$\vdots$	$\vdots$	2^{n+m}
	$(X_1, \ldots X_N)$	$(R_1, \ldots R_M)$	$(I_1, \ldots I_M)$	
	$\vdots$	$\vdots$	$\vdots$	

TABLE OF MODEL COMBINATIONS

FIG. 2.2

the model variables were structured, the intersection and system network functions form a Two-valued Boolean Algebra. Thus, for example, each intersection type may be expressed as a sum of products (or its dual, a product of sums) of the driver input variables. A typical decision response function might be:

$$A = \bar{x}_1 \bar{x}_2 + x_3 x_4 \qquad (2.1)$$

where

$\bar{x}_1$ = no vehicle is in close proximity

$\bar{x}_2$ = not at desired speed

x_3 = vehicle at intersection

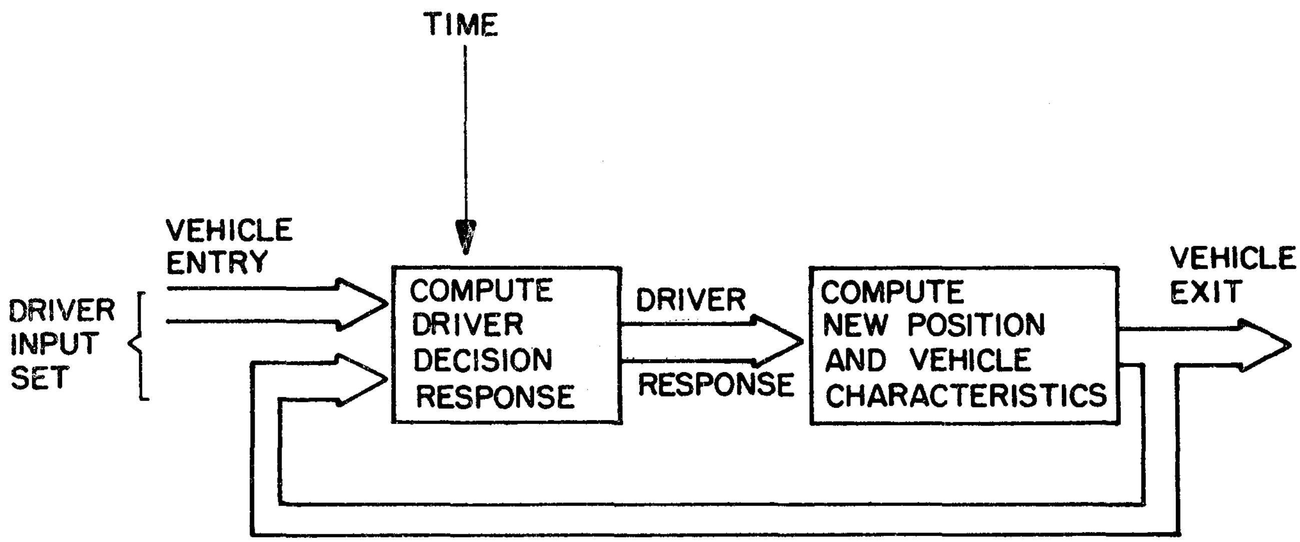
x_4 = first vehicle at intersection

A = accelerate vehicle

By expressing these functions in Boolean Algebra, the standard rules regarding Boolean simplification apply.

Figure 2.3 depicts symbolically the traffic network model. As may be noted in this figure, vehicles enter the network system at the specified speed, lane approach, and time. Then the appropriate response is determined; that is, accelerate, decelerate, etc., via the driver response functions. Once the proper responses have been determined, the vehicle's new position, speed, etc., are found. This operation continues as the vehicle is stepped through the network in accordance with the prespecified flow. When the vehicle reaches the edge of the network system it is logged out of the model. By observing this process for selected vehicle speeds, directions, and time combinations, traffic flow through the network model is realized.

In the section which follows, the four-way stop, single intersection network will be considered.



TRAFFIC NETWORK MODEL

FIG. 2.3

2.2 Four-Way Stop Single Intersection Network

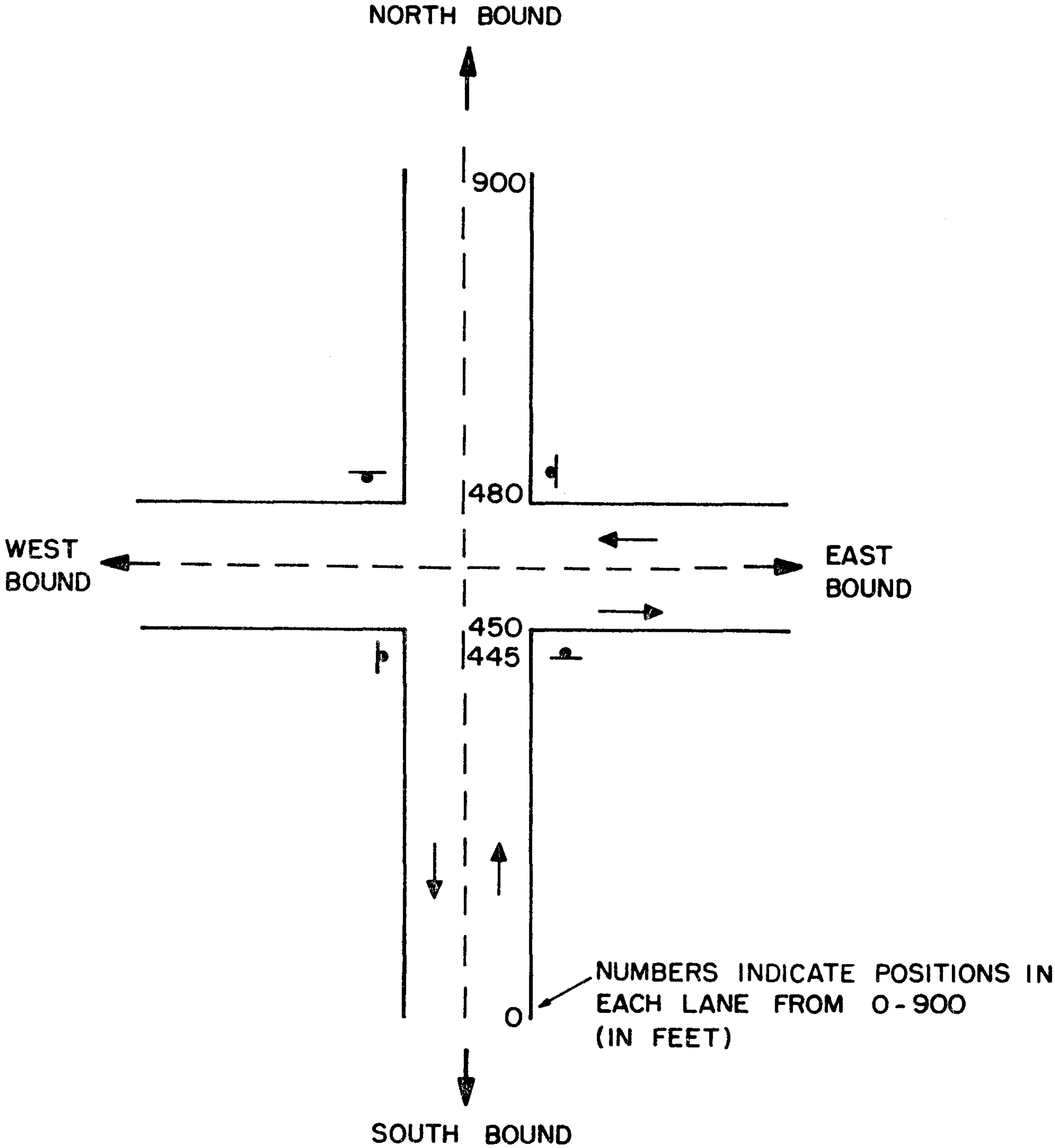
Consider a four-way stop, single lane, single intersection network as depicted in Fig. 2.4. For the particular system shown, vehicles enter the system 450 feet before reaching the near side of the intersection and exit 450 feet beyond this point. These dimensions are, of course, completely arbitrary and may be selected to describe a particular intersection of interest. The intersection accomodates one lane for each of the four directions - North, South, East, West - with stop signs on each approach lane at the intersection.

The four-way stop intersection, as the name implies, requires that all vehicles stop before entering the intersection. The decision to proceed into the intersection is then based on the particular traffic laws of the city or state where the intersection is located. The rules or laws selected for the four-way stop model being described are those recommended by the National Committee on Uniform Traffic Laws and Ordinances, and are similar to those applicable in the state of Texas. These rules may be stated as follows:[7]

a) "When two vehicles approach or enter an intersection from different highways at approximately the same time, the driver of the vehicle on the left shall yield the right-of-way to the vehicle on the right."

b) "The driver of a vehicle intending to turn left within an intersection or into an alley, private road or driveway shall yield the right-of-way to any vehicle approaching from the opposite direction which is within the intersection or so close thereto as to constitute an immediate hazard."

c) "Except when directed to proceed by a police officer or traffic-control signal, every driver of a vehicle approaching a stop intersection indicated by a stop sign shall stop at a clearly marked stop line, but if none, before entering the crosswalk on the near side of the inter-



FOUR-WAY STOP INTERSECTION

FIG. 2.4

section or, if none, then at the point nearest the intersecting roadway where the driver has a view of approaching traffic on the intersecting roadway before entering the intersection. After having stopped, the driver shall yield the right-of-way to any vehicle which has entered the intersection from another highway or which is approaching so closely on said highway as to constitute an immediate hazard during the time when such driver is moving across or within the intersection."

These rules are used in the model to govern the driver as he steps through the intersection. By flow-charting the driver's actions as he transverses the network system, the decision response flow chart is generated. In developing this chart, the vehicle input and response sets are defined. Figure 2.5 depicts the decision response flow chart for the four-way stop. Table 2.1 lists the necessary driver inputs and driver responses used by the driver as he transverses the network system. As noted, only three responses are actually required by the driver, namely, increase, decrease, or maintain speed.

To stop the vehicle, the vehicle speed is simply decreased until its speed is zero. Similarly, when the vehicle is stationary, movement is initiated by increasing velocity.

Turning movements are not considered responses by the model but part of the predetermined vehicle path. That is, when a vehicle enters the system, the driver destination has been prespecified as a left turn, right turn, or straight through movement. Then, as the vehicle is stepped through the network, if a turn is specified for that vehicle, it is automatically initiated as he follows the appropriate movement flow path.

Also indicated in both the vehicle input set and the decision response flow chart are the inputs, vehicle less than, greater than, or equal to critical distance. The critical distance criterion is perhaps the most sensitive of all vehicle characteristics and certainly one of the more

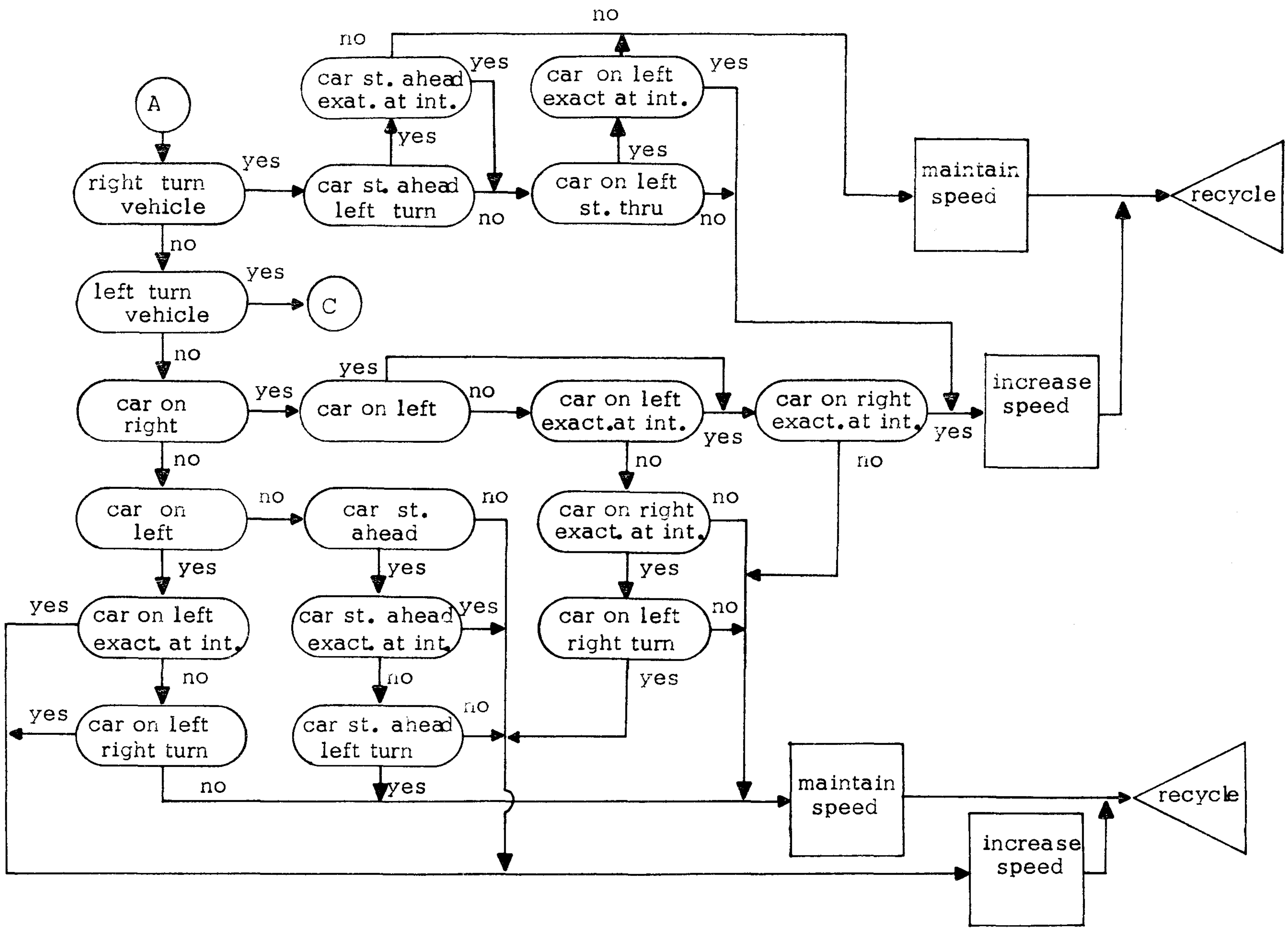

A
right turn vehicle
car st. ahead left turn
car st. ahead exat. at int.
car on left st. thru
car on left exact at int.
maintain speed
recycle
left turn vehicle
C
car on right
car on left
car on left exact.at int.
car on right exact.at int.
increase speed
car on left
car st. ahead
car on right exact. at int.
car on left exact. at int.
car st. ahead exact. at int.
car on left right turn
car on left right turn
car st. ahead left turn
maintain speed
recycle
increase speed

TABLE 2.1

VEHICLE INPUT AND RESPONSE FOR FOUR-WAY STOP

Driver Inputs		Driver Responses
Distance less than critical distance		Increase speed
Distance equal to critical distance		Decrease speed
Distance greater than critical distance		Maintain present speed
Vehicle in intersection		
Vehicle exactly at intersection		
First vehicle exactly at intersection		
Right turning vehicle		
Left turning vehicle		
Straight thru vehicle		
Vehicle at desired speed		
Vehicle less than desired speed		
Vehicle stopped		
Car on right		
Car on left		
Car straight ahead		
Car on right exactly at intersection		
Car on left exactly at intersection		
Car straight ahead exactly at intersection		
Car on right	right turning	
Car on right	left turning	
Car on right	straight thru	
Car on left	right turning	
Car on left	left turning	
Car on left	straight thru	
Car straight ahead	right turning	
Car straight ahead	left turning	
Car straight ahead	straight thru	

difficult ones to model realistically. The term critical distance as used in the model is defined as that distance which is required to stop or slow the vehicle sufficiently to prevent it from colliding with an adjacent vehicle or object. Thus, when the input, vehicle less than critical distance, is true, the model should slow the vehicle. The amount of slowing or decelerating is important since insufficient deceleration will result in a collision, and too much might result in oscillation about the proper critical distance as the driver tends to over-correct.

Appendix 2 provides a description of the critical distance criterion and associated deceleration rates. Briefly, the critical distance function was obtained by first establishing a relation between acceptable deceleration rates and vehicle velocity. Once this function was found, it was then used to express the distance required to stop the vehicle as a function of the vehicle velocity.

Vehicle acceleration is also important although not as critical as deceleration since the risk of a collision is very small for reasonable rates of acceleration. For the model, vehicle acceleration is always assumed fixed at 2 ft/sec^2; however, it can be related according to some other criterion.

An accurate traffic model will be quite sensitive to these acceleration, deceleration, and critical distance rates. To obtain better relations for these rates, one should compare predicted and actual traffic flows. Regression analysis or some similar technique can then be used to obtain more realistic relations for the class of traffic being simulated.

The model operation is as depicted symbolically in Fig. 2.3. Descriptive characteristics for a set of vehicles is first generated by specifying for each vehicle an entry time, speed, lane approach, and vehicle flow path. This set may be determined either randomly with certain characteristics such as turning direction or lanes biased according to some observed or preestablished probabilities, or by other such means. Then at the appropriate time each vehicle is queued into the system model. Upon entry, the vehicle joins the other vehicles already in the system and the proper vehicle input set is generated. For each time increment, each vehicle is sequentially examined, first determining the appropriate decision response from the vehicle input set in accordance with the decision response flow chart of Fig. 2. 5, and then, as directed by this response, updating the vehicle input set.

Chapter III describes the program written to characterize this model. Chapter IV then discusses some results of using this model for simulating traffic flow behavior for the four-way stop single intersection network with 100 input vehicles.

2. 3 Multiple Intersections

The network system described in the preceding section was for the four-way stop single intersection network. The procedure used to model the multiple intersection network of k intersection types is similar to that just described; that is, a set of k decision flow charts and the respective vehicle input and response sets are developed. Vehicle entry information, however, must include, in addition to the entry information previously described, the entry intersection and an expanded data flow path which specifies the intersection combination sequence. Then for the model operations, the vehicle is simply stepped from one intersection to the next as prescribed by the data flow path until it leaves the network system. Hence, the multiple intersection model is simply composed of multiple single intersection models or subsystems.

CHAPTER III

PROGRAM DESCRIPTION

The basic concepts for the traffic network model were discussed in Chapter II. This chapter describes the computer program developed to realize these concepts. The program was written in the FORTRAN language for execution on a Control Data Corporation 6600 computer. Once a model has been developed, the implementation or realization of such a model by a computer program can be difficult, if due consideration is given to program costs. Program costs are directly proportional to such requirements as memory size, computer time, input/ output, and program usage. For a multi-processing environment characteristic of the CDC 6600 computer system, additional consideration must be given to program throughput. Typical of all such environments, storage needs should be minimized to permit a more adaptable program for loading requirements. It should also be noted that core memory is almost always a sensitive constraint with the increased usage of the "mini" computer in control systems.

In consideration of the above points, the traffic network program described in this chapter was developed in accordance with the following criteria and in the order specified:

(a) program flexibility

(b) minimum memory requirements

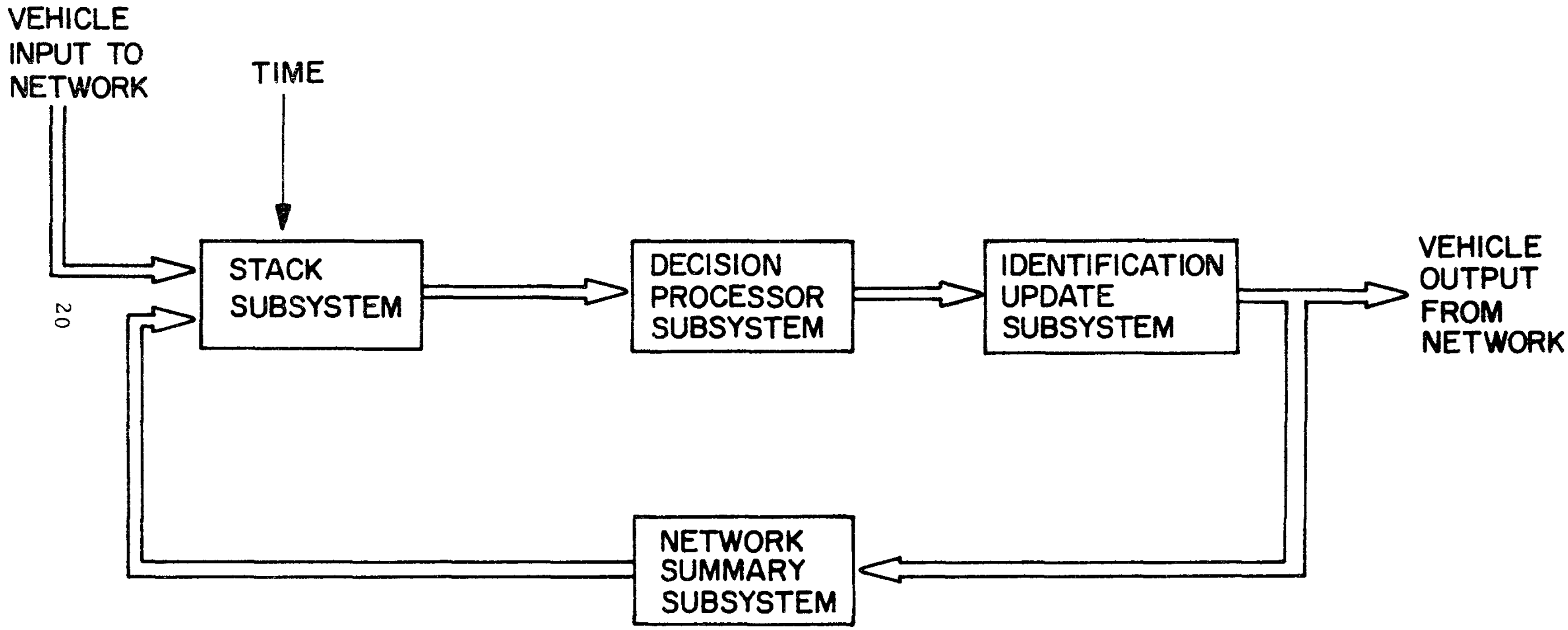
(c) program structure and usage

(d) minimum central processor or raw computer requirements.

