UMTA-MD-06-0047-84-1
JHU/APL CP 086/TPR-047

U.S. Department of Transportation

Urban Mass Transportation Administration

A NETWORK MANAGEMENT SIMULATION OF AUTOMATED GUIDEWAY RAPID TRANSIT SYSTEMS UNDER VEHICLE-FOLLOWER CONTROL

H. Y. Chiu
D. L. Kershner
P. J. McEvaddy

Applied Physics Laboratory
The Johns Hopkins University
Laurel, Md. 20707

October 1984
Final Report

UMTA Technical Assistance Program

Technical Report Documentation Page

1. Report No.
UMTA-MD-06-0047-84-1

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
A Network Management Simulation of Automated Guideway Rapid Transit Systems Under Vehicle-Follower Control

5. Report Date
October 1984

6. Performing Organization Code

7. Authors
H. Y. Chiu, D. L. Kershner, and P. J. McEvaddy

8. Performing Organization Report No.
JHU/APL CP 086/TPR-047

9. Performing Organization Name and Address
The Johns Hopkins University
Applied Physics Laboratory
Johns Hopkins Road
Laurel, MD 20707

10. Work Unit No. (TRAIS)
MD-06-0047

11. Contract or Grant No.
DOT-UT-90042

13. Type of Report and Period Covered
Final Report

12. Sponsoring Agency Name and Address
Department of Transportation
Urban Mass Transportation Administration

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract
A computer simulation was developed for the evaluation of station operations, vehicle management techniques, and vehicle-follower longitudinal control algorithms in a full automated guideway rapid transit network context. The simulation was then exercised to investigate the performance characteristics of interactions among vehicle control, network management, and station models.

17. Key Words
Transportation Network Simulation
Automated Guideway Transit
Downtown People Mover
Vehicle Management, Vehicle Follower Control
Network Performance Assessment

18. Distribution Statement
This document is available to the public through the National Technical Information Service
Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	155	

ABSTRACT

A computer simulation was developed for the evaluation of station operations, vehicle management techniques, and vehicle-follower longitudinal control algorithms in a full automated guideway rapid transit network context. The simulation was then exercised to investigate the performance characteristics of and interactions among vehicle control, network management, and station models.

CONTENTS

ILLUSTRATIONS

TABLES

1.0 INTRODUCTION

The use of automated guideway rapid transit (AGRT) has been proposed as a means of alleviating many of the problems associated with conventional transportation modes in urban areas. The concept of AGRT requires the deployment of remotely controlled vehicles in a dedicated guideway network. Many control strategies have been proposed to handle the simultaneous longitudinal control of possibly hundreds of these vehicles. One generic approach is vehicle-follower, in which vehicle state information is used to command vehicle spacing and velocity of the immediately following vehicle. Vehicle-follower control has been extensively examined at the individual vehicle interaction level (Refs. 1 through 3); however, the network effects of vehicle-follower are relatively unassessed.

The principal objective of this study is to examine the network level behavior of a vehicle-follower approach to longitudinal control in AGRT systems. Some questions to be addressed include the following:

1. What is the practical line capacity that can be achieved, and to what extent does system service degrade as this capacity is approached?

2. In Ref. 1, vehicle-follower control tended to increase the "bunchiness" (i.e., the percentage of flow at minimum headway) of vehicle flow. Does this tendency have a detrimental effect on successive downstream merges and, if so, does resolution of the problem require the implementation of a network level debunching function?

3. What is the nature of the interaction of station operations with mainline operations using vehicle-follower control, and what is the practical capacity of stations to process vehicles?

4. What are the energy consumption characteristics of a system using vehicle-follower control?

To answer these questions, we need to evaluate a vehicle-follower control system design within the context of a representative AGRT system network. The vehicle-follower control system developed at APL and discussed in Ref. 3 as "Suboptimal II" was selected for the study. A network level simulation focusing entirely on the "vehicle side" of system operation was developed as the tool for conducting this study.

Network simulations for evaluating AGRT systems have been developed in the past. These simulations typically were discrete event models that were oriented toward the network management and passenger service aspects of AGRT. A review of the modeling approaches used in these simulations lead to the conclusion that those approaches did not satisfy the requirements of this study. In general, vehicle movements on the system guideway were simulated in discrete event models by shifting vehicles from a queue of one link to the queue associated with the next link. Neither the dynamics of vehicle-to-vehicle interactions (which are fundamental to vehicle-follower control) nor the potential impact of frequent on-line accelerations and decelerations on energy consumption are easily captured by such an approach.