3.1 Program Structure

The program structure is similar to the model structure depicted in Fig. 2.3. Figure 3.1 illustrates the general basic traffic network program. As shown in this figure, there are four major subsystems,

TRAFFIC MODEL
NETWORK SYSTEM
FIG. 3.1



the vehicle stack, the decision processor, the vehicle identification update, and the network summary (the term subsystem and subroutine are used synonymously throughout this chapter).

The stack is used to maintain all vehicle identification. It contains primarily two word types for each object (vehicle, stop sign, or input/output dummies) in the system. The first word type, vehicle parameter identification, contains characteristic information of the vehicle needed for computing position, speed, direction, etc. The second word type, vehicle variable information, contains all the Boolean input variables as required in computing the decision response functions. Thus, the vehicle variable information word contains the vehicle input set described in Chapter II. The decision processor subsystem computes the proper driver responses from the Boolean decision response functions and the variable information words (vehicle input set).

The vehicle identification update subsystem uses the response set (consisting of only one element for the four-way stop, i.e., increase, decrease, or maintain speed) to compute the vehicle's new position, speed, and direction. The new set of vehicle input variables is then generated by this subsystem in accordance with the new computed parameters. If a vehicle's new position falls outside the system, the vehicle is taken from the stack and its exit from the model is noted.

The network summary subsystem provides selectable vehicle summary printout for either individual or gross summary data. The four subsystems are thus utilized in the program as each vehicle is processed for each time increment by the program.

As indicated above, the program structure was developed in an attempt to maintain the four constraints of flexibility, memory, usage, and computing requirements. Program flexibility was considered to be of prime importance since a useful program must be adaptable to multiple vehicles with various characteristics traveling through multiple and dissimilar intersections. Because of this, the input and decision response set concept developed in Chapter II was employed. Each new intersection type or vehicle type is specified completely by these two sets. Solution of the Boolean decision response functions are then easily found by the AND computer operators.[8] Minimum memory requirement is maintained by using single words to contain several characteristic parameters and the vehicle input information (see Program Tables in Appendix 3). This was accomplished somewhat at the expense of computing time as shifting in FORTRAN is achieved by the $2^{\pm n}$ computation. However, an assemble language routine for shifting would render this expense negligible, thereby considerably reducing present computer time requirements. In addition to reducing memory needs by so specifying the many vehicle input variables in one word, the computation of the logical decision response functions required very few instructions (see decision processor subsystem description and Appendix 4). Ease in program usage was maintained by the procedure used to input the vehicle characteristics and the decision response functions. The decision response functions are read into the program in the standard Boolean sum product representation (see Appendix 1). Minimum computing requirements were then observed after the above items were exploited.

Following is a detailed description of the four primary subsystems in which these constraints become more apparent.

3.2 Stack

As indicated above the stack is used to maintain vehicle identification and consists of two word types, vehicle characteristic information and vehicle input information. For each vehicle in the system two entries of each word type are made. Appendix 3, Table A3. 1, provides the information contained in each entry of each word type. As specified in this table, IVCH and IVCH1 contain the vehicle parameter information, i. e., speed, position, etc. IVEH and IVEH1 contains the vehicle input information. As noted in the vehicle input words, IVEH and IVEH1, two bits are used to specify each input variable. These bits are used as follows:

00 = not applicable

01 = variable true

10 = variable not true

11 = not used.

As will be discussed in the Vehicle Identification Update Section, the stop signs, and dummy input/output objects also contain stack entries.

Four stack entry types are used in the model, namely the vehicle entry, the stop sign entry, the intersection input entry and the intersection output entry (dummy input/output objects).

Each entry has the above two word types assigned to it, although only the vehicle entry has need for all the information contained in these words. The other three entries use primarily the object position parameter and the nearest object index (see Vehicle Identification Update Section). The stop sign entry is used to indicate the stop sign location, i. e., position and approach. The intersection input entry, referred to as a dummy object, is used primarily so that the proper nearest object can be assigned to the proper vehicle as it enters the intersection. The intersection output entry is also referred to as a dummy object and is used to indicate the end of the intersection for system exit operations or in the multiple intersection problem, for entry into the next intersection.

3.3 Decision Processor

The decision processor subsystem computes the proper driver response in accordance with the decision response functions. As discussed in Chapter II, an intersection is described by the set of driver inputs and driver responses which are characteristic of a particular intersection. The decision processor examines all input combinations associated with a particular response, and if the proper combinations of inputs are true, the associated response is indicated. Since each decision response function is a Boolean function, the following technique was employed to process this response:

Each driver response is first expressed in a sum product form. Two computer words are then used to contain each product term, where each variable is expressed by a two bit code as follows:

00 = "don't care"

01 = true condition

10 = not true condition

11 = not used.

The set of all such word product terms then describes the response function and all such sets describe the intersection type.

Each single product term or two word group is then ANDed with the input set. If the resulting ANDed word group is identical to the product term, the response is set true. Accordingly, if all terms are not true, the response is set false.

This operation continues until all response functions have been examined. The Boolean sum product functions are read in as system decision response data cards, one product term per card. Appendix 4 provides the input card images for the four-way intersection. As may be noted from these cards, the response number corresponding to the bit location within the response word (used by the program) is specified for each product term. All card images containing the same response number constitute the total sum product Boolean expression. Figure 3.2 depicts the first two card images corresponding to the distance less than critical distance flow path of Fig. 2.4 in Chapter II. The two Boolean terms expressed by these card images may be written for the increase speed response as

$$\bar{x}_2 \, \bar{x}_3 \, x_6 \tag{3.1}$$

and for decrease speed response as

$$\bar{x}_2 \, \bar{x}_3 \, \bar{x}_6 \tag{3.2}$$

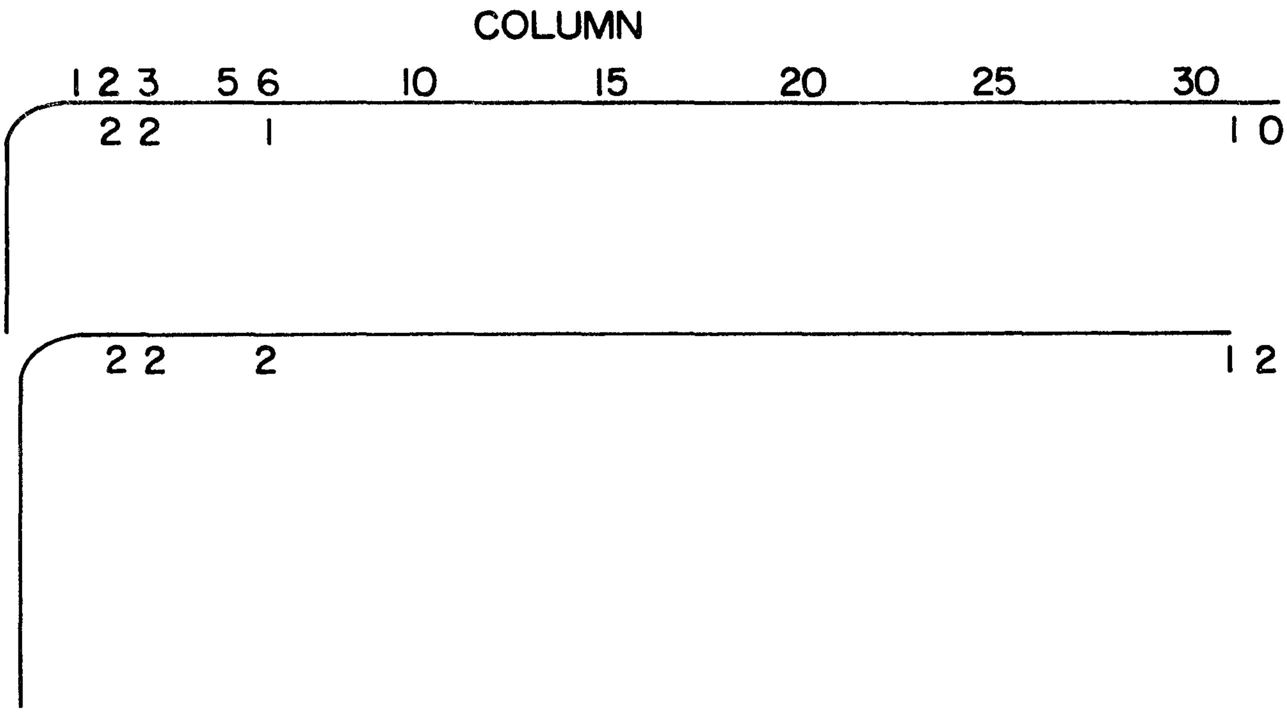
where

x_2 = vehicle at desired speed

x_3 = vehicle's speed greater than desired speed

x_6 = vehicle in intersection

The 2's in columns two and three and the 1 in column six represent equation (3.1). The 10 in columns thirty-one and thirty-two specify the increase speed response. Similarly, the 2 in column six and 12 in columns thirty-one and thirty-two of the second card represent the decrease speed term of equation (3.2) and, in fact, for this case, the total decrease speed Boolean expression. The increase speed term, equation (3.1), was necessary to initiate movement from the stopped position five feet in front of the stop sign (near side of the intersection) or nearest object. **Starting movement was initiated in this way since, for the four-way stop single intersection example, all cars accelerated at the same rate. Hence, the only time the distance to the nearest object is less than the critical distance for the intersection is when the nearest object is a stop sign.** These responses are set only when either



DECISION RESPONSE DATA CARDS

FIG. 3.2

equation (3.1) or (3.2) is true. The decision processor subsystem thus compares each of the system decision response function cards with the vehicle input variables and sets the proper response.

3.4 Vehicle Identification Update

The identification update subsystem is the largest of the four subsystems. It performs the following five major functions:

(1) Computes the new vehicle speed and position in accordance with the specified response and proper acceleration/deceleration value;

(2) Determines the nearest object;

(3) Places the vehicle on the proper lane if turning has been specified; and, if the vehicle is at the intersection, determines the new nearest object, if necessary;

(4) Computes the new critical distance;

(5) Updates the vehicle characteristic word and vehicle input word (new driver input set).

The identification update subsystem computes the new vehicle speed according to the standard speed acceleration equation; i.e.,

$$s = s_o + a\Delta t \qquad (3.3)$$

where

s = new vehicle speed

s_o = old vehicle speed

a = acceleration

Δt = time increment

The acceleration, a, is zero for the maintain speed response, 2 ft/sec^2 for the increase speed response, and

$$[+(6./400.)\,d - 6]\,\text{ft/sec}^2, \qquad (3.4)$$

for the decrease speed response, where d = distance to nearest object.

Once the vehicle's new speed has been determined, the standard equation for computing position is used for determining the new vehicle position, i.e.,

$$p = p_o + s\,\Delta t \tag{3.5}$$

where

p = new vehicle position

p_o = old vehicle position

s = vehicle speed

Δt = time increment

To determine the vehicle's position relative to the other vehicles, as well as to the stop signs within the system, the nearest object's stack index is continually updated by each stack entry. As each vehicle is processed by the identification update subsystem, the nearest object index is extracted from the vehicle characteristic information word, and the distance to this object computed. Hence, each object (vehicle, stop sign or dummy input/output object) continually keeps track of the nearest object immediately ahead.

Passing may be accomplished as follows (for the four-lane intersection example only stop signs and dummy input/output objects are passed):

Let A be a vehicle or dummy output object

B be a stop sign

C be a vehicle and

D be a vehicle or dummy input object where the initial positions are $A > B > C > D$. Then, in order for vehicle C to pass stop sign B,

(i) the nearest object A of B is found and assumed by C as its new nearest object;

(ii) B is given C as its new nearest object;

(iii) the pass indicator in vehicle C's characteristic information word and the index of the vehicle's old nearest object index B ("Pass" and "Object Index for Pass" entries, IVCH, Appendix 3) are set.

When D is processed, its nearest object is found by first checking for the pass flag of its old nearest object C. If this flag is set, the new object index, B, is extracted from the proper entry of the characteristic information word and used as the new nearest object. The pass flag is then cleared.

When passing a dummy input object, the same procedure is used, where A is a stop sign or a vehicle, B is the input object, and C is the vehicle (D is not used). As the nearest object of B is found, the pass flag is always checked first. When exiting from the intersection, that is passing the dummy output object, the above procedure is once again employed. In this case A is the dummy output object, B the vehicle, and C a vehicle or stop sign. The vehicle exiting is kept in the stack (although no longer processed) until the pass indicator is cleared or the object that was directly behind it, C, extracts the proper nearest object index, A.

Turning movements are initiated in the model when the vehicle passes the stop sign in the intersection and, at such time, the following procedure is used:

(i) The stop sign on the vehicle's right (left), depending on whether a left (right) turn had been indicated, is found and its nearest object determined.

(ii) This nearest object is tnen used as the new nearest object for the turning vehicle and this vehicle in turn is selected as the nearest object of the stop sign.

(iii) The pass flag and the vehicle's old nearest object are set in the pass procedure above. It should be noted, however, that in this instance two objects, the stop sign and the object that was immediately behind the vehicle, are both referencing the same nearest object, that is, the vehicle. Thus, the stop sign's ignore pass indicator in its characteristic information word IVCH, Appendix 3) is set so that the other referencing object may obtain the correct nearest object.

(iv) The turning vehicle is placed in the proper lane five feet behind the end of the intersection, if a right turn was indicated, or ten feet in front of the beginning of the intersection, if a left turn was indicated.

Objects on the right, left, or straight ahead of a vehicle are determined for the turning process and for updating the vehicle input information word (driver input set) as follows:

The four lane directions are indicated by 2 bits, where

00 = 0 = North bound

01 = 1 = East bound

10 = 2 = South bound

11 = 3 = West bound.

Table 3. 1 lists the four directions and the respective immediate right, left, and straight ahead directions. Equations (3. 6) - (3. 8) below are used to relate these immediate directions to the vehicle's direction as specified in the table.

TABLE 3.1

VEHICLE - DIRECTION RELATIONSHIPS

Direction of			
Vehicle	Immediate Object		
	On right	On left	Straight Ahead
00 (North)	01	11	10
01 (East)	10	00	11
10 (South)	11	01	00
11 (West)	00	10	01

Object on right:

$$d_v = (d_r + 1)\ .AND.\ 3 \tag{3.6}$$

Object on left:

$$d_l = (d_v + 1)\ .AND.\ 3 \tag{3.7}$$

Object straight ahead:

$$d_v = \begin{cases} (d_{sa} + 2)\ .AND.\ 3, & d_{sa} \leq 2 \\ (d_{sa} - 2)\ .AND.\ 3, & d_{sa} = 3 \end{cases} \tag{3.8}$$

where

d_v	=	Direction of vehicle
d_r	=	Direction on right
d_l	=	Direction on left
d_{sa}	=	Direction straight ahead
.AND.	=	Boolean logical AND operator

Computation of the new critical distance is found for each vehicle for each time increment. The critical distance is used by the decision response functions (see Fig. 2.4) to increase, decrease or maintain speed, depending on whether the distance to the nearest object is greater than, less than, or equal to the critical distance, respectively. The critical distance, acceleration and deceleration criteria are closely related as described in Appendix 2 and are probably the most sensitive of all criteria used in the model. Consequently, these criteria are the most likely to be used for adjustment to improve correlations between actual and predicted traffic flow. Hence the computations of these criteria are treated as separate independent subroutines in the program. The critical distance function for the example is computed from the ellipse equation (see Appendix 2).

$$C_o = -(A/B) * (B^2 - V^2) + H \tag{3.9}$$

where

C_o = critical distance

V = relative velocity between vehicle and nearest object

A = 395

B = 48

H = 400

The vehicle characteristics and the input words in the stack are updated according to the new position, speed and model environmental conditions (positions of other objects in the system). These words are then used for processing during the next time cycle.

3.5 Network Summary

The Network Summary subsystem is entered once each cycle for each vehicle. Essentially it is used to provide selectable inprocess summary information for each vehicle or gross summary data for all vehicles during selectable time periods. The information provided for individual vehicles is as follows:

(i) simulated model time,

(ii) stack entry index,

(iii) vehicle ident (selectable vehicle identification number from 0-511),

(iv) position in lane (feet),

(v) direction as defined in previous section,

(vi) turn type (0 = straight thru vehicle, 1 = right turning vehicle, and 3 = left turning vehicle),

(vii) velocity (mph),

(viii) IVEH and IVEH1 stack words as defined in Appendix 3,

(ix) IVCH and IVCH1 stack words as defined in Appendix 3,

(x) IY, driver response word, as defined in Appendix 3.

Gross information at selectable time increments provides the following summary printout by approach lanes:

(i) simulated model time.

(ii) average stop time delay for each movement (i. e., straight through, right or left) by turning movements.

(iii) average time delay for vehicles equal to or less than x (selectable speed) mph for each movement.

(iv) number of vehicles in system for each movement.

(v) average time in system for each movement.

In addition to the above information the program prints out several other individual and gross information types, although not in the network summary routine. However, since this information is summary data, it is presented in this section. This information is as follows:

1. Vehicle input card image. An exact duplicate of the vehicle input card image is provided each time an object is considered for entry by the model. Accompanying this input information is the time increment for model termination (see Appendix 1) and the time the object is brought into the model for entry consideration. (Note that the object is not entered into the stack until its entry time is reached.)

2. Vehicle exit information. If this printout option is selected, each time a vehicle leaves the system the following information is provided:

(i) vehicle identification,

(ii) stack index,

(iii) movement type as defined above,

(iv) direction as defined above,

(v) time of exit from the system.

3. Grand total summary information. At model termination, the grand total of the gross summary information described above is provided if specified. In addition to this information, the actual time the system was used is also provided.

3.6 Other Subsystems

Several other subroutines are used in the program but have not been described. These routines were written only for ease in program development and will not be described here. Appendix 3 provides a short description of all routines in the program.

3.7 Multiple Intersection Considerations

As has been indicated, the current program accommodates only the single intersection system. It can, however, be modified easily to solve the multiple intersection problem. The necessary changes should be made in the vehicle exit processing, beginning with statement number 40, in CHAR, Appendix 5. At this point, the vehicle position and lane identification should be changed to place the vehicle in the proper intersection. Of course a check should be included to see if all intersections specified have been entered and, if so, the normal system exit taken. The changes necessary for multiple intersection types are more extensive. However, the basic structure of the program was developed for the multiple intersection problem and these changes should be the add on type only, much in the same manner that additional stories are added to a building. Primarily, the stack information words must be augmented to include any additional parameters and driver input variables that are required for the particular intersection types (see multiple intersection description, Chapter II). Then, the necessary logic must be added to update this information in the vehicle identification update subsystem.

CHAPTER IV

APPLICATION IN TRAFFIC SIMULATION

The program model developed and described in Chapters II and III can be used for traffic flow simulation studies for the single lane, single intersection four-way stop. Modifications to the program will make it possible to accommodate the multiple intersection problem. This chapter describes the use of the program for traffic flow simulation through a four-way stop intersection in which 100 randomly generated vehicles are entered into the system over a 23 minute time period. By following the program usage instructions described in Appendix 1, any desired vehicle input combination can be specified for traffic simulation studies.

Several example problems were run during program checkout, one of which will be described in this section. A short program was written in which the required characteristics of 100 vehicles were generated. As noted from the program usage instruction, Appendix 1, several parameters are required for each vehicle in order that they may be properly queued into the system. Table 4.1 lists these parameters and the selection technique used in obtaining each of them. As indicated in the table and as may be noted in the summary information discussed later, the direction selection was slightly biased due to the technique employed. As a result of this bias more vehicles were selected for the northbound approach than for the other directions. This, however, is quite realistic as in many cases one approach often accommodates more vehicles than the others. This is easily confirmed by observing early morning and five o'clock traffic conditions. The 100 vehicles were entered into the system over a 23 minute time period. The 100 vehicles used and the accompanying parameters are listed in Appendix 4 as input

TABLE 4.1

VEHICLE SELECTION TECHNIQUE

	Parameter	Selection Technique
1.	Entry position	Constant at 0 feet
2.	Vehicle speed	Random
3.	Desired speed	Constant at 30 mph
4.	Lane direction	Random, northbound biased
5.	Nearest object index	Same as lane direction
6.	Vehicle length	Alternate selection between 18 and 22 feet
7.	Object type	Constant at 0, for vehicle random
8.	Movement type	Random
9.	Indent	Sequentially ordered
10.	Time entering system	Random monotonically increasing

card images. The columns of information are in the same order as listed in Table 4.1. Thus, for instance, the first vehicle with an Ident of 1 entered the system at a time of 12.3 seconds on a westbound approach at a velocity of 6 miles per hour. Similarly, the last vehicle with an Ident of 100 entered the system at 1302.4 seconds on a northbound approach at 27 miles per hour. Table 4.2 provides the results of the simulation run. Only the vehicle exit information and the grand total printouts were selected. As noted, the first vehicle left the system at 53 seconds and was under 10 miles per hour for 12 seconds. Similarly, the last vehicle left the system at 1336 seconds. The model terminated at 1403 seconds, approximately 100 seconds after the last vehicle entered the system (as specified in the data cards). The last vehicle traveled under 10 miles per hour for only 8 seconds, because of its faster entry velocity. It exited on the westbound lane after turning left off the northbound approach at the intersection. The grand total printout provides a summary of the number of vehicles counted into the system, the average stop time delay, the average delay below 10 miles per hour, and average time the vehicles were in the system. This information, as may be noted, is given by lane approach and turning movements. The actual time the system was used, 1275 seconds, is also provided for intersection study purposes.

The real time to computer time ratio for this particular problem was approximately 25 to 1. As noted in Chapter III, this ratio can be significantly increased by using an assembler (COMPASS) language shift routine (possibly to as high as 150 to 1). It should also be noted that by the nature of the program technique, computer time requirements are primarily dependent on the cycle time selected (0.5 second for this example) and the number of vehicles in the system at a given time. It is relatively insensitive to the number of intersections or to the intersection type.

TABLE 4.2

RESULTS OF COMPUTER RUN, 100 VEHICLES

```
VEHICLE LEFT SYSTEM     IDENT=     1 ISI=    13 TURN TYPE= 0 DIRECTION= 3 TIME=  53.00000
TIME IN SYSTEM=  40.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  12.00000

VEHICLE LEFT SYSTEM     IDENT=     2 ISI=    14 TURN TYPE= 1 DIRECTION= 0 TIME= 118.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM     IDENT=     3 ISI=    15 TURN TYPE= 1 DIRECTION= 3 TIME= 135.50000
TIME IN SYSTEM=  37.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM     IDENT=     4 ISI=    13 TURN TYPE= 0 DIRECTION= 0 TIME= 161.50000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   7.00000

VEHICLE LEFT SYSTEM     IDENT=     5 ISI=    14 TURN TYPE= 0 DIRECTION= 2 TIME= 179.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM     IDENT=     6 ISI=    16 TURN TYPE= 3 DIRECTION= 1 TIME= 197.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM     IDENT=     7 ISI=    17 TURN TYPE= 1 DIRECTION= 0 TIME= 211.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM     IDENT=     8 ISI=    18 TURN TYPE= 0 DIRECTION= 0 TIME= 221.00000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM     IDENT=     9 ISI=    13 TURN TYPE= 0 DIRECTION= 3 TIME= 227.00000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM     IDENT=    10 ISI=    19 TURN TYPE= 0 DIRECTION= 2 TIME= 237.50000
TIME IN SYSTEM=  36.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM     IDENT=    11 ISI=    14 TURN TYPE= 0 DIRECTION= 2 TIME= 253.00000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM     IDENT=    12 ISI=    15 TURN TYPE= 0 DIRECTION= 2 TIME= 272.50000
TIME IN SYSTEM=  36.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM     IDENT=    13 ISI=    17 TURN TYPE= 3 DIRECTION= 3 TIME= 282.50000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM     IDENT=    14 ISI=    14 TURN TYPE= 1 DIRECTION= 2 TIME= 300.50000
TIME IN SYSTEM=  41.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  13.00000

VEHICLE LEFT SYSTEM     IDENT=    15 ISI=    13 TURN TYPE= 1 DIRECTION= 0 TIME= 318.00000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000
```

```
VEHICLE LEFT SYSTEM    IDENT=   16 ISI=   15 TURN TYPE= 1 DIRECTION= 0 TIME= 330.50000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM    IDENT=   17 ISI=   18 TURN TYPE= 1 DIRECTION= 1 TIME= 339.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   18 ISI=   19 TURN TYPE= 0 DIRECTION= 2 TIME= 345.00000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM    IDENT=   19 ISI=   20 TURN TYPE= 3 DIRECTION= 2 TIME= 358.50000
TIME IN SYSTEM=  42.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  13.00000

VEHICLE LEFT SYSTEM    IDENT=   20 ISI=   13 TURN TYPE= 0 DIRECTION= 2 TIME= 384.50000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   21 ISI=   14 TURN TYPE= 0 DIRECTION= 2 TIME= 394.50000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   7.00000

VEHICLE LEFT SYSTEM    IDENT=   22 ISI=   16 TURN TYPE= 0 DIRECTION= 0 TIME= 398.50000
TIME IN SYSTEM=  36.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   23 ISI=   19 TURN TYPE= 0 DIRECTION= 3 TIME= 404.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   24 ISI=   13 TURN TYPE= 1 DIRECTION= 1 TIME= 438.00000
TIME IN SYSTEM=  44.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  17.00000

VEHICLE LEFT SYSTEM    IDENT=   25 ISI=   15 TURN TYPE= 0 DIRECTION= 2 TIME= 491.00000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM    IDENT=   26 ISI=   17 TURN TYPE= 0 DIRECTION= 0 TIME= 494.00000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM    IDENT=   27 ISI=   18 TURN TYPE= 0 DIRECTION= 0 TIME= 501.00000
TIME IN SYSTEM=  36.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   28 ISI=   14 TURN TYPE= 1 DIRECTION= 3 TIME= 517.00000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   7.00000
 COLLISION   ISI=       19INA=      14

VEHICLE LEFT SYSTEM    IDENT=   29 ISI=   16 TURN TYPE= 3 DIRECTION= 3 TIME= 537.50000
TIME IN SYSTEM=  36.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   30 ISI=   14 TURN TYPE= 0 DIRECTION= 3 TIME= 555.00000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000
```

```
VEHICLE LEFT SYSTEM    IDENT=   31 ISI=   17 TURN TYPE= 0 DIRECTION= 0 TIME= 560.50000
TIME IN SYSTEM=  39.00000 STOP TIME DELAY=    6.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  13.00000

VEHICLE LEFT SYSTEM    IDENT=   32 ISI=   19 TURN TYPE= 3 DIRECTION= 2 TIME= 566.00000
TIME IN SYSTEM=  42.00000 STOP TIME DELAY=    6.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  18.00000

VEHICLE LEFT SYSTEM    IDENT=   33 ISI=   16 TURN TYPE= 3 DIRECTION= 3 TIME= 581.50000
TIME IN SYSTEM=  38.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM    IDENT=   34 ISI=   15 TURN TYPE= 1 DIRECTION= 1 TIME= 597.50000
TIME IN SYSTEM=  38.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   35 ISI=   14 TURN TYPE= 3 DIRECTION= 3 TIME= 605.50000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM    IDENT=   36 ISI=   13 TURN TYPE= 0 DIRECTION= 0 TIME= 623.50000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   7.00000

VEHICLE LEFT SYSTEM    IDENT=   38 ISI=   17 TURN TYPE= 3 DIRECTION= 3 TIME= 642.50000
TIME IN SYSTEM=  36.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   37 ISI=   16 TURN TYPE= 1 DIRECTION= 0 TIME= 650.00000
TIME IN SYSTEM=  43.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  15.00000

VEHICLE LEFT SYSTEM    IDENT=   39 ISI=   14 TURN TYPE= 0 DIRECTION= 0 TIME= 659.00000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM    IDENT=   40 ISI=   13 TURN TYPE= 0 DIRECTION= 1 TIME= 669.50000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   7.00000

VEHICLE LEFT SYSTEM    IDENT=   41 ISI=   18 TURN TYPE= 0 DIRECTION= 2 TIME= 683.00000
TIME IN SYSTEM=  36.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   42 ISI=   15 TURN TYPE= 0 DIRECTION= 0 TIME= 688.50000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   43 ISI=   16 TURN TYPE= 3 DIRECTION= 2 TIME= 702.50000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   44 ISI=   14 TURN TYPE= 0 DIRECTION= 0 TIME= 714.50000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   45 ISI=   18 TURN TYPE= 3 DIRECTION= 3 TIME= 730.50000
TIME IN SYSTEM=  36.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   46 ISI=   15 TURN TYPE= 1 DIRECTION= 0 TIME= 739.50000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000
```

```
VEHICLE LEFT SYSTEM     IDENT=   47 ISI=   19 TURN TYPE= 3 DIRECTION= 3 TIME= 745.50000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM     IDENT=   48 ISI=   14 TURN TYPE= 3 DIRECTION= 3 TIME= 764.00000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM     IDENT=   49 ISI=   17 TURN TYPE= 1 DIRECTION= 1 TIME= 776.00000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM     IDENT=   50 ISI=   13 TURN TYPE= 0 DIRECTION= 0 TIME= 794.00000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM     IDENT=   51 ISI=   18 TURN TYPE= 1 DIRECTION= 0 TIME= 804.50000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM     IDENT=   52 ISI=   15 TURN TYPE= 0 DIRECTION= 3 TIME= 820.50000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   7.00000

VEHICLE LEFT SYSTEM     IDENT=   53 ISI=   19 TURN TYPE= 1 DIRECTION= 1 TIME= 825.50000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM     IDENT=   54 ISI=   13 TURN TYPE= 0 DIRECTION= 3 TIME= 838.50000
TIME IN SYSTEM=  36.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM     IDENT=   55 ISI=   14 TURN TYPE= 0 DIRECTION= 2 TIME= 844.00000
TIME IN SYSTEM=  40.00000 STOP TIME DELAY=   3.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  12.00000

VEHICLE LEFT SYSTEM     IDENT=   57 ISI=   17 TURN TYPE= 0 DIRECTION= 0 TIME= 844.50000
TIME IN SYSTEM=  37.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM     IDENT=   56 ISI=   20 TURN TYPE= 3 DIRECTION= 0 TIME= 850.00000
TIME IN SYSTEM=  45.00000 STOP TIME DELAY=   7.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  16.00000

VEHICLE LEFT SYSTEM     IDENT=   58 ISI=   21 TURN TYPE= 3 DIRECTION= 3 TIME= 852.00000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   2.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM     IDENT=   59 ISI=   15 TURN TYPE= 0 DIRECTION= 2 TIME= 858.00000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   2.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM     IDENT=   60 ISI=   16 TURN TYPE= 0 DIRECTION= 0 TIME= 867.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM     IDENT=   61 ISI=   18 TURN TYPE= 3 DIRECTION= 2 TIME= 872.50000
TIME IN SYSTEM=  38.00000 STOP TIME DELAY=   3.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  12.00000
```

```
VEHICLE LEFT SYSTEM    IDENT=  62 ISI=   13 TURN TYPE= 0 DIRECTION= 3 TIME= 887.50000
TIME IN SYSTEM=  41.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  12.00000

VEHICLE LEFT SYSTEM    IDENT=  63 ISI=   14 TURN TYPE= 1 DIRECTION= 3 TIME= 893.50000
TIME IN SYSTEM=  37.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM    IDENT=  64 ISI=   15 TURN TYPE= 3 DIRECTION= 3 TIME= 901.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM    IDENT=  65 ISI=   17 TURN TYPE= 0 DIRECTION= 0 TIME= 907.00000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM    IDENT=  66 ISI=   20 TURN TYPE= 1 DIRECTION= 0 TIME= 930.00000
TIME IN SYSTEM=  44.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  17.00000

VEHICLE LEFT SYSTEM    IDENT=  67 ISI=   13 TURN TYPE= 3 DIRECTION= 3 TIME= 936.00000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM    IDENT=  68 ISI=   14 TURN TYPE= 0 DIRECTION= 2 TIME= 954.00000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM    IDENT=  69 ISI=   15 TURN TYPE= 0 DIRECTION= 0 TIME= 960.50000
TIME IN SYSTEM=  36.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   7.00000

VEHICLE LEFT SYSTEM    IDENT=  70 ISI=   16 TURN TYPE= 1 DIRECTION= 1 TIME= 972.50000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM    IDENT=  71 ISI=   17 TURN TYPE= 1 DIRECTION= 3 TIME= 989.50000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM    IDENT=  72 ISI=   18 TURN TYPE= 1 DIRECTION= 0 TIME= 993.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=  73 ISI=   13 TURN TYPE= 0 DIRECTION= 3 TIME=1013.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM    IDENT=  74 ISI=   15 TURN TYPE= 3 DIRECTION= 3 TIME=1033.50000
TIME IN SYSTEM=  36.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=  75 ISI=   17 TURN TYPE= 0 DIRECTION= 2 TIME=1048.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM    IDENT=  76 ISI=   13 TURN TYPE= 0 DIRECTION= 0 TIME=1061.00000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM    IDENT=  78 ISI=   14 TURN TYPE= 0 DIRECTION= 0 TIME=1068.50000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000
```

```
VEHICLE LEFT SYSTEM     IDENT=   77 ISI=   19 TURN TYPE= 0 DIRECTION= 3 TIME=1074.00000
TIME IN SYSTEM=  45.00000 STOP TIME DELAY=    5.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  17.00000

VEHICLE LEFT SYSTEM     IDENT=   79 ISI=   18 TURN TYPE= 0 DIRECTION= 3 TIME=1084.50000
TIME IN SYSTEM=  36.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM     IDENT=   80 ISI=   15 TURN TYPE= 1 DIRECTION= 1 TIME=1093.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM     IDENT=   82 ISI=   16 TURN TYPE= 3 DIRECTION= 1 TIME=1112.00000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM     IDENT=   81 ISI=   13 TURN TYPE= 0 DIRECTION= 0 TIME=1122.50000
TIME IN SYSTEM=  45.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  17.00000

VEHICLE LEFT SYSTEM     IDENT=   83 ISI=   15 TURN TYPE= 0 DIRECTION= 3 TIME=1130.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM     IDENT=   84 ISI=   14 TURN TYPE= 1 DIRECTION= 0 TIME=1147.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM     IDENT=   85 ISI=   18 TURN TYPE= 3 DIRECTION= 3 TIME=1156.00000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM     IDENT=   86 ISI=   13 TURN TYPE= 0 DIRECTION= 1 TIME=1170.00000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM     IDENT=   87 ISI=   15 TURN TYPE= 1 DIRECTION= 1 TIME=1185.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM     IDENT=   88 ISI=   16 TURN TYPE= 0 DIRECTION= 0 TIME=1200.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM     IDENT=   89 ISI=   13 TURN TYPE= 3 DIRECTION= 1 TIME=1208.50000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM     IDENT=   90 ISI=   14 TURN TYPE= 0 DIRECTION= 3 TIME=1225.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  10.00000

VEHICLE LEFT SYSTEM     IDENT=   91 ISI=   15 TURN TYPE= 3 DIRECTION= 2 TIME=1238.00000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM     IDENT=   92 ISI=   18 TURN TYPE= 0 DIRECTION= 3 TIME=1248.00000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=    1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000
```

```
VEHICLE LEFT SYSTEM    IDENT=   93 ISI=   14 TURN TYPE= 0 DIRECTION= 3 TIME=1269.00000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   94 ISI=   17 TURN TYPE= 0 DIRECTION= 2 TIME=1274.50000
TIME IN SYSTEM=  39.00000 STOP TIME DELAY=   3.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=  12.00000

VEHICLE LEFT SYSTEM    IDENT=   95 ISI=   19 TURN TYPE= 0 DIRECTION= 2 TIME=1280.50000
TIME IN SYSTEM=  35.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   96 ISI=   15 TURN TYPE= 0 DIRECTION= 0 TIME=1290.50000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   7.00000

VEHICLE LEFT SYSTEM    IDENT=   97 ISI=   18 TURN TYPE= 1 DIRECTION= 0 TIME=1295.50000
TIME IN SYSTEM=  37.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000

VEHICLE LEFT SYSTEM    IDENT=   98 ISI=   16 TURN TYPE= 3 DIRECTION= 3 TIME=1312.00000
TIME IN SYSTEM=  38.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=   99 ISI=   17 TURN TYPE= 1 DIRECTION= 3 TIME=1324.00000
TIME IN SYSTEM=  34.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   9.00000

VEHICLE LEFT SYSTEM    IDENT=  100 ISI=   14 TURN TYPE= 3 DIRECTION= 3 TIME=1336.00000
TIME IN SYSTEM=  33.00000 STOP TIME DELAY=   1.00000  TIME DELAY FOR VEHICLES UNDER  10.00000MPH=   8.00000
```

```
 GRAND TOTAL SUMMARY INFORMATION FOR TIME =1403.00000
SYSTEM USE TIME=1275.00000


NORTHBOUND APPROACH
 STOP TIME DELAY...RT TURN=   1.00000  LEFT TURN=   1.06667  ST THRU=   1.25000
TIME DELAY FOR  10.00000  MPH=.... RT TURN=  10.12500  LEFT TURN=   8.66667  ST THRU=   9.30000
TOTAL VEHICLES COUNTED=....RT TURN=     8  LEFT TURN=    15  ST THRU=    20  AVERAGE TIME IN SYSTEM=  35.39535


EAST BOUND APPROACH
 STOP TIME DELAY...RT TURN=   1.00000  LEFT TURN=   7.00000  ST THRU=   1.00000
TIME DELAY FOR  10.00000  MPH=.... RT TURN=  13.00000  LEFT TURN=  16.00000  ST THRU=   7.50000
TOTAL VEHICLES COUNTED=....RT TURN=     1  LEFT TURN=     1  ST THRU=     2  AVERAGE TIME IN SYSTEM=  38.25000


SOUTHBOUND APPROACH
 STOP TIME DELAY...RT TURN=   1.00000  LEFT TURN=   1.00000  ST THRU=   1.33333
TIME DELAY FOR  10.00000  MPH=.... RT TURN=   8.40000  LEFT TURN=   9.00000  ST THRU=   9.00000
TOTAL VEHICLES COUNTED=....RT TURN=     5  LEFT TURN=     3  ST THRU=    15  AVERAGE TIME IN SYSTEM=  35.04348


WESTBOUND APPROACH
 STOP TIME DELAY...RT TURN=   1.00000  LEFT TURN=   2.40000  ST THRU=   1.28571
TIME DELAY FOR  10.00000  MPH=.... RT TURN=  10.09091  LEFT TURN=  12.00000  ST THRU=  10.00000
TOTAL VEHICLES COUNTED=....RT TURN=    11  LEFT TURN=     5  ST THRU=    14  AVERAGE TIME IN SYSTEM=  36.53333
```

CHAPTER V

TRAFFIC CONTROL

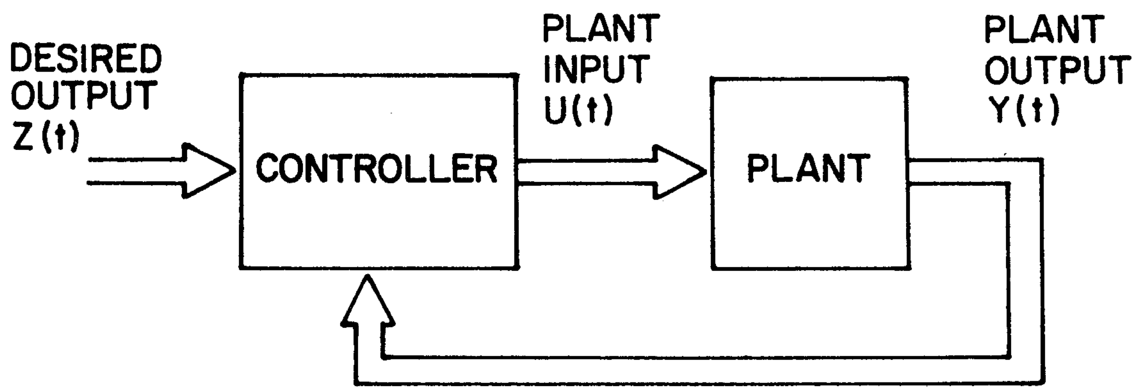
Use of the digital computer for traffic control has increased somewhat in the past few years. Several systems in Germany, Canada and the United States[9, 10] have been implemented and are currently providing limited traffic control with its use. The need for traffic control can be noted from the following quotation which appeared in a recent magazine article.[9]

> "It's no exaggeration that a kid on roller skates can propel himself crosstown from the East River to the North River, on any of Manhattan's midtown streets in considerably less time than that required by a motorist."