A simulation development effort was necessary for the study to proceed. In order to meet the study objectives, the vehicle-follower control logic had to be included in the simulation of vehicle mainline operations. This required a continuous (or time-step) modeling approach for the control system development studies of Ref. 3. Recognizing that existing network simulations were computer resource intensive, and that this modeling approach for mainline operations would only increase the computational burden, the passenger side of operations and its associated computations were ignored. Although passenger arrivals and associated trip demands represent the "driver" in most network simulations, a fixed route, fixed frequency type of service structure can be employed with the route frequencies themselves used as means of maintaining the flow of vehicles through the network. This form of service has been used in most simulation studies for peak period demand operations and, because of station operation complexities (from the system and passenger's viewpoints; Ref. 4), may be the preferable mode of operation for initial AGRT deployment.

A final simulation design decision concerned modeling of the vehicle side of station operations. It was decided that a discrete event modeling approach to shifting vehicles (1) from queue to queue within a station and (2) from mainline exit to the initiation of vehicle acceleration for station egress would require substantially less computation than a continuous modeling approach, while preserving a level of fidelity sufficient for the purposes of the study. The direction for the simulation development effort was, therefore, to design and program a combined continuous/discrete-event simulation incorporating vehicle-following logic to control mainline vehicle motion, using predetermined service route frequencies (i.e., vehicles per hour per route) in lieu of arriving passengers to drive the vehicle flow.

2.0 SUMMARY

The network behavior of an AGRT system under vehicle-follower control was studied. To perform this study, a network management simulation incorporating both discrete-event and time-step approaches was developed. Vehicle dynamics under vehicle-follower control was faithfully maintained using the time-step approach, while the discrete-event approach permitted efficient execution of vehicle management and station operations.

2.1 SIMULATION OVERVIEW

The objectives of the study required a simulation that models the dynamics of vehicle flow under vehicle-follower control, while keeping computer resources (e.g., memory and central processing unit time) within reasonable limits to prevent excessive run costs. Furthermore, the simulation was required to model the management of vehicles both on the guideway and within stations (including berth assignment and queue-shifting logic) as well as to assess energy consumption characteristics of the system.

The simulation architecture selected to satisfy the study objectives is shown in Fig. 2.1. An event-scheduling approach characteristic of discrete-event simulations was used for pri-

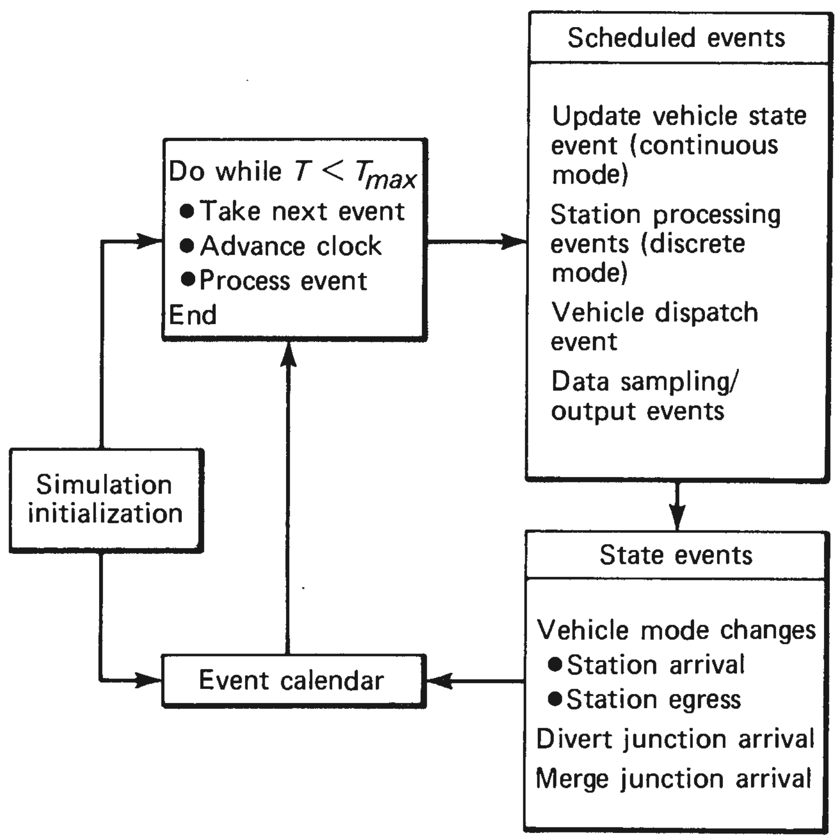

Fig. 2.1 General network management simulation architecture.

mary control of the sequence of execution. Vehicle mainline operations (simulated using a continuous or time-step approach) were integrated into this event-based process by simply scheduling state update events at regular intervals equal to the selected time-step interval. As indicated in Fig. 2.1, the types of events that were scheduled included station-related events for vehicles in the discrete-event mode, the state-update event for vehicles in continuous mode, and state-related events to transfer vehicles from one mode to the other and to actuate logic for merge control and network path selection.

2.1.1 Vehicle Dynamics

The network management simulation uses two approaches to handle individual vehicle dynamics—continuous and discrete. Vehicles on the main guideway (or mainline) function in a "continuous" manner; all other vehicles operate in a "discrete" mode.