The successful uses of the computer in process control and other related fields has increased confidence in the computer for the traffic control problem. Unfortunately, it is the system or control technique of which the computer is but one subsystem, and not just the computer alone that is required for solving the traffic control problem. As will be discussed subsequently, the model developed in this study appears useful for the control problem with and without feedback, as well as for the adaptive control problem.

5.1 Control System Types

Three types of control techniques are prevalent in the control field, namely, control without feedback (open loop), control with feedback (closed loop), and adaptive control. Figure 5.1 depicts the typical control system.[11] The plant shown in the figure is the part of the system which is to be controlled. For the traffic control problem, the plant represents the actual traffic network, that is the traffic lights. Thus, the inputs $\underline{u}(t) = (u_1(t), \ldots u_r(t))^T$ represent the control signal commands.



TYPICAL CONTROL SYSTEM

FIG. 5.1

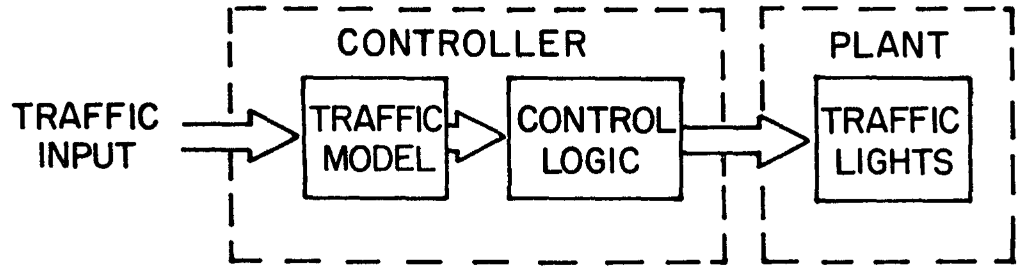
The controller is the unit which controls the plant in some acceptable manner. Thus, for the traffic control problem, the controller is the control computer and software control logic. Its outputs are the control signals $\underline{u}(t)$ selected so that the system performs in the acceptable manner. The desired system outputs $\underline{z}(t) = (z_1(t), \ldots z_n(t))^T$ are the inputs for the criteria used to insure the system is performing as specified. The plant outputs, $\underline{y}(t) = (y_1(t), \ldots y_n(t))^T$ represent the various traffic summary results, such as traffic density, delay time, etc. If the controller outputs do not depend on the plant outputs, $\underline{y}(t)$, then the system is referred to as an open-loop control system[11] or control system without feedback. When the controller uses the $\underline{y}(t)$ to generate the control signals, the system is referred to as a closed-loop control system or control system with feedback.

The adaptive control system is one which measures a performance and modifies its parameters to approach an optimum set of values.[12] The adaptive system may or may not be closed-loop. The adaptive control system changes the parameters used in computing the control signals as opposed to a simple closed-loop system which only modifies the variables.

Thus, for the traffic control problem the control signal selection technique or model would be modified to accommodate unexpected traffic conditions. The computer traffic control systems currently in use are crude forms of both the open and closed-loop systems. None appears to be of the adaptive control type, although this typically is one of the more difficult control techniques to employ. Following is a discussion of some likely ways the traffic simulation model might be implemented to perform in the control environments.

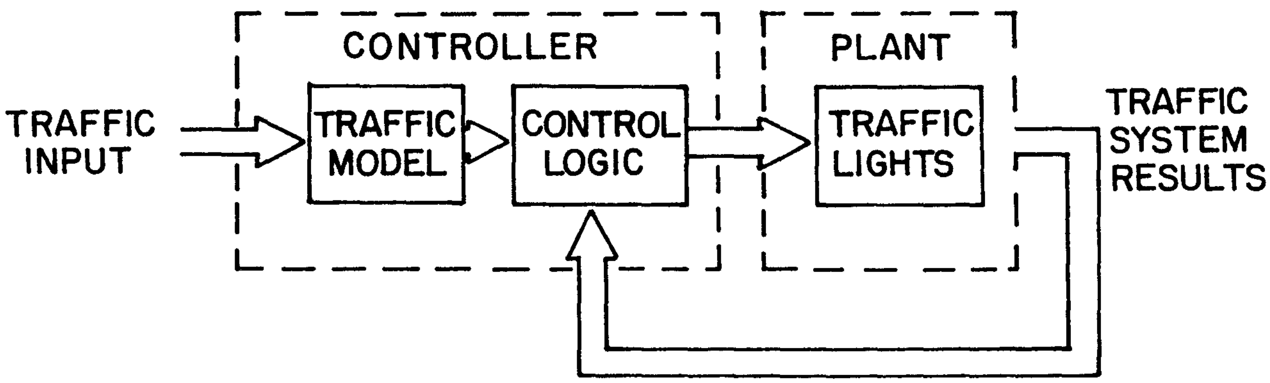
5.2 Use of the Model in Traffic Control

Operation of the model in an open-loop control system is depicted in Figure 5.2a. Prior to control operations, the percentages of cars



OPENLOOP CONTROL SYSTEM

FIG. 5.2a



CLOSEDLOOP CONTROL SYSTEM

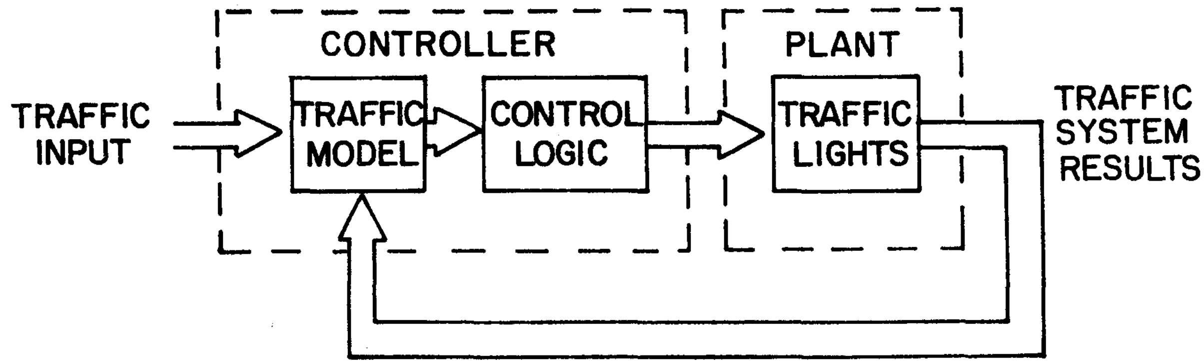
FIG. 5.2b

taking a particular path must be determined. Similar percentages need to be found for entry speeds and for vehicle lengths, if the necessary sensors are not used. The model then functions by queueing vehicles into the system according to the presence detectors located at each of the system entry points. The proper proportion of vehicles taking a particular path is determined by the predetermined percentages. The model then predicts the necessary summary statistics, such as the stop time delay, traffic flow, etc. The control signals are then modified so as to decrease this delay or increase traffic flow, depending on the particular desired performance criteria. As noted in the figure, two control blocks are indicated. The first block is the traffic model and provides predictions of the various network summary statistics. The second control block selects the proper control signals in accordance with this summary data.

Open loop operations have three weak points. The first is that the numbers of vehicles taking a particular path or traveling at a particular speed, if not computed, are not the actual traffic conditions. Secondly, the model has no way of knowing how well it is achieving the desired performance. Finally, the model has no way of determining how well it is actually predicting the traffic flow data.

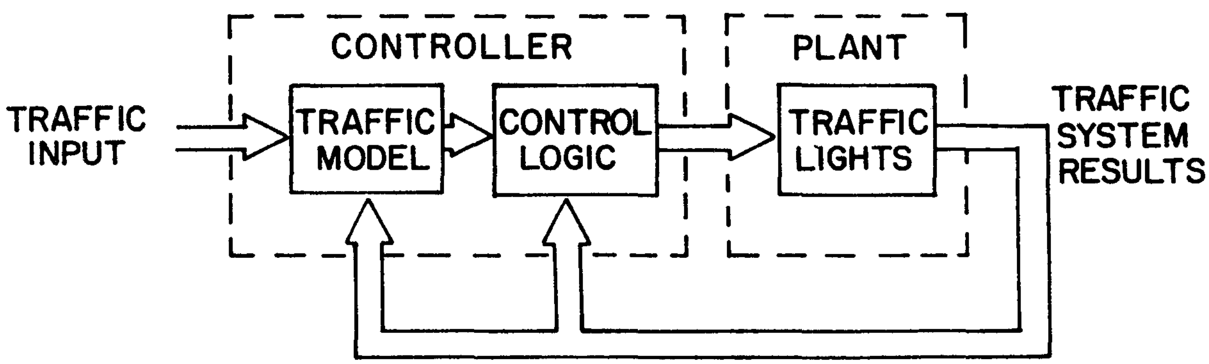
The closed-loop control technique, depicted in Fig. 5.2b reduces these weaknesses to two. With feedback, if properly applied, the model can determine how well it is controlling traffic by comparison of desired and actual traffic flow. The traffic control program can use search procedures for determining optimum performance criteria by making incremental changes in the various control logic criteria. If, for instance, an improvement was not observed, the controlling variables could be changed back to their original values and a new direction taken. This is distinguished from the adaptive control problem in that the controlling logic or control signals are being modified as a function of the plant output signals. The closed loop system can thus take various forms of complexity depending on the extent to which control over the controlling logic is desired.

The adaptive control system is depicted in Fig. 5.3 for both the open-loop and closed-loop control systems. If the adaptive system is used in the closed-loop system, only the first weakness described above remains. In the adaptive system, the model is modified according to the actual traffic conditions. Similar to the closed loop problem there are mathematical techniques available such as steepest ascent, etc. for determining optimum model operations. The adaptive control technique would, of course, be the most desirable system although its increased advantages might not merit the increased costs. A study of the possible use of the model in traffic control is strongly recommended if it is proven that the model can satisfactorily perform traffic simulations.



OPEN LOOP ADAPTIVE CONTROL SYSTEM

FIG. 5.3a



CLOSE LOOP ADAPTIVE CONTROL SYSTEM

FIG. 5.3b

CHAPTER VI

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

A modeling method for traffic simulation studies has been developed using a step-through procedure as opposed to the usual queueing theory approach. A computer program was written for implementing this technique using a hypothetical intersection controlled by four-way stop signs. By employing modern systems engineering techniques, this model promises to become a valuable tool as an aid to real-time signal control. The model developed offers the following advantages over similar modeling techniques.

1. It is readily adaptable to the traffic network problem.
2. The program used for implementing this model requires much less memory and computer time requirements than similar simulation models.
3. The modular design of the model permits ease in expansion to include the many and varied intersection types.
4. The program can be implemented on a small process control computer such as the IBM 1800 for real-time traffic control.

It is recommended that a study be performed where actual traffic data is compared with the model predictions and the program corrected accordingly. The study can then be expanded to include the multiple intersection problem. Once the model has been fitted, a useful tool for traffic simulation studies can be ascertained. Studies can then be expanded to include the traffic control problem. Thus it is recommended that the model and program developed herein be the basis of additional research. It is also recommended that the modeling technique be further investigated for its possible use for other simulation problems as well as for consideration in the development of a new problem oriented language.

APPENDIX 1

PROGRAM USAGE INSTRUCTIONS

Six input card types are required by the model program for traffic simulation operation. These six card input types are:

(1) Program Printout Options,

(2) Model Operation Variables,

(3) Driver Response Functions,

(4) Stop Sign and Dummy I/0 Objects,

(5) Vehicles,

(6) Model Terminator.

Following is a description of each of these input card types. Figures A1. 1 - A1. 5 provide a guide useful for generating these cards and indicating their proper order in the data deck.

<u>Printout Options:</u> Seven printout options are available during program execution as represented by the variables KCT(1) to KCT(7). To select any option, the variable KCT(j) is set to 1. These options are:

1. KCT(1), Print Driver Response Input Card. If this option is specified, the driver response card images will be printed.

2. KCT(2), Print System Generated Response. If this option is specified, the response functions as used by the decision processor are printed. This option was provided primarily for debugging purposes.

3. KCT(3), Print Vehicle Information: If this option is specified, the object input card image is printed as described in the network summary description of Chapter III.

4. KCT(4), Print Inprocess Summary Information. If this option is specified, the individual summary information as

FIG. A 1 . 1

PRINTOUT OPTION DATA

CARD FORMAT

Column	
1	KCT(1) = 1 - print driver response input
2	KCT(2) = 1 - print system generated response
3	KCT(3) = 1 - print vehicle input infor.
4	KCT(4) = 1 - print inprocess summary infor.
5	KCT(5) = 1 - print vehicle exit infor.
6	KCT(6) = 1 - print gross summary infor.
7	KCT(7) = 1 - print grand total summary infor.
8 – 9	not used
10 – 11	CTT(1) - Time increment for KCT(4) - F FORMAT
20 – 21	CTT(2) - Time increment for KCT(6) - F FORMAT
30	
31 – 80	not used

FIG. A 1 . 2

MODEL OPERATION VARIABLE DATA

CARD FORMAT

Column	
1 – 5	NR = number of decision response functions; I FORMAT
6 – 15	FT = program termination time; F FORMAT
16 – 25	DT = model time increment; F FORMAT

FIG A1.4

OBJECT DATA CARD FORMAT

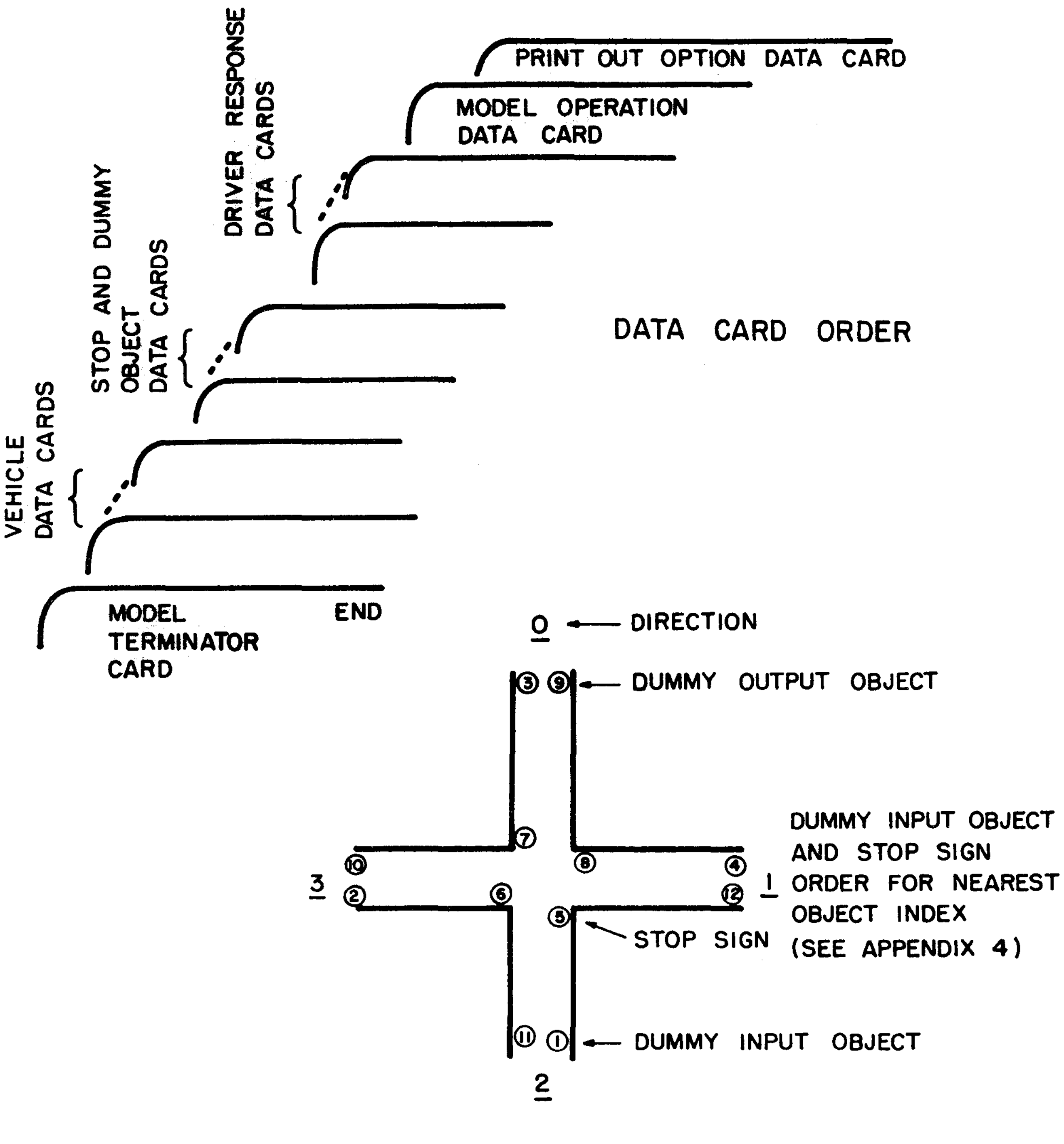
Column	Field
1–5	IP = ENTRY POSITION
6–10	IS = VEHICLE SPEED
11–15	IDS = DESIRED SPEED
16–20	ID = LANE DIRECTION
21–25	IND = NEAREST OBJECT INDEX
26–30	IVL = VEHICLE LENGTH
31–35	IC = OBJECT TYPE
36–40	IT = TURN TYPE
41–45	IDENT = VEHICLE IDENT
46–55	T = ENTRY TIME F FORMAT
56–58	EXI = MODEL TERMINATOR A FORMAT

I FORMAT (IP, IS, IDS, ID, IND, IVL, IC, IT, IDENT)

Fig. A1.3

DRIVER RESPONSE FUNCTION DATA CARD FORMAT

Column	Field
1–23	COLUMNS EQUAL TO: 1 true condition 2 not true condition 0 "don't care" condition; I FORMAT
31–32	RESPONSE FUNCTION INDEX; I FORMAT



DATA CARD ORDER

FIG. A1·5

described in the network summary section of Chapter III is provided every CTT(1) seconds.

5. **KCT(5), Print Vehicle Exit Information.** If this option is specified, the vehicle exit information as specified in the network summary section of Chapter III is provided.

6. **KCT(6), Print Gross Summary Information.** If this option is specified, the gross summary information as described in the network summary section of Chapter III is provided every CTT(2) seconds.

7. **Print Grand Total Summary Information.** If this option is specified, the grand total gross summary information is provided upon model termination.

Figure Al. 1 illustrates the format for the printout option parameters.

Model Operation Variables: The three variables needed by the model for initializing the program should be provided on one card as specified in Fig. Al. 2. These three variables are as follows:

1. NR--the number of decision response function cards.

2. FT--the time in seconds the model is to run following the model terminator card before program termination.

3. DT--the time increment used by the model during simulation.

Driver Response Functions: The driver response functions are entered into the model with this card. Figure Al. 3 depicts the card format for entering this information and the decision process section in Chapter III provides an example. The first 30 columns of the card are used to specify a product term of 30 variables (only 23 are currently used). Columns 31 and 32 are used for entering the response bit location (see Table A3. 1 of Appendix 3). Each column number represents the vehicle input number listed in Table A3. 1.

Stop Sign and Dummy I/0 Objects, Vehicles and Model Terminator:

All three of these card types use the same card format as shown in Fig. Al. 4. The variables shown in this figure are

(i) IP - entry position

(ii) IS - vehicle speed

(iii) IDS - desired speed (IS $\leq$ IDS)

(iv) ID - lane direction

0 = Northbound

1 = Eastbound

2 = Southbound

3 = Westbound

(v) INO - (Nearest Object Index (same as direction for vehicles)

(vi) IVL - Vehicle length

(vii) IC - Object type indicator

0 = vehicle

1 = stop sign

11 = input object

101 = output object

(viii) IT - Turn type

0 = no turn

1 = right turn

3 = left turn

(ix) Ident - Vehicle Identification Tag (0-511)

(x) T - Time of entry into stack (seconds)

(xi) EXT - Model Terminator, "END" for model termination.

Stop sign objects use the position variable, IP, the lane direction, ID, the nearest object index, INO, and the object type indicator, IC. IP is set to the location of the stop sign, INO to the index of the output object (see example, Fig. A1. 5) and IC = 1. The remaining variables are set to zero. The dummy input/output objects use these same variables with IC = 11 and 101, respectively.

The vehicles entering into the system use all the variables indicated except the EXI variable.

The model terminator card uses only the EXI variable which is set to the word END. The other variables may be left blank.

Figure A1. 5 provides an example card input and Appendix 4 provides the input cards for the data run example described in Chapter IV.

APPENDIX 2

CRITICAL DISTANCE CRITERION

The three most sensitive variables in the traffic simulation model are the acceleration, deceleration and critical distance criteria. Each of these parameters is computed in independent subroutines which may easily be changed to accommodate more exact criterion. The critical distance criterion currently used was devised from the deceleration function shown in Fig. A2. 1, where the maximum deceleration of six feet per second per second approximates the maximum used in everyday traffic conditions.[13] Critical distance was defined in Chapter II as that distance which is required to stop or slow the vehicle sufficiently to prevent it from colliding with an adjacent vehicle or object. Thus, the critical distance function depicted in Fig. A2. 1 was developed so that the vehicle could decelerate according to the deceleration function and also be consistent with the critical distance definition. As noted, the critical distance criterion used in Fig. A2. 1 is the standard ellipse equation. Suppose, for example, that a vehicle traveling 30 feet per second comes to within 90 feet of a stop sign for a particular time increment. Then, in order to stop five feet in front of the stop sign he must begin decelerating initially at about 4. 6 feet per second per second.

Acceleration criterion was selected as a constant two feet per second per second.

The above criteria was selected as a first approximation. Experiments in which actual and predicted traffic flow comparisons are performed will probably modify these estimates. These functions, however, provide useful criteria necessary in the initial development of the program model.

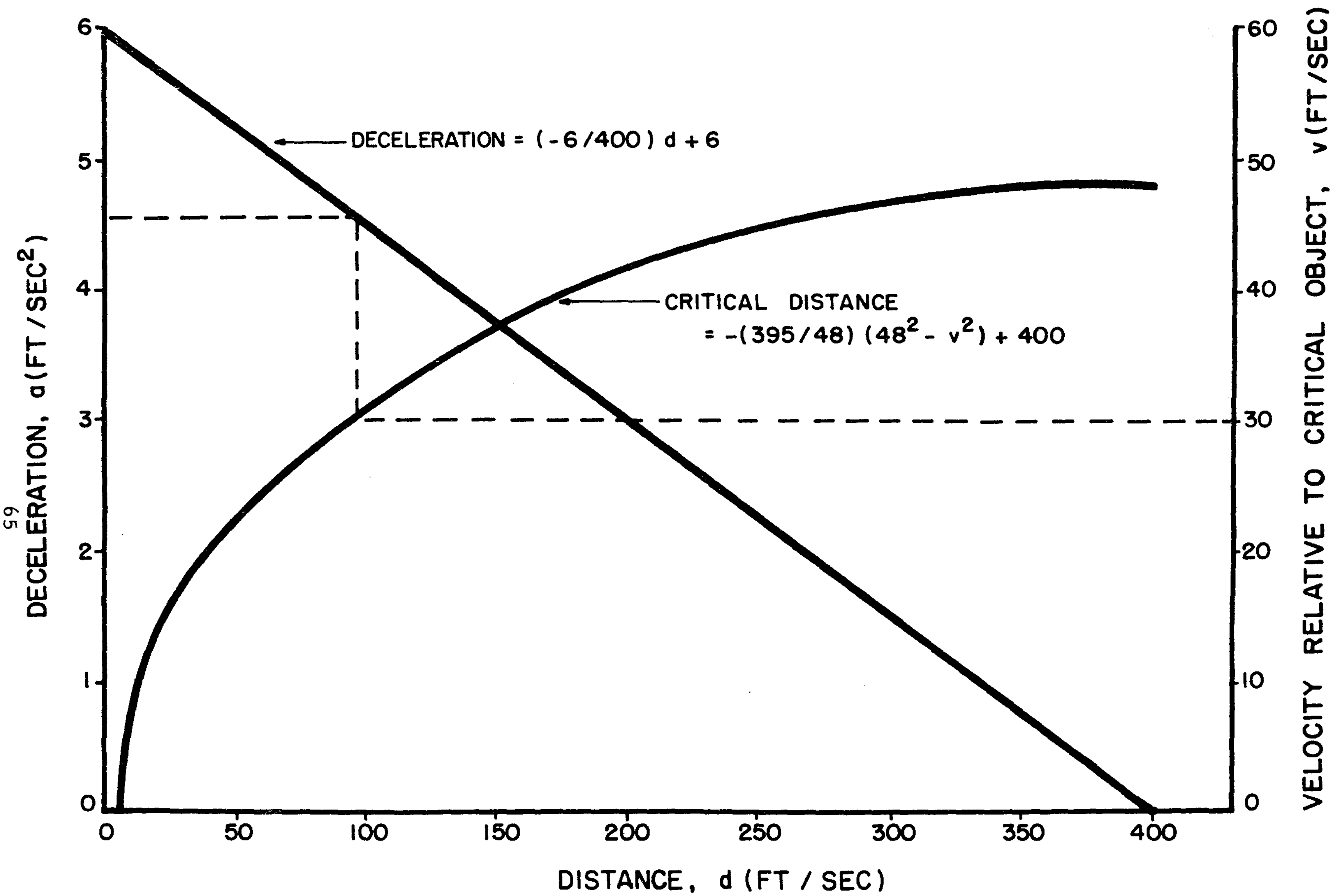


FIG. A 2·1
CRITICAL DISTANCE AND DECELERATION FUNCTIONS

APPENDIX 3

PROGRAM TABLES AND FLOW CHARTS

TABLE A3.1

PROGRAM TABLES

IVCH

Bit	59	58	57	56	47	46	37	35	33	32	23	21	15	9
Field	object	in-put	out-put		pass	pass object index	appro-ach	turn	ignore pass flag	nearest object index	direct-ion	desired speed	present speed	position

IVCHI

Bit		32	26	20	14	8
Field		total time delay	stop time delay	time in system	ident tag	vehicle length

IVEH*

Bit		46	44	42	40	38	36	34	32	30	28	26	24	22	20	18	16	14	12	10	8
Number		20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

IVEH* (low-order field): index of IY mask

IVEHI*

Bit		5	3	1
Number		23	22	21

IY

Bit		13	11	
Field		neg accel	pos accel	

* See following page for number-variable relationships.

TABLE A3.1 (cont'd)

NUMBER - VARIABLE RELATIONSHIPS

Number	Variable
1	At desired speed
2	Greater than desired speed
3	At critical distance
4	Greater than critical distance
5	Stopped
6	In intersection
7	First vehicle in intersection
8	Car on right
9	Car on left
10	Car straight ahead
11	Exactly at intersection
12	Car on right exactly at intersection
13	Car on left exactly at intersection
14	Car straight ahead exactly at intersection
15	Car on right right turning
16	Car on right left turning
17	Car on left right turning
18	Car on left left turning
19	Car straight ahead right turning
20	Car straight ahead left turning
21	Right turning vehicle
22	Left turning vehicle
23	Straight thru vehicle

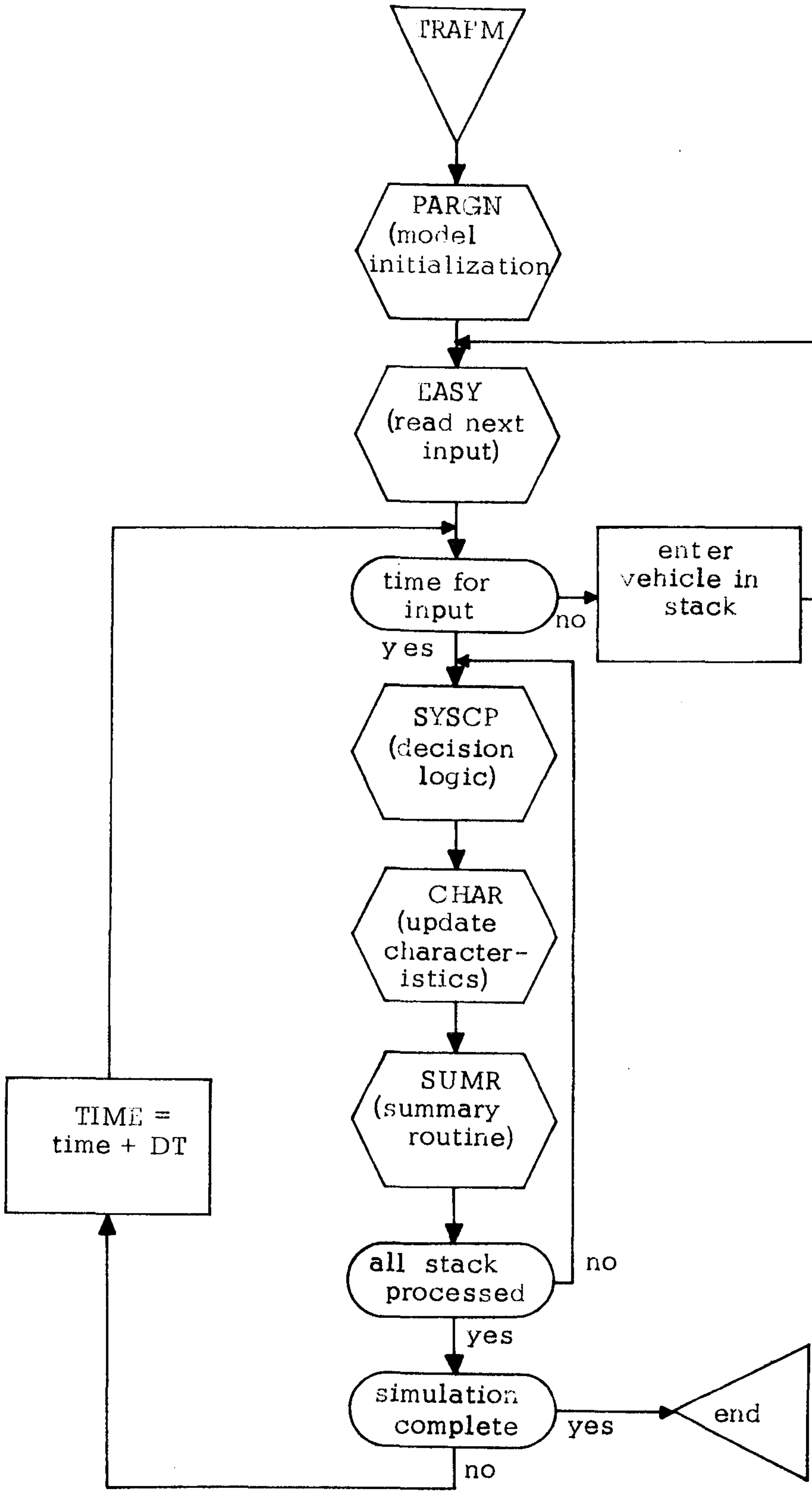
See pages 22-23 for discussion.

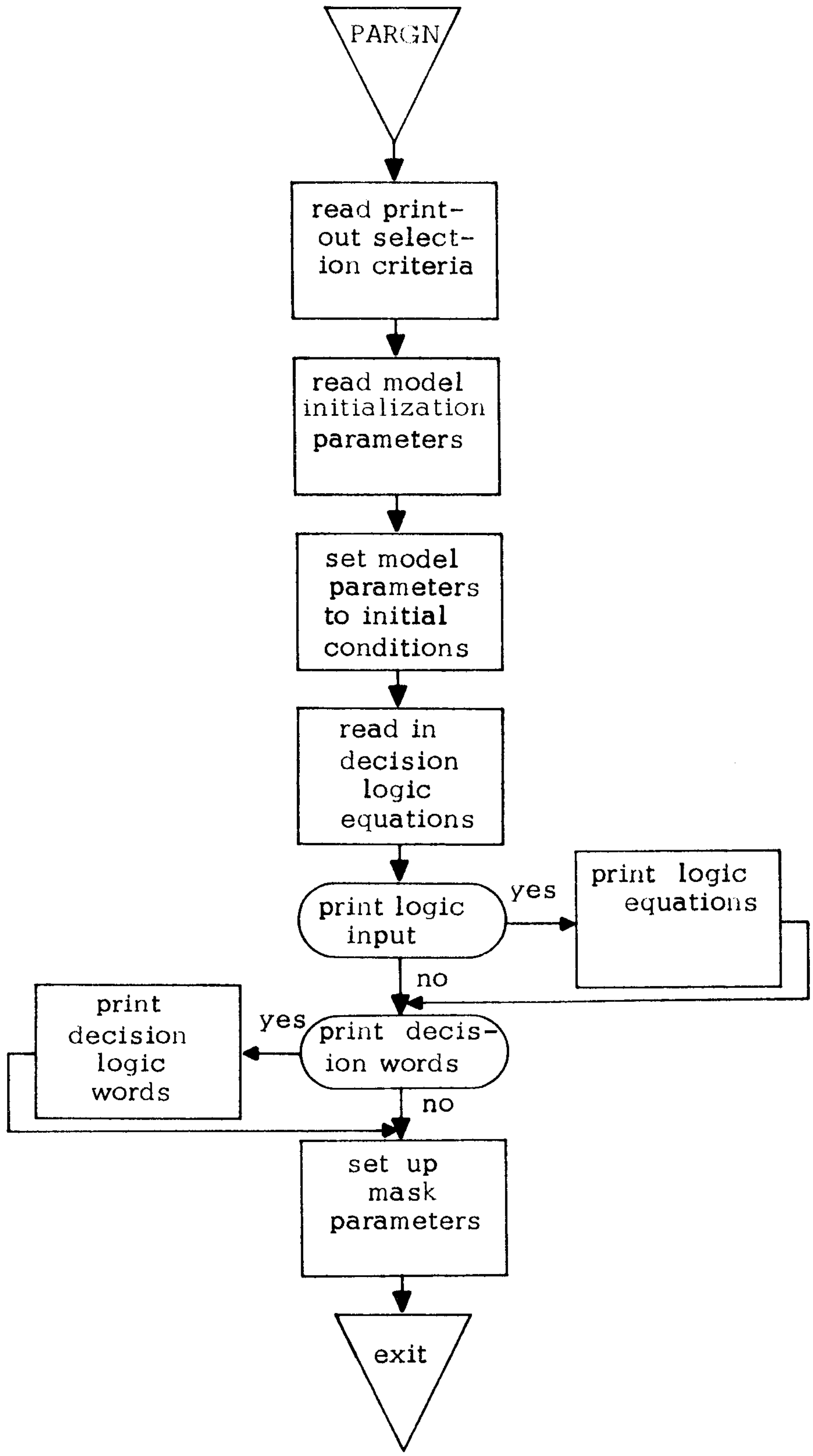
TABLE A3.2

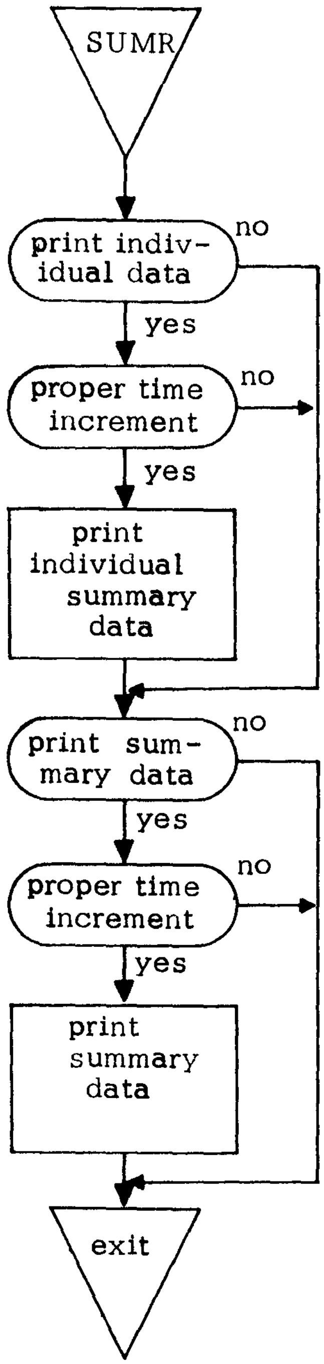
LIST OF PROGRAM SUBROUTINE

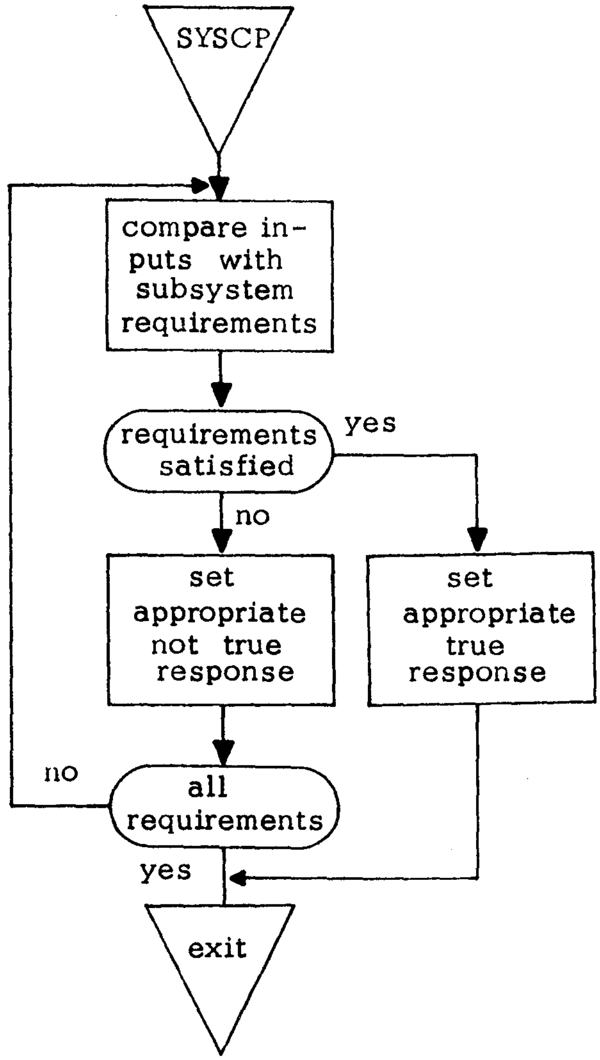
Subroutine	Purpose
1. TRAFM	Main driver subroutine; begins program; call main processing routines until time for program termination.
2. SUMR	Network summary subroutine; see description, Chapter III.
3. EASY	Subroutine used for reading object data; sets up stack entry for each object.
4. PARGN	Subroutine used for program initialization; sets up all program parameters.
5. SYSCP	Decision processor subroutine; see description, Chapter III.
6. CHAR	Identification update subroutine; see description, Chapter III.
7. NOI	Subroutine used for obtaining nearest object index.
8. OBVC	Subroutine used to extract stack identification information.
9. SETUP	Subroutine used to set up new stack identification information.
10. ACCEL	Subroutine used to compute acceleration and deceleration values; see Appendix 2.

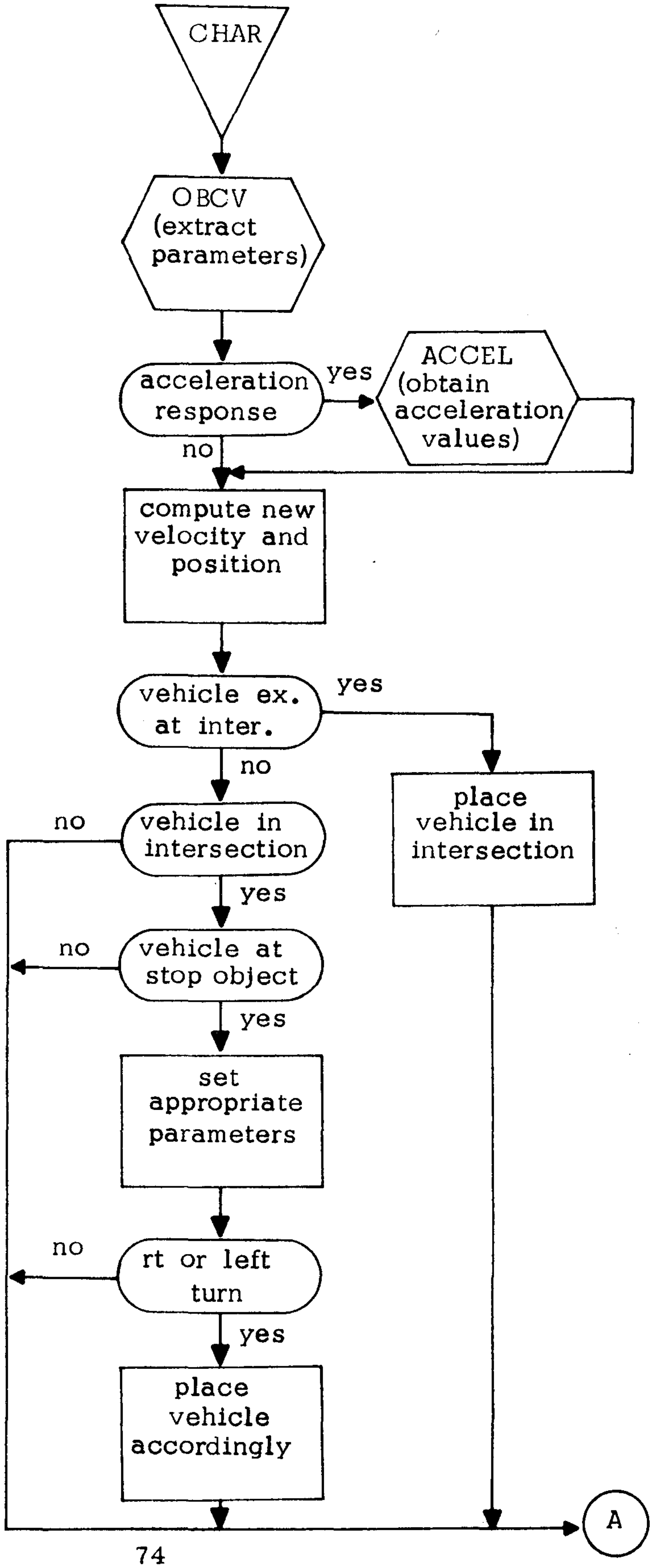
PROGRAM FLOW CHARTS

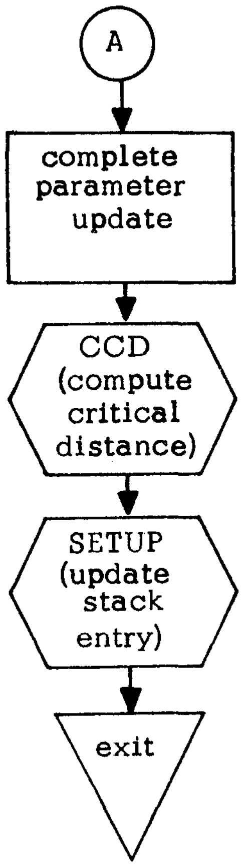


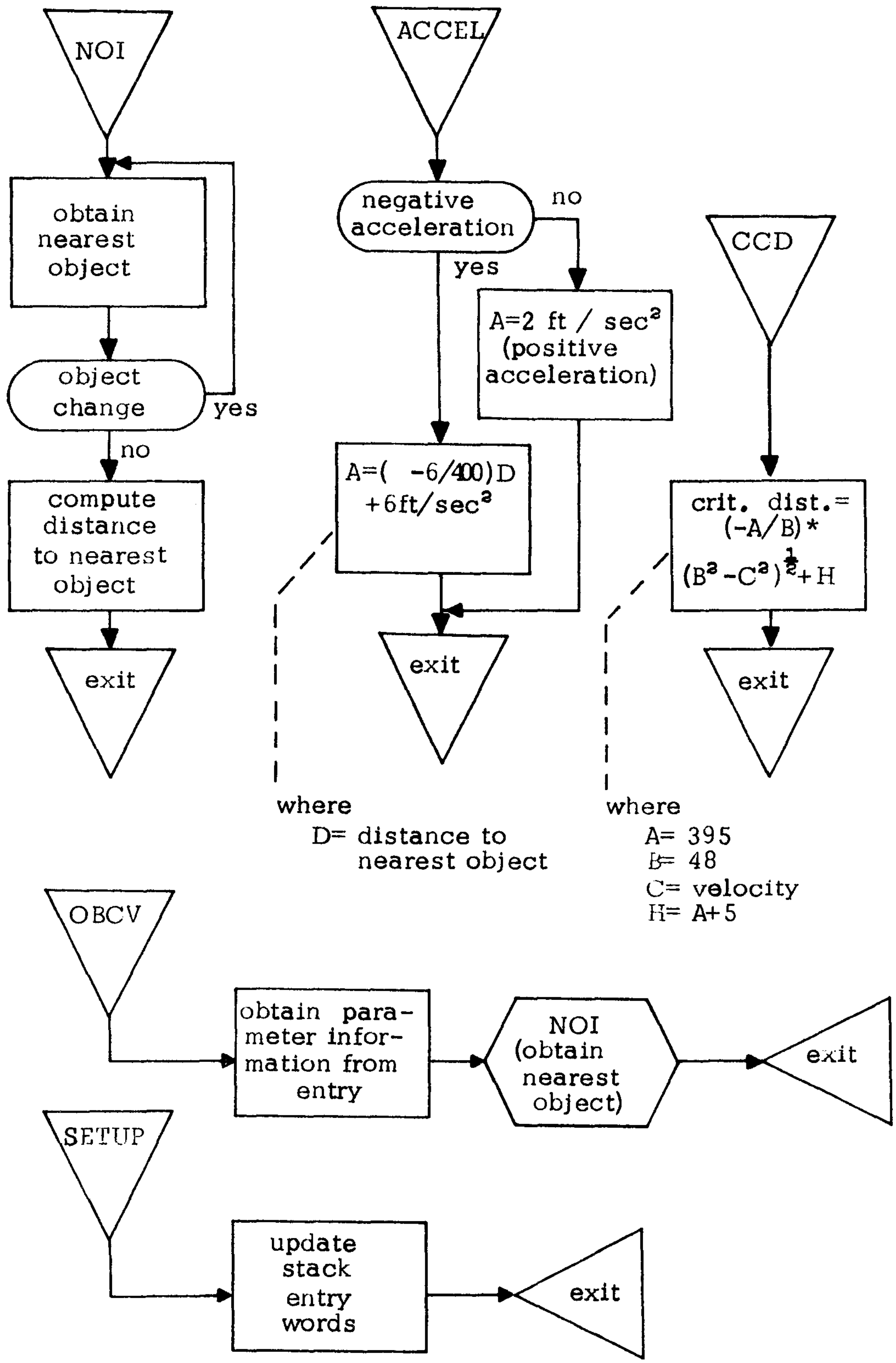

PARGN
read print-out selection criteria
read model initialization parameters
set model parameters to initial conditions
read in decision logic equations
print logic input
yes
print logic equations
no
print decision words
yes
print decision logic words
no
set up mask parameters
exit


SUMR
print indiv-
idual data
no
yes
proper time
increment
no
yes
print
individual
summary
data
print sum-
mary data
no
yes
proper time
increment
no
yes
print
summary
data
exit


SYSCP
compare in-puts with subsystem requirements
requirements satisfied
yes
no
set appropriate not true response
set appropriate true response
no
all requirements
yes
exit


CHAR
OBCV
(extract
parameters)
acceleration
response
yes
no
ACCEL
(obtain
acceleration
values)
compute new
velocity and
position
vehicle ex.
at inter.
yes
no
place
vehicle in
intersection
no
vehicle in
intersection
yes
no
vehicle at
stop object
yes
set
appropriate
parameters
no
rt or left
turn
yes
place
vehicle
accordingly
A


A
complete parameter update
CCD (compute critical distance)
SETUP (update stack entry)
exit


NOI
obtain nearest object
object change
yes
no
compute distance to nearest object
exit
ACCEL
negative acceleration
no
yes
A=2 ft / sec²
(positive acceleration)
A=(-6/400)D +6ft/sec²
exit
where
D= distance to nearest object
CCD
crit. dist.= (-A/B)* (B²-C²)^½+H
exit
where
A= 395
B= 48
C= velocity
H= A+5
OBCV
obtain parameter information from entry
NOI (obtain nearest object)
exit
SETUP
update stack entry words
exit

APPENDIX 4

PROGRAM LISTING

SIMULATION PROGRAM

```
      PROGRAM TRAFM(INPUT,OUTPUT)
      COMMON/SYS/ISYSI(60),ISYSY(60)
      COMMON/CHV/IPOS,IVEL,IDVL,ID,INA,IACL,IPSI,IDUM(20),INSEC(11)
     1,IT,IDUM1(10),ITIM(3),FDT,IXIT
      COMMON/STACK/IVEH(100),IVEH1(100),IVCH(100),IVCH1(100),IY
      COMMON/MSK/MASK(25),K(25),N(25),NR,DI,INSP1,INSP2,IVII,FI
      COMMON/SUMRY/KCT(10),CTT(10),IFCT(10,2,4)
      COMMON/CRDP/H,A,B
      DIMENSION PRI(7,4)
      L=-1
      FDM=0
      IME2=0
      SYSTM=0
      RT=FDT=0
      ISC=0
      CALL PARGN
5     CALL EASY(IVH,IVH1,IVC,IVC1,T,RT,IV1,IV2,IV3)
      IF(ISC.EQ.0)13,9
9     IF(T.GT.RT)10,12
12    DO 15 I=1,ISC
      IF(IVCH(I).EQ.L)16,15
15    CONTINUE
13    ISC=ISC+1
      I=ISC
16    IVEH(I)=IVH
      IVEH1(I)=IVH1
      IVCH(I)=IVC
      IVCH1(I)=IVC1
      RTT=RT+FT
      IF(IV1.EQ.0)94,83
94    IF(IV2.EQ.1)81,82
81    IFCT(8,1,IV3)=IFCT(8,1,IV3)+1$IFCT(8,2,IV3)=IFCT(8,2,IV3)+1
      GO TO 83
82    IF(IV2.EQ.3)84,85
84    IFCT(9,1,IV3)=IFCT(9,1,IV3)+1$IFCT(9,2,IV3)=IFCT(9,2,IV3)+1
      GO TO 83
85    IFCT(10,1,IV3)=IFCT(10,1,IV3)+1$IFCI(10,2,IV3)=IFCT(10,2,IV3)+1
83    CONTINUE
      GO TO 5
10    FDM=FDM+DT
      ISTIME=FDM-IME2
      IF (ISTIME.GE.1)GO TO 341
      FDT=0$GO TO 343
341   FDT=ISTIME
343   IME2=FDM
      IRJ=1
      DO 100 ISI=1,ISC
      IF(IVCH(ISI).EQ.L)100,201
201   CONTINUE
      IF(IVCH(ISI))304,307,307
307   CONTINUE
      GO TO (303,304)IRJ
303   IRJ=2$SYSTM=SYSTM+DT
304   CONTINUE
C     PROCESS CYCLE
```

```
      CALL SYSCP(ISI)
      CALL CHAR(ISI,RT)
      IF(IVCH(ISI))100,43,43
43    CALL SUMR(ISI,ISC,RT)
100   CONTINUE
90    RT=RT+DT $GO TO 9
26    IF(KCT(7).EQ.1)31,32
31    CONTINUE
      PRINT 110,RT,SYSTM
110   FORMAT (1H1,*  GRAND TOTAL SUMMARY INFORMATION FOR TIME =*,F10.5/
     1*  SYSTEM USE TIME=*,F10.5)
      DO 120 J=1,4
      GO TO (121,122,123,124)J
121   PRINT 125$GO TO 130
122   PRINT 126$GO TO 130
123   PRINT 127$GO TO 130
124   PRINT 128$GO TO 130
128   FORMAT(///*  WESTBOUND APPROACH*)
125   FORMAT(///*  NORTHBOUND APPROACH*)
126   FORMAT(///*  EAST BOUND APPROACH*)
127   FORMAT(///*  SOUTHBOUND APPROACH*)
130   J1=8
      DO 18 I=2,4
      IF(IFCT(J1,1,J).EQ.0)186,182
186   PRI(I,J)=0$GO TO 18
182   PRI(I,J)=IFCT(I,1,J)/(CTT(6)*IFCT(J1,1,J))
18    J1=J1+1
      J1=8
      DO 180 I=5,7
      IF(IFCT(J1,1,J).FQ.0)187,183
187   PRI(I,J)=0$GO TO 180
183   PRI(I,J)=IFCT(I,1,J)/(CTT(6)*IFCT(J1,1,J))
180   J1=J1+1
      PD=IFCT(8,1,J)+IFCT(9,1,J)+IFCT(10,1,J)
      PRI(1,J)=IFCT(1,1,J)/(CTT(6)*PD)
      PRINT 61,(PRI(I,J),I=2,4),CTT(3),(PRI(II,J),II=5,7),
     1(IFCT(LL,1,J),LL=8,10),PRI(1,J)
61    FORMAT(
     1*   STOP TIME DELAY...RT TURN=*,F10.5,*  LEFT TURN=*,F10.5,
     2*  ST THRU=*,F10.5,/*  TIME DELAY FOR*,F10.5,*  MPH=.... RT TURN=*
     3,F10.5,*  LEFT TURN=*,F10.5,*  ST THRU=*,F10.5/*  TOTAL VEHICLES C
     4OUNTED=....RT TURN=*,I5   ,*  LEFT TURN=*,I5   ,*  ST THRU=*,I5
     5/*  AVERAGE TIME IN SYSTEM=*,F10.5)
120   CONTINUE
32    CALL EXIT
      END
      SUBROUTINE SUMR(ISI,ISC,RT)
      COMMON/SUMRY/KCT(10),CTT(10),IFCT(10,2,4)
      COMMON/STACK/IVEH(100),IVEH1(100),IVCH(100),IVCH1(100),IY
      COMMON/CHV/IPOS,IVEL,IDVL,ID,INA,IACL,IPSI,IDUM(20),INSEC(11)
     1,IT,IDUM1(10),ITIM(3),FDT,IXIT
      COMMON/MSK/MASK(25),K(25),N(25),NR,DT,INSP1,INSP2,IVII,FT
      DIMENSION PRI(7,4)
C     IFCT(1,M)=TOTAL TIME SYSTEM USED
C     IFCT(2,M)=STOP TIME DELAY RT TURN
```

```
C     IFCT(3,M)=STOP TIME DELAY LEFT TURN
C     IFCT(4,M)=STOP TIME DELAY ST THRU
C     IFCT(5,M)=TOTAL TIME DELAY RT TURN
C     IFCT(6,M)=TOTAL TIME DELAY LEFT TURN
C     IFCT(7,M)=TOTAL TIME DELAY ST THRU
C     IFCT(8,M)=NUMBER OF VEHICLES RT TURN
C     IFCT(9,M)=NUMBER OF VEHICLES LEFT TURN
C     IFCT(10,M)=NUMBER OF VEHICLES ST THRU
C     M=1 FOR TOTAL, M=2 FOR SUB TOTAL
C     THE THIRD DIMENSION IS TO IDENTIFY APPROACH
      IVEL=IVCH(ISI).A.MASK(2)
      IVEL=IVEL/2**K(2)
      ID=IVCH(ISI).A.MASK(4)
      ID=ID/2**K(4)
      IF(KCT(4).EQ.1)31,32
31    IF(RT.GE.CTT(4))21,32
21    IF(ISI.EQ.ISC)41,42
41    CTT(4)=RT+CTT(1)
42    VEL=(15./22.)*IVEL
      IDT=IVCH1(ISI).A.MASK(10)
      IDT=IDT/2**K(10)
      IT=IVCH(ISI).A.MASK(14)
      IT=IT/2**K(14)
      PRINT 10,RT,ISI,IDT,IPOS,ID,IT,VEL,IVEH(ISI),IVEH1(ISI),
     1IVCH(ISI),IVCH1(ISI),IY
10    FORMAT(/*  TIME=*,F10.5,*  ISI=*,I5,*  IDENT=*,I5,*  POSITION=*,
     1I5,*  DIRECTION=*,I2,*  TURN TYPE=*,I2,*  VELOCITY=*,F10.5/
     2*  IVEH=*,2(2X,O20),*  IVCH=*,2(2X,O20),*  IY=*,O20)
32    IF(KCT(6).EQ.1)33,34
33    IF(RT.GE.CTT(5))26,34
26    IF(ISI.EQ.ISC)43,34
43    CTT(5)=RT+CTT(2)
      PRINT 110,RT
110   FORMAT (1H1,             *  SUMMARY INFORMATION FOR TIME =*,F10.5)
      DO 120 J=1,4
      GO TO (121,122,123,124)J
121   PRINT 125$GO TO 130
122   PRINT 126$GO TO 130
123   PRINT 127$GO TO 130
124   PRINT 128$GO TO 130
128   FORMAT(///*  WESTBOUND APPROACH*)
125   FORMAT(///*  NORTHBOUND APPROACH*)
126   FORMAT(///*  EAST BOUND APPROACH*)
127   FORMAT(///*  SOUTHBOUND APPROACH*)
130   J1=8
      DO 18 I=2,4
      IF(IFCT(J1,2,J).EQ.0)186,182
186   PRI(I,J)=0$GO TO 18
182   PRI(I,J)=IFCT(I,2,J)/(CTT(6)*IFCT(J1,2,J))
18    J1=J1+1
      J1=8
      DO 180 I=5,7
      IF(IFCT(J1,2,J).EQ.0)187,183
187   PRI(I,J)=0$GO TO 180
183   PRI(I,J)=IFCT(I,2,J)/(CTT(6)*IFCT(J1,2,J))
```

```
180    J1=J1+1
       PD=IFCT(8,2,J)+IFCT(9,2,J)+IFCT(10,2,J)
       PRI(1,J)=IFCT(1,2,J)/(CTT(6)*PD)
       PRINT 61,(PRI(I,J),I=2,4),CTT(3),(PRI(II,J).II=5,7),
      1(IFCT(LL,2,J),LL=8,10),PRI(1,J)
61     FORMAT(
      1*   STOP TIME DELAY...RT TURN=*,F10.5,*  LEFT TURN=*,F10.5,
      2*  ST THRU=*,F10.5,/*  TIME DELAY FOR*,F10.5,*  MPH=.... RT TURN=*
      3,F10.5,*  LEFT TURN=*,F10.5,*  ST THRU=*,F10.5/*  TOTAL VEHICLES C
      4OUNTED=....RT TURN=*,I5   ,*  LEFT TURN=*,I5   ,*  ST THRU=*,I5
      5/*  AVERAGE TIME IN SYSTEM=*,F10.5)
120    CONTINUE
       DO 63 J=1,4
       DO 63 I=1,7
63     IFCT(I,2,J)=0
34     CONTINUE
       RETURN
       END
       SUBROUTINE EASY(IVH,IVH1,IVC.IVC1,T,RT,IV1,IV2,IV3)
       COMMON/MSK/MASK(25),K(25),N(25),NR,DT,INSP1,INSP2,IVII,FT
       COMMON/SUMRY/KCT(10),CTT(10),IFCT(10,2,4)
       DATA END/3HEND/
C     IP=POSITION(1024)
C      IS=SPEED  (64)
C      IDS=DESIRED SPEED (64)
C      ID=DIRECTION (3)
C      IDENT = VEHICLE IDENT TAG(512)
C      INO=NEAREST OBJECT INDEX  (512)
C      IVL=VEHICLE LENGTH  (64)
C      IC=CAR OR OBJECT
C         =001 FOR OBJECT,011 FOR INPUT OBJECT,101 FOR OUTPUT OBJECT
C      IT=RT OR LT, NO TURN OR TURN  (0,1)
       IVC=IVC1=IVH=0
       IVH1=52525252525252525252B
       READ 9,IP,IS,IDS,ID,INO,IVL,IC,IT,IDENT,T,FXI
9      FORMAT(9I5, F10.5,A3)
       IF(FXI.EQ.END)41,42
41     T=RT+FT+5$GO TO 43
42     CONTINUE
5      FORMAT(9I5,2F10.5)
       IF(KCT(3).EQ.1)31,32
31     PRINT 33,T
       PRINT 5,IP,IS,IDS,ID,INO,IVL,IC,IT,IDENT,T,FT
33     FORMAT(//*  VEHICLE INPUT FOR T=*,F10.5)
32     IV1=IC$IV2=IT
       IV3=ID+1
       J1=25252525252525252231B
C      SET UP IVCH(IVC)
       IVC=IVC.O.IP
       IJ=ID
       IJ=IJ*2**K(20)
       IVC=IVC.O.IJ
       IS= IS*(22./15.)
       IS=IS*2**K(2)
       IVC=IVC.O.IS
```

```
      IT=IT*2**K(14)
      IVC=IVC.O.IT
      IDS=IDS*(22./15.)
      IDS=IDS*2**K(3)
      IVC=IVC.O.IDS
      IF(IC.EQ.0)8,7
8     INO=ID+1
7     ID=ID*2**K(4)
      IVC=IVC.O.ID
      INO=INO*2**K(5)
      IVC=IVC.O.INO
      IVC1=IVC1.O.IVL
      IDENT=IDENT*2**K(10)
      IVC1=IVC1.O.IDENT
      IF(IC.EQ.0)12,22
22    IF(IC.EQ.1)10,16
16    IF(IC.EQ.11)18,19
18    IVC=IVC.O.MASK(9)$GO TO 12
19    IVC=IVC.O.K(9)$GO TO 12
10    IVC=IVC.O.MASK(15)
12    CONTINUE
C    SET UP IVEH
      J1=J1*2**7
      J2=177B
      IVH=IVH.O.J1
      IVH=IVH.O.J2
43    CONTINUE
      RETURN
      END
      SUBROUTINE PARGN
      COMMON/STACK/IVEH(100),IVEH1(100),IVCH(100),IVCH1(100),IY
      COMMON/CRDP/H,A,B
      COMMON/SYS/ISYSI(60),ISYSY(60)
      COMMON/MSK/MASK(25),K(25),N(25),NR,DT,INSP1,INSP2,IVII,FT
      COMMON/SUMRY/KCT(10),CTT(10),IECT(10,2,4)
      COMMON/CHV/IPOS,IVEL,IDVL,ID,INA,IACL,IPSI,IDUM(20),INSEC(11)
     1,IT,IDUM1(10),ITIM(3),EDT,IXIT
      DIMENSION KK(30)
C    KCT(1)=PRINT SYSTEM INPUT CARD
C    KCT(2)=PRINT SYSTEM GENERATED INFORMATION
C    KCT(3)=PRINT VEHICLE INPUT INFORMATION
C    KCT(4)=PRINT IN PROCESS SUMMARY INFORMATION EVERY CTT(1) SECONDS
C    KCT(5)=PRINT VEHICLE EXIT INFORMATION
C    KCT(6)=PRINT SUMMARY INFORMATION EVERY CTT(2) SECONDS
C    KCT(7)=PRINT GRAND TOTAL SUMMARY INFORMATION
C    FT =MAX TIME INCREMENT ABOVE T
C    CTT(3) USED FOR TOTAL TIME DELAY PARAMETER
C    CTT(4) USED FOR IN PROCESS SUMMARY COUNT
C    CTT(5) USED FOR SUMMARY COUNT
C     CTT(6)=DELTA T FOR COUNT INFORMATION
      PRINT 101
101    FORMAT(1H1)
      READ8,(KCT(I),I=1,7),(CTT(J),J=1,6)
8     FORMAT(7I1,2X,6F10.3)
      CTT(4)=CTT(1)$CTT(5)=CTT(2)
```

```
      READ1,NR,FT,DT
1     FORMAT(I5,2F10.5)
      IXIT=0
      IVII=0
      A=395.$B=48.
      H=A+5
      IF(CTT(6).EQ.0)91,92
91    CTT(6)=1.
92    DO 93 J=1,4
      DO 93 JJ=1,2
      DO 93 JJJ=1,10
93    IFCT(JJJ,JJ,J)=0
      INSP1=445$INSP2=480
      DO 12 II=1,11
12    INSEC(II)=0
      DO 11 I=1,NR
      ISYSI(I)=ISYSY(I)=0
      IK=0$IKK=7
      READ2,(KK(I1),I1=1,30),KR
2     FORMAT(30I1,I2)
      DO 63 J=1,30
      IF(KK(J).EQ.0)62,63
62    KK(J)=0
63    CONTINUE
      IF(KCT(1).EQ.1)41,42
41    PRINT 15,NR,KK,KR
15    FORMAT(2X,  *NR=*,I5,3X,*  SYSTEM INPUT=*,30I1,I2)
42    CONTINUE
      ISYSI(I)=KR
      DO 31 I1=1,20
      KK(I1)=KK(I1)*2**IKK
      IKK=IKK+2
31    ISYSI(I)=KK(I1).O.ISYSI(I)
      DO 32 I1=21,30
      KK(I1)=KK(I1)*2**IK
      IK=IK+2
32    ISYSY(I)=KK(I1).O.ISYSY(I)
11    CONTINUE
C     MASK(1)=POSITION
      MASK(1)=1777B
      K(1)=1
C  MASK(2)=CURRENT SPEED
      MASK(2)=176000B
      K(2)=10
C   MASK(3)=DESIRED SPEED
      MASK(3)=17600000B
      K(3)=16
C  MASK(4)=DIRECTION
      MASK(4)=60000000B
C     MASK(5)=NEAREST OBJECT INDEX
      K(4)=22
      MASK(5)=77700000000B
      K(5)=24
C    MASK(6)=OBJECT CHANGE
      K(6)=33
```

```
      MASK(6)=1000000000000B
C    MASK(7)=ACCELERATION INDICATOR(IY)
      K(7)=10
      MASK(7)=36000B
C     MASK(8)=MASK TO OBTAIN IVEH PARAMETERS
      MASK(8)=37777777777777600B
      K(8)=7
C    MASK(9) USED FOR INPUT OBJECT MASK
C    K(9) USED FOR OUTPUT OBJECT MASK
      MASK(9)=6000000000           B
      K(9)=5000000000           B
C     MASK(10) USED FOR VEHICLE IDENT
      MASK(10)=77700B
      K(10)=6
C     MASK(11)=PASS INDICATOR
      MASK(11)=40000000000000  B
C    MASK(12)=NEAREST OBJECT INDEX FOR PASS
      MASK(12)=37740000000000000B
      K(12)=38
C    MASK(13)=VEHICLE LENGTH
      K(13)=1
      MASK(13)=77B
C     MASK(14)=TURNING INFORMATION
      K(14)=34
      MASK(14)=600000000000B
C    MASK(15) USED FOR BIT 15
      MASK(15)=400000000           B
C    K(15) USED FOR IDUM1 MASK
      K(15)=3777777B
C    MASK(16) USED TO DETERMINE OBJECT TYPE
      MASK(16)=600000000           B
C    MASK(17) USED FOR ITIM(1)
C    MASK(18) USED FOR ITIM(2)
C    MASK(19) USED FOR ITIM(3)
      MASK(17)=7700000B
      MASK(18)=770000000B
      MASK(19)=77000000000B
      K(17)=15
      K(18)=21
      K(19)=27
C    MASK(20)   USED FOR APPROACH INDICATOR
      MASK(20)=3000000000000B
      K(20)=36
C    MASK(21) USED FOR INA PASS IGNORE
      MASK(21)=1000000000000B
      K(21)=33
      PRINT 6,NR,DT
      IF(KCT(2).EQ.1)51,52
51    PRINT7,(I,ISYSI(I),ISYSY(I),I=1,NR)
6     FORMAT(2X,*NR=*,I5,                *DT=*,E20.10//)
7     FORMAT( *  I=*,I5,*ISYSI=*,O20,*ISYSY=*,O20)
52    CONTINUE
80    FORMAT(  *  POSITION,SPEED,DESIRED SPEED,DIRECTION,INO,CLASS,TURN,
     1IDENT,TIME,STOP TIME*)
      PRINT 101
```

```
      RETURN
      END
      SUBROUTINE SYSCP(ISI)
      COMMON/STACK/IVEH(100),IVEH1(100),IVCH(100),IVCH1(100),IY
      COMMON/SYS/ISYSI(60),ISYSY(60)
      COMMON/MSK/MASK(25),K(25),N(25),NR,DT
      IF(IVCH(ISI))10,9,9
9     CONTINUE
      MYI=177B
      MYFIX=1
      IY=0
      DO 100 I=1,NR
      IYI=ISYSI(I).A.MYI
      IC1=IVEH(ISI).A.ISYSI(I)
      IC2=IVEH1(ISI).A.ISYSY(I)
      IF(IC1.EQ.ISYSI(I))4,2
4     IF(IC2.EQ.ISYSY(I))7,2
2     IYI=IYI+1
7     IYFIX=MYFIX*(2**IYI)
100   IY=IY.O.IYFIX
10    CONTINUE
      RETURN
      END
      SUBROUTINE CHAR(ISI,RT)
      COMMON/SYS/ISYSI(60),ISYSY(60)
      COMMON/STACK/IVEH(100),IVEH1(100),IVCH(100),IVCH1(100),IY
      COMMON/MSK/MASK(25),K(25),N(25),NR,DT,INSP1,INSP2,IVII,FT
      COMMON/SUMRY/KCT(10),CTT(10),IFCT(10,2,4)
      COMMON/CHV/IPOS,IVEL,IDVL,ID,INA,IACL,IPSI,IDUM(20),INSEC(11)
     1,IT,IDUM1(10),ITIM(3),FDT,IXIT
C      IDVI=VELOCITY GT DESIRED VELOCITY INDEX
C      IPOS=CURRENT POSITION
C      IVEL=CURRENT SPEED
C      IDVL=DESIRED SPEED
C      ID=DIRECTION
C      INA=ACTUAL OBJECT INDEX
C      IACL=ACCELERATION INDICATOR
C      IDST=DISTANCE TO NEAREST OBJECT
C      NVEL=NEW VELOCITY
C     IT=TURN INDICATOR  00=NO TURN, 01=RT TURN, 11=LT TURN
C      NPOS=NEW POSITION
C      NACL=ACCELERATION
C      IPSI=POSITION OF NEAREST OBJECT
C      ICD=CRITICAL DISTANCE,ALSO CRITICAL DISTANCE INDICATOR
C      IVID=ZERO VELOCITY INDICATOR
C     IVII=INTERSECTION INDEX
C     INSEC=INTERSECTION STACK
C     INSEC(11)=FIRST VEHICLE AT INTERSECTION
C     IDD=DIRECTION OF SECOND VEHICLE AT INTERSECTION
C     INSP1=BEGINNING OF INTERSECTION
C     INSP2=ENDING OF INTERSECTION
C      IDUM(1)=AT DESIRED SPEED
C      IDUM(2)=GT DESIRED SPEED
C      IDUM(3)=AT CRITICAL DISTANCE
C      IDUM(4)=GT CRITICAL DISTANCE
```

```
C      IDUM(5)=VEHICLE STOPPED
C      IDUM(6)=VEHICLE IN INTERSECTION
C      IDUM(7)=FIRST VEHICLE AT INTERSECTION
C      IDUM(8)=CAR ON RIGHT
C     IDUM9)=CAR ON LEFT
C     IDUM(10)=CAR STRAIGHT AHEAD
C      IDUM(11)=VEHICLE EXACTLY AT INTERSECTION
C     IDUM(12)=CAR ON RT EXACTLY AT INTERSECTION
C     IDUM(13)=CAT ON LT EXACTLY AT INTERSECTION
C     IDUM(14)=CAR AT CT EXACTLY AT INTERSECTION
C     IDUM(15)=CAR ON RT TURNING RT
C     IDUM(16)=CAR ON RT TURNING LEFT
C     IDUM(17)=CAR ON LEFT TURNING RT
C     IDUM(18)=CAR ON LEFT TURNING LEFT
C     IDUM(19)=CAR ST AHEAD TURNING RT
C     IDUM(20)=CAR ST AHEAD TURNING LEFT
C     IDUM1(1)=VEHICLE RT TURN
C     IDUM1(2)=VEHICLE LEFT TURN
C     IDUM1(3)=VEHICLE ST  THRU
C     ITIM(1)=TIME IN SYSTEM
C     ITIM(2)=STOP TIME DELAY
C     ITIM(3)=TOTAL TIME DELAY
       CALL OBCV(ISI)
C  OBTAIN CHARACTERISTICS
       IVC=IVCH(ISI)
       IF(IVC)39,6,6
6      IACL=IACL.A.5
       IF(IACL)4,5,4
4      CALL ACCEL(IPOS,IPSI,IACL,NACL,INA,IVEL)
       NVEL=IVEL+NACL*DT
       IF(NVEL)55,1,1
55     NVEL=0$GO TO 1
5      NVEL=IVEL
1      VEL=NVEL
       NPOS=IPOS+(VEL*DT+0.9)
       IC1=INSP1-5
       IF(NPOS.GT.IC1   )133,83
133    IF(IDUM(6).EQ.1)68,131
131    IF(NPOS.GT.INSP2)83,84
84     IC1=IPSI-NPOS
       IF(IC1.GT.5)83,85
85     NPOS=IPSI-5
       VEL=NVEL=0
C     CHECK IF VEHICLE AT INTERSECTION
83     IF(IDUM(6).EQ.1)68,62
62     IF(NPOS.EQ.INSP1)61,72
61     IDUM(6)=1
       IDUM(11)=1
       IF(IT.EQ.1)173,172
173    IDUM1(1)=1$IDUM1(2)=2$GO TO 171
172    IF(IT.EQ.3)174,371
371    IDUM1(3)=1$GO TO 171
174    IDUM1(1)=2$IDUM1(2)=1
171    IVII=IVII+1
       DO 59 II=1,10
```

```
      IF(INSEC(II).EQ.0)58,59
59    CONTINUE
      PRINT 57
57    FORMAT(*    LOGIC ERROR 1*)
      CALL EXIT
58    INSEC(II)=ISI.O.MASK(9)
C    BITS 58,59 USED TO INDICATE VEHICLE IN INTERSECTION
C     CHECK IF VEHICLE FIRST VEH AT INSECTION
68    IF(INSEC(11).EQ.ISI)103,104
104   IF(INSEC(11).EQ.0)101,102
101   INSEC(11)=ISI$IDUM(7)=1$GO TO 103
102   IDUM(7)=2$GO TO 103
103   CONTINUE
C    K1 USED FOR CAR ON RIGHT
C    K2 USED FOR CAR ON LEFT
C     K3 USED FOR CAR STRAIGHT AHEAD
      K1=K2=K3=2
65    DO 66 IJ1=1,10
      IF(INSEC(IJ1))67,66,67
67    IRO=INSEC(IJ1)
      IRO=IRO.A.777B
      IF(IRO.EQ.ISI)66,320
320   ID1=IVCH(IRO).A.MASK(4)
      ID1=ID1/2**K(4)
      ID3=ID2=ID1
      ID4=ID
      ID5=IVEH(IRO).A.3000000000B
      ID5=ID5/2**27
      ID6=IVCH(IRO).A.MASK(14)
      ID6=ID6/2**K(14)
      GO TO (111,110)K1
C    CAR ON RT
110   ID1=ID1+1$ID1=ID1.A.3
      IF(ID.EQ.ID1)112,111
112   K1=1
      IDUM(12)=ID5
C    CHECK FOR CAR ON RT TURNING RT OR LEFT
      IF(ID6.EQ.1)300,301
300   IDUM(15)=1$IDUM(16)=2$GO TO 111
301   IF(ID6.EQ.3)302,111
302   IDUM(16)=1$IDUM(15)=2
C    CAR ON LT
111   GO TO (116,114)K2
114   ID4=ID4+1$ID4=ID4.A.3
      IF(ID2.EQ.ID4)115,116
115   K2=1
      IDUM(13)=ID5
C    CHECK FOR CAR ON LT TURNING RT OR LEFT
      IF(ID6.EQ.1)304,305
304   IDUM(17)=1$IDUM(18)=2$GO TO 116
305   IF(ID6.EQ.3)306,116
306   IDUM(18)=1$IDUM(17)=2$GO TO 116
116   GO TO (66,117)K3
117   IF(ID3.LE.2)118,119
118   ID3=ID3+2$GO TO 120
```

```
119    ID3=ID3-2
120    ID3=ID3.A.3
       IF(ID3.EQ.ID)121,66
121    K3=1
       IDUM(14)=ID5
C      CHECK FOR CAR ST AHEAD TUNRING RT OR LT
       IF(ID6.EQ.1)154,155
154    IDUM(19)=1$IDUM(20)=2$GO TO 66
155    IF(ID6.EQ.3)156,66
156    IDUM(20)=1$IDUM(19)=2$GO TO 66
66     CONTINUE
       IDUM(8)=K1
       IDUM(9)=K2
       IDUM(10)=K3
       IF(IDUM(6).EQ.1)71,72
71     IF(NPOS.EQ.INSP1)72,78
78     IDUM(11)=IDUM(12)=IDUM(13)=IDUM(14)=2
       IF(ISI.EQ.INSEC(11))105,106
105    IDUM(7)=2$INSEC(11)=0
106    IF(NPOS.GT.INSP2)73,72
C     VEHICLE LEFT INTERSECTION
73     DO 74 IJ1=1,10
       IC1=INSEC(IJ1).A.777B
       IF(IC1.EQ.ISI)75,74
74     CONTINUE
       PRINT 77
77     FORMAT(*  LOGIC ERROR 2*)
       CALL EXIT
75     INSEC(IJ1)=0
       IVII=IVII-1
C     CAR TAKEN OUT OF INTERSECTION
       IDUM(6)=2
       IDUM(8)=IDUM(9)=IDUM(10)=2
       IDUM(15)=IDUM(16)=IDUM(17)=IDUM(18)=IDUM(19)=IDUM(20)=2
72     CONTINUE
C  CHECK IF VELOCITY G.T. DESIRED VELOCITY
       IF(NVEL.GT.IDVL)21,22
22     IF(NVEL.EQ.IDVL)23,24
21     IDUM(1)=2$IDUM(2)=1$GO TO 25
23     IDUM(1)=1$IDUM(2)=2$GO TO 25
24     IDUM(1)=2$IDUM(2)=2
C  CHECK IF VELOCITY EQUAL TO ZERO
25     IF(NVEL)27,28,27
27     IDUM(5)=2$GO TO 29
28     IDUM(5)=1
29     IDST=IPSI-NPOS
C   CHECK IF DISTANCE =0,IE. PASS CONDITION
       IF(IDST)13,13,12
13     IF(IVCH(INA))17,16,16
C   STOP VEHICLE
16     PRINT 41,ISI,INA
41     FORMAT(*   COLLISION   ISI=  *,I6,*INA= *,I6)
       NPOS=IPSI-5
       IDST=5$NVEL=0
       GO TO 12
```

```
17      IC1=IVCH(INA).A.K(9)
        IF(IC1.EQ.K(9))42,43
42      CONTINUE
        IVC=IVC.O.770000000000
        IJ=IVC.A.MASK(20)
        IJ=IJ/2**K(20)
        IJ=IJ+1
        IF(IT.EQ.3)330,331
330     IR=9$GO TO 343
331     IF(IT.EQ.1)340,341
340     IR=8$GO TO 343
341     IR=10
343     CONTINUE
        IFCT(IR,2,IJ)=IFCT(IR,2,IJ)-1
C      SET VEHICLE EXIT FLAG
        GC1=ITIM(1)/CTT(6)
        GC2=ITIM(2)/CTT(6)
        GC3=ITIM(3)/CTT(6)
        IF(GC2.EQ.0)368,369
368     GC2=1
        IR=IR-6
        IFCT(IR,1,IJ)=IFCT(IR,1,IJ)+1$IFCT(IR,2,IJ)=IFCT(IR,2,IJ)+1
369     CONTINUE
        IDT=IVCH1(ISI).A.MASK(10)
        IDT=IDT/2**K(10)
        IVCH1(ISI)=K(9)
        PRINT 510,IDT,ISI,IT,IR,RT
510     FORMAT(//*  VEHICLE LEFT SYSTEM     IDENT=*,I5,
       1* ISI=*,I5,* TURN TYPE=*,I2,* DIRECTION=*,I2,* TIME=*,F10.5)
        PRINT 511,GC1,GC2,CTT(3),GC3
511     FORMAT(*  TIME IN SYSTEM=*F10.5,  * STOP TIME DELAY=* ,F10.5,
       1*  TIME DELAY FOR VEHICLES UNDER*,F10.5,*[illegible]=*,F10.5)
        IVC=IVC.O.MASK(11)
        IC1=INA*2**K(12)
        IVC=IVC.O.IC1
        GO TO 373
501     CONTINUE
43      IC1=IVCH(INA).A.MASK(9)
        IF(IC1.EQ.MASK(9))80,81
81      IVC=IVC.O.MASK(11)
C       SET OBJECT INDEX,PASS
C     TURN LOGIC
C      SET PASS FLAG
C      PUT INA IN PASS  INA
C      PICK UP STOP SIGN ON RT/LT AND OBTAIN NEW INA
C      SET ISI IN STOP SIGN INA
C      CHANGE NPOS
C      CHANGE DIRECTIONS
        IF(IT.EQ.0)161,160
C      VEHICLE TURNING
160     IC1=INA*2**K(12)
        IVC=IVC.O.IC1
        DO 164 I=1,12
        IC1=IVCH(I).A.MASK(16)
        IF(IC1.EQ.MASK(15))165,164
```

```
165    IF(IT.EQ.1)162,163
C     RT TURN, PICK UP STOP SIGN ON LT
162    ID2=IVCH(I).A.MASK(4)
       ID5=ID2
       ID2=ID2/2**K(4)
       ID4=ID
       ID4=ID4+1$ID4=ID4.A.3
       IF(ID2.EQ.ID4)167,164
166    CALL NOI(I,IC1)
       IVCH(I)=IVCH(I).A..N.MASK(5)
       IC2=ISI*2**K(5)
       IVCH(I)=IVCH(I).O.IC2
C     SET INA PASS IGNORE
       IVCH(I)=IVCH(I).O.MASK(21)
       INA=IC1
       GO TO 170
C      LT TURN PICK UP STOP SIGN ON RT
163    ID1=IVCH(I).A.MASK(4)
       ID5=ID1
       ID1=ID1/2**K(4)
       ID1=ID1+1$ID1=ID1.A.3
       IF(ID.EQ.ID1)168,164
167    NPOS=INSP2-5$GO TO 166
168    NPOS=INSP1+10$GO TO 166
170    IVC=IVC.A..N.MASK(4)
       IVC=IVC.O.ID5
       IDST=IPSI-NPOS
       GO TO 12
164    CONTINUE
       PRINT 200
200    FORMAT(*    LOGIC ERROR 3*)
       CALL EXIT
161    IC1=INA*2**K(12)
       IVC=IVC.O.IC1
C     SET PASS IGNORE
       IVCH(INA)=IVCH(INA).O.MASK(21)
C   PICK UP NEW NEAREST OBJECT INDEX
C     IC1=NEW INDEX
30     CALL NOI(INA,IC1)
C      SET UP OBJECT PASSED
       IC2=ISI*2**K(5)
       IVCH(INA)=IVCH(INA).A..N.MASK(5)
       IVCH(INA)=IVCH(INA).O.IC2
       INA=IC1
12     CONTINUE
       ITVEL=IVCH(INA).A.MASK(2)
       ITVEL=ITVEL/2**K(2)
       ITVEL=IABS(NVEL-ITVEL)
C     NEG IVCH1 INDICATES VEHICLE OUT OF INTERSECTION
C      CHECK IF NEAREST OBJECT OUTPUT
       IC1=IVCH(INA).A.K(9)
       IF(IC1.EQ.K(2))30,130
130    CONTINUE
46     CALL CCD(ITVEL,IDST,ICD)
       IF(IDST.GT.ICD)30,31
```

```
31      IF(IDST.EQ.ICD)32,33
30      IDUM(3)=2$IDUM(4)=1$GO TO 35
32      IDUM(3)=1$IDUM(4)=2$GO TO 35
33      IDUM(3)=2$IDUM(4)=2
35      IPOS=NPOS$IVEL=NVEL
36      CONTINUE
C     SET PROPER COUNT IN SUMMARY PARAMETERS
        IJ=IVC.A.MASK(20)
        IJ=IJ/2**K(20)
        IJ=IJ+1
500     IFCT(1,1,IJ)=IFCT(1,1,IJ)+FDT
        IFCT(1,2,IJ)=IFCT(1,2,IJ)+FDT
C   INCREMENT TIME INDICATORS
        FVL=CTT(3)*(22./15.)
        ITIM(1)=ITIM(1)+FDT
        IF(IVEL.EQ.0)370,375
370     ITIM(2)=ITIM(2)+FDT
        IF(IT.EQ.3)502,503
502     IFCT(3,1,IJ)=IFCT(3,1,IJ)+FDT$IFCT(3,2,IJ)=IFCT(3,2,IJ)+FDT
        GO TO 506
503     IF(IT.EQ.1)504,505
504     IFCT(2,1,IJ)=IFCT(2,1,IJ)+FDT$IFCT(2,2,IJ)=IFCT(2,2,IJ)+FDT
        GO TO 506
505     IFCT(4,1,IJ)=IFCT(4,1,IJ)+FDT$IFCT(4,2,IJ)=IFCT(4,2,IJ)+FDT
506     CONTINUE
375     IF(IVEL.GT.FVL)373,372
372     ITIM(3)=ITIM(3)+FDT
        IF(IT.EQ.3)602,603
602     IFCT(6,1,IJ)=IFCT(6,1,IJ)+FDT$IFCT(6,2,IJ)=IFCT(6,2,IJ)+FDT
        GO TO 606
603     IF(IT.EQ.1)604,605
604     IFCT(5,1,IJ)=IFCT(5,1,IJ)+FDT$IFCT(5,2,IJ)=IFCT(5,2,IJ)+FDT
        GO TO 606
605     IFCT(7,1,IJ)=IFCT(7,1,IJ)+FDT$IFCT(7,2,IJ)=IFCT(7,2,IJ)+FDT
606     CONTINUE
373     IVCH(ISI)=IVC
30      CALL SETUP(ISI)
        RETURN
        END
        SUBROUTINE NOI(ISI,INN)
        COMMON/MSK/MASK(25),K(25),N(25),NR,DT,INSP1,INSP2
        COMMON/STACK/IVEH(100),IVEH1(100),IVCH(100),IVCH1(100),IY
        COMMON/CHV/IPOS,IVEL,IDVL,ID,INA,IACL,IPS1,IDUM(20),INSEC(11)
       1,IT,IDUM1(10),ITIM(3),FDT,IXIT
C       PICK UP OBJECT INDEX
        IVC=IVCH(ISI)
        INO=IVC.A.MASK(5)
        INO=INO/2**K(5)
5       INN=INO
C   ACTUAL NEAREST OBJECT INDEX=INN
2       IC1=IVCH(INN).A.MASK(11)
        IF(IC1.EQ.MASK(11))12,13
12      IC1=IVCH(INN).A.MASK(12)
        IC5=IVC.A.MASK(21)
        IF(IC5.EQ.MASK(21))23,24
```

```
23     IVC=IVC.A..N.MASK(21)
       GO TO 13
24     CONTINUE
       IVCH(INN)=IVCH(INN).A..N.MASK(11)
       IVCH(INN)=IVCH(INN).A..N.MASK(12)
       IF(IVCH1(INN).EQ.K(9))21,22
21     IXIT=INN
22     INN=IC1/2**K(12)
13     IPSI=IVCH(INN).A.MASK(1)
C   ADD VEHICLE LENGTH
       IVLGT=IVCH1(INN).A.MASK(13)
       IPSI=IPSI-IVLGT
       RETURN
       END
       SUBROUTINE ORCV(ISI)
       COMMON/CHV/IPOS,IVEL,IDVL,ID,INA,IACL,IPSI,IDUM(20),INSEC(11)
      1,IT,IDUM1(10),ITIM(3),EDT,IXIT
       COMMON/SYS/ISYS1(60),ISYSY(60)
       COMMON/STACK/IVEH(100),IVEH1(100),IVCH(100),IVCH1(100),IY
       COMMON/MSK/MASK(25),K(25),N(25),NR,DT,INSP1,INSP2
C      IPOS=CURRENT POSITION
C      IT = TURN INDICATOR
C      IVEL=CURRENT SPEED
C      IDVL=DESIRED SPEED
C      ID=DIRECTION
C      INO=NEAREST OBJECT INDEX(TEMP)
C      INA=ACTUAL OBJECT INDEX
C      IACL=ACCELERATION INDICATOR
C      IPSI=POSITION OF NEAREST OBJECT
C   OBTAIN CHARACTER VALUES FOR,
       IVC=IVCH(ISI)
C   POSITION
       IPOS=IVC.A.MASK(1)
C    SPEED
C     NOFVP=NUMBER OF IVEH PARAMETERS
       NOFVP=10
       CALL NOI(ISI,INA)
       IF(IVC)8,9,9
9      CONTINUE
       IVEL=IVC.A.MASK(2)
       IVEL=IVEL/2**K(2)
C    DESIRED SPEED
       IDVL=IVC.A.MASK(3)
       IDVL=IDVL/2**K(3)
C    DIRECTION
       ID=IVC.A.MASK(4)
       ID=ID/2**K(4)
C    NEAREST OBJECT INDEX
       IT=IVC.A.MASK(14)
       IT=IT/2**K(14)
C     TIME INDICATORS
       IK=17
       DO 6 II=1,3
       ITIM(II)=IVCH1(ISI).A.MASK(IK)
       ITIM(II)=ITIM(II)/2**K(IK)
```

```
6     IK=IK+1
C   ACCELERATION
      IACL=IY.A.MASK(7)
      IACL=IACL/2**K(7)
C     OBTAIN IVEH PARAMETERS  IDUM(1)TO IDUM(8)
      IK=7
      DO 10 II=1,20
      IDUM(II)=IVEH(ISI)/2**IK
      IDUM(II)=IDUM(II).A.3
10    IK=IK+2
      IK=0
      DO 20 II=1,NOFVP
      IDUM1(II)=IVEH1(ISI)/2**IK
      IDUM1(II)=IDUM1(II).A.3
20    IK=IK+2
8     CONTINUE
      RETURN
      END
      SUBROUTINE SETUP(ISI)
      COMMON/MSK/MASK(25),K(25),N(25),NR,DT,INSP1,INSP2
      COMMON/SYS/ISYSI(60),ISYSY(60)
      COMMON/STACK/IVEH(100),IVEH1(100),IVCH(100),IVCH1(100),IY
      COMMON/CHV/IPOS,IVEL,IDVL,ID,INA,IACL,IPSI,IDUM(20),INSEC(11)
     1,IT,IDUM1(10),ITIM(3),FDT,IXIT
C  SET UP CHARACTER VALUES FOR,
      IVC=IVCH(ISI)
      IF(IXIT.EQ.0)22,21
21    IVCH1(IXIT)=IVEH(IXIT)=IVEH1(IXIT)=0
      IVCH(IXIT)=-1$IXIT=0
22    CONTINUE
C   NEAREST OBJECT INDEX
      IVC=IVC.A..N.MASK(5)
      INA=INA*2**K(5)
      IVC=IVC.O.INA
      IF(IVC)7,8,8
8     CONTINUE
C   POSITION
      IVC=IVC.A..N.MASK(1)
      IVC=IVC.O.IPOS
C   SPEED
C    NOFVP=NUMBER OF IVEH PARAMETERS
      NOFVP=10
      IVC=IVC.A..N.MASK(2)
      IVEL=IVEL*2**K(2)
      IVC=IVC.O.IVEL
C    TIME INDICATORS
      IK=17
      DO 6 II=1,3
      IVCH1(ISI)=IVCH1(ISI).A..N.MASK(IK)
      ITIM(II)=ITIM(II).A.77B
      ITIM(II)=ITIM(II)*2**K(IK)
      IVCH1(ISI)=IVCH1(ISI).O.ITIM(II)
6     IK=IK+1
C     PICK UP IVEH PARAMETERS
C     IDUM(1) TO IDUM(8)
```

```
      IVEH(ISI)=IVEH(ISI).A..N.MASK(8)
      IVEH1(ISI)=IVEH1(ISI).A..N.K(15)
      IK=7
      DO 10 II=1,20
      IDUM(II)=IDUM(II)*2**IK
      IVEH(ISI)=IVEH(ISI).O.IDUM(II)
10    IK=IK+2
      IK=0
      DO 20 II=1,NOFVP
      IDUM1(II)=IDUM1(II)*2**IK
      IVEH1(ISI)=IVEH1(ISI).O.IDUM1(II)
20    IK=IK+2
7     CONTINUE
      IVCH(ISI)=IVC
      RETURN
      END
      SUBROUTINE ACCEL(VPOS,OPOS,ACCI,NACL,INA,IVEL)
      COMMON/CRDP/H,A,P
      INTEGER VPOS,OPOS,ACCI,DIFF
C    NEG ACC=4, POS ACC = 1
      IF(ACCI.EQ.4)8,4
4     NACL=2
      RETURN
8     DIFF=OPOS-VPOS
      IF(DIFF.LE.H)32,12
32    SLOPE=-6./400.
      NACL=SLOPE*DIFF+6
23    NACL=-NACL
      RETURN
12    NACL=0
      RETURN
      END
      SUBROUTINE CCD(NVEL,IDST,ICD)
      COMMON/CRDP/H,A,P
      ICD=-(A/P)*SQRT(P**2-NVEL**2)+H
      RETURN
      END
```

MODEL OPERATION VARIABLES AND
DRIVER RESPONSE FUNCTIONS

```
0001101       1.      30.       10.
   54   100.      .5
  22 1                        10
  22 2                        12
2  1                          10
  1 1 122 1           1       10
  1 1 121 1 1         1       10
  1 1 112 11          1       10
  1 1 111 111         1       10
  1 1 1 1 1     1  21         10
  1 1 1 1 1  1  1  11         10
  1 1 1 1 1      1 21         10
  1 1 1 1 1  1   1 11         10
  1 1 1 1 1 1   22 21         10
  1 1 1 1 1 11  22 11         10
  1 1 1 2 1        21         10
  1 1 1 2 1  1     11         10
  1 1 2 1 1     1  21         10
  1 1 2 1 1      1 21         10
  1 1 2 2 1        21         10
  1 1 12221          1        10
  1 1 12211  1       1        10
  1 1 12211  2     1 1        10
  1 1 12121 1        1        10
  1 1 12111 11       1        10
  1 1 12111 12     1 1        10
  1 1 12121 2   1    1        10
  1 1 12111 21  1    1        10
  1 1 12111 22  1  1 1        10
  1 1 112211         1        10
  1 1 112111 1       1        10
  1 1 112111 2     1 1        10
  1 1 1112111        1        10
  1 1 11111111       1        10
  1 1 11111112     1 1        10
  1 1 1112112   1    1        10
  1 1 11111121  1    1        10
  1 1 11111122  1  1 1        10
  1 1 112212  1      1        10
  1 1 112112 11      1        10
  1 1 112112 21    1 1        10
  1 1 1112121 1      1        10
  1 1 111112111      1        10
  1 1 111112121    1 1        10
  1 1 1112122 1 1    1        10
  1 1 111112211 1    1        10
  1 1 111112221 1  1 1        10
  1 1  2221          1        10
  1 1  2211        1 1        10
  1 1  2121     1    1        10
  1 1  2111     1  1 1        10
  1 1  1221   1      1        10
  1 1  1211   1    1 1        10
  1 1  1121   1 1    1        10
  1 1  1111   1 1  1 1        10
```

1 1 122 1 10

STOP SIGN AND DUMMY I/O OBJECTS

0	0			5	0	11	0	0	0
0	0	0	1	6	0	11	0	0	0
0	0	0	2	7	0	11	0	0	0
0	0	0	3	8	0	11	0	0	0
450	0	0	0	9	0	1	0	0	0
450	0	0	1	10	0	1	0	0	0
450	0	0	2	11	0	1	0	0	0
450	0	0	3	12	0	1	0	0	0
900	0	0	0	9	0	101	0	0	0
900	0	0	1	10	0	101	0	0	0
900	0	0	2	11	0	101	0	0	0
900	0	0	3	12	0	101	0	0	0

VEHICLE INPUTS

0	6	30	3	0	18	0	0	1	12.[illegible]3821
0	22	30	3	0	22	0	1	2	82.69[illegible]73
0	12	30	2	0	18	0	1	3	98.94071
0	23	30	0	0	22	0	0	4	127.71970
0	20	30	2	0	18	0	0	5	143.14242
0	24	30	2	0	22	0	3	6	161.74544
0	22	30	3	0	18	0	1	7	175.68231
0	29	30	0	0	22	0	0	8	186.18342
0	26	30	3	0	18	0	0	9	193.32943
0	19	30	2	0	22	0	0	10	201.71013
0	23	30	2	0	18	0	0	11	219.34115
0	20	30	2	0	22	0	0	12	236.54777
0	27	30	0	0	18	0	3	13	248.96370
0	7	30	1	0	22	0	1	14	259.54250
0	27	30	3	0	18	0	1	15	284.97500
0	29	30	3	0	22	0	1	16	296.08924
0	22	30	0	0	18	0	1	17	303.56197
0	21	30	2	0	22	0	0	18	310.58069
0	4	30	3	0	18	0	3	19	316.14351
0	29	30	2	0	22	0	0	20	349.91675
0	23	30	2	0	18	0	0	21	360.56956
0	20	30	0	0	22	0	0	22	362.89465
0	20	30	3	0	18	0	0	23	368.41815
0	1	30	0	0	22	0	1	24	393.64830
0	26	30	2	0	18	0	0	25	457.95270
0	21	30	0	0	22	0	0	26	459.98123
0	16	30	0	0	18	0	0	27	464.38812
0	21	30	2	0	22	0	1	28	482.40056
0	22	30	0	0	18	0	3	29	501.77726
0	29	30	3	0	22	0	0	30	520.44874
0	23	30	0	0	18	0	0	31	521.18535
0	22	30	3	0	22	0	3	32	523.88406
0	11	30	0	0	18	0	3	33	543.48814
0	9	30	0	0	22	0	1	34	559.36953
0	24	30	0	0	18	0	3	35	571.27887
0	26	30	0	0	22	0	0	36	590.12383
0	2	30	3	0	18	0	1	37	606.76771
0	20	30	0	0	22	0	3	38	606.79483
0	21	30	0	0	18	0	0	39	624.62549
0	21	30	1	0	22	0	0	40	635.33424
0	17	30	2	0	18	0	0	41	646.60054
0	25	30	0	0	22	0	0	42	653.36115
0	28	30	3	0	18	0	3	43	667.57602
0	24	30	0	0	22	0	0	44	679.46613
0	22	30	0	0	18	0	3	45	694.50586
0	20	30	3	0	22	0	1	46	704.38267
0	27	30	0	0	18	0	3	47	711.68220
0	23	30	0	0	22	0	3	48	730.32010
0	27	30	0	0	18	0	1	49	742.88147
0	28	30	0	0	22	0	0	50	759.32332
0	26	30	3	0	18	0	1	51	771.67385
0	21	30	3	0	22	0	0	52	786.15564
0	28	30	0	0	18	0	1	53	791.34040
0	20	30	3	0	22	0	0	54	802.51875
0	13	30	2	0	18	0	0	55	803.99447

0	10	30	1	0	22	0	3	56 8[illegible]4.74182
0	20	30	0	0	18	0	0	57 80[illegible].71550
0	23	30	0	0	22	0	3	58 817.28645
0	26	30	2	0	18	0	0	59 823.10976
0	22	30	0	0	22	0	0	60 831.31568
0	22	30	3	0	18	0	3	61 834.40351
0	6	30	3	0	22	0	0	62 846.75857
0	12	30	2	0	18	0	1	63 856.76708
0	25	30	0	0	22	0	3	64 865.86296
0	27	30	0	0	18	0	0	65 873.85039
0	1	30	3	0	22	0	1	66 885.72257
0	21	30	0	0	18	0	3	67 901.79820
0	27	30	2	0	22	0	0	68 920.24072
0	14	30	0	0	18	0	0	69 924.18243
0	22	30	0	0	22	0	1	70 937.14587
0	28	30	2	0	18	0	1	71 955.18183
0	20	30	3	0	22	0	1	72 957.98462
0	24	30	3	0	18	0	0	73 977.96794
0	22	30	0	0	22	0	3	74 997.76280
0	22	30	2	0	18	0	0	751012.31477
0	28	30	0	0	22	0	0	761026.17310
0	7	30	3	0	18	0	0	771028.29164
0	22	30	0	0	22	0	0	781033.59255
0	20	30	3	0	18	0	0	791048.73637
0	22	30	0	0	22	0	1	801057.89274
0	1	30	0	0	18	0	0	811077.77946
0	26	30	2	0	22	0	3	821078.68797
0	22	30	3	0	18	0	0	831094.09251
0	20	30	3	0	22	0	1	841111.98520
0	28	30	0	0	18	0	3	851121.20429
0	26	30	1	0	22	0	0	861136.60642
0	22	30	0	0	18	0	1	871149.92704
0	22	30	0	0	22	0	0	881164.16336
0	29	30	2	0	18	0	3	891173.71136
0	24	30	3	0	22	0	0	901189.59340
0	23	30	3	0	18	0	3	911204.14492
0	27	30	3	0	22	0	0	921214.24440
0	20	30	3	0	18	0	0	931233.14090
0	22	30	2	0	22	0	0	941235.87668
0	27	30	2	0	18	0	0	951245.97064
0	21	30	0	0	22	0	0	961256.17234
0	12	30	3	0	18	0	1	971258.74820
0	9	30	0	0	22	0	3	981273.26612
0	28	30	2	0	18	0	1	991289.91978
0	27	30	0	0	22	0	3	1001302.43965

END

REFERENCES

1. Falls, M. P., Anthony Petry, Roger S. Walker, G. G. Zeigler, Engineering Task Report for ARTRAC Display and Control, Vol. 1, RCA Systems Engineering, White Sands Missile Range, New Mexico, October, 1966.

2. Barnett, C. C., Jr., and Associates, The Future of the Computer Utility, The American Management Association, 1967, pp. 27.

3. Vodrazka, Walter C., Queueing Theory and Its Applications to Some Highway Traffic Situations (unpublished), June, 1965.

4. Neymark, Yu. I., M. A. Fedotkin, and A. M. Preobrazhenskaya, "The Operation of a Feedback Automation which Controls Traffic Movement at an Intersection," Engineering Cybernetics, No. 5 1968, pp. 114.

5. Lewis, Russell M., The Simulation of Traffic Flow to Obtain Volume Warrants for Intersection Control, Purdue University, Lafayette, Indiana, September, 1962.

6. Fox, Phyllis, and Frederick G. Lehman, Safety in Car Following, Newark College of Engineering, Newark, New Jersey, 1967. Report to the Public Health Service, U. S. Department of Health, Education and Welfare under Research Grant AC 00236.

7. Uniform Vehicle Code and Model Traffic Ordinance, Revised - 1968, National Committee on Uniform Traffic Laws and Ordinances, 525 School Street, S. W., Washington, D. C. 20024, pp. 145-146.

8. Control Data 6400/6500/6600 Computer Systems Fortran Reference Manual, Control Data Corporation, Minneapolis, Minnesota, May, 1967.

9. Friedlander, Gordon D., "Computer-Controlled Vehicular Traffic," IEEE Spectrum, February, 1969, pp. 30.

10. Kohnert, Dietrich, and Achim Musiol, "Traffic Signal Control by Computer," Control Engineering, June, 1969, pp. 115.

11. Hsu, Jay C., and Andrew U. Meyer, Modern Control Principles and Applications, McGraw-Hill, Inc., 1968.

12. Eveleigh, Virgil W., Adaptive Control and Optimization Techniques, McGraw-Hill, Inc., 1967.

13. Traffic Engineering Handbook, Institute of Traffic Engineers, Washington, D.C., pp. 27.