The continuous approach updated vehicle velocity and position by integrating these states based on acceleration commands of a regulation controller. The regulation controller (Ref. 3) used a vehicle-follower, state-constrained approach. The controller generates an acceleration command derived from the error between the actual spacing and a desired spacing based on kinematics and operational constraints. When vehicles are spaced far apart, vehicle control reverts to an open-loop mode of operation, the velocity-command mode. Here the vehicles achieve and maintain the maximum link speeds until the vehicles either reach their destination station or overtake another vehicle, forcing them back into the vehicle-follower mode.

2.1.2. Vehicle Management

The problem of effectively managing the vehicle flows in an AGRT network has both long-term, network-wide components and short-term, localized components. Vehicle management approaches that address the long-term flow problem attempt to maintain steady-state vehicle flows at levels below the system guideway and station capacities. This problem was handled in the simulation by means of a network route planning and frequency allocation strategy. The approach consisted of partitioning the stations into sets, defining service routes that connect all pairs of station sets, and then allocating flows (number of vehicles per hour) to each route such that the combined steady-state flows of all routes do not exceed system capacities. The routes and frequency allocations were generated by a network flow model used as a simulation preprocessor (Ref. 5).

This approach was implemented in the simulation by checking the route assigned to each vehicle as it entered a station. If the station is first in the destination set of the vehicle's currently assigned route, a new route is selected as the vehicle's next assignment. The new route is chosen from the routes that originate at the current station and is the route for which a vehicle assignment is most behind schedule (as established by the route's allocated vehicle frequency of service) or least ahead of schedule.

The short-term, localized aspect of vehicle management results from the random nature of vehicle flows under vehicle-follower control. The goal of management strategies focusing on this short term problem is to prevent the occurrence of large strings of vehicles at merges or station entrances. Such strings can saturate the capacity of a merge and cause a vehicle backup or can overload a station and result in vehicle rejections. Vehicle debunching strategies and vehicle reservation schemes attempt to achieve more uniform spacing of vehicles or to meter the arrival rate of vehicles at merges or stations in order to avoid short-term saturation effects. A vehicle reservation scheme to meter vehicle arrivals during specified time windows at stations and merges was selected for this investigation.

2.1.3 Output and Performance Measures

The primary output of the network management simulation is a printed tabulation of various network parameters and performance measures at specified time intervals. Some of the performance measures available include: station rejections, service adherence distributions, vehicle headway distributions, vehicle velocity distributions, travel time distributions, and energy consumption statistics.

In addition, the simulation has provision for providing data to be used for a graphic display of vehicles in the network as well as post-processing for determining other performance measures.

2.1.4 User Options

The initialization phase of the simulation was developed to minimize the user's effort required to execute a run, while providing a high degree of flexibility with respect to system parameters and run control options. Default values are automatically assigned to most variables during initialization. The user can override the defaults by specifying values for those variables where changes are desired. Run control option variables, system-related parameters, and link and station characteristics can be input by the user via a run setup data file containing the appropriate job control language and data values. Service route data (route descriptions and frequencies), fleet dispatch data, point-to-point distance and travel time tables, and network path definition tables are initialized from data sets. These data sets are created by a modified version of the APL flow simulation (Ref. 5) again functioning as a preprocessor for the network simulation.

Run control options built into the simulation include output control options (file or printed), run time control, system modeling, and a save and restart capability. Output control option variables permit selection of an initial time and a sampling interval for vehicle-, link-, and station-related data. For debugging purposes, detailed vehicle and station status data can be printed by setting appropriate flags. Status data can also be written onto a disk file at selected intervals for post-processing. For example, vehicle position data has been retrieved from the file and input to a network graphics model to observe the flow dynamics of the network. This capability is a very useful debugging tool as well as an effective means of observing network performance.

The save and restart options provide the user with a capability to change one or more system parameters and restart from some point of a previous run after steady-state conditions had been established (i.e., when the vehicle fleet has been dispatched, distributed over the network, and assigned to service routes).

2.2 APPLICATIONS OF NETWORK MANAGEMENT SIMULATION TO VEHICLE-FOLLOWER STUDIES

2.2.1 Network Scenario

The network that was modeled is shown in Fig. 2.2. The network consisted of 20 stations and about 8 miles of unidirectional guideway. The simulation could handle as many as 200 vehicles; however, typical fleet sizes were about 100.

Demand levels for this network were assumed to be about 10,000 to 15,000 passengers per hour. A flow model analysis was used to define vehicle service routes and to estimate the required fleet size. The flow model also gave an estimate of the link flows, which indicated how near the system was to its capacity. The flow model produced fleet sizes of 72, 103, and 128 vehicles to simulate light, moderate, and heavily congested systems.

With the network management simulation, three major subject areas were investigated:

1. Network level interactions of the vehicle-follower control system,
2. Vehicle management interactions, and
3. Energy consumption.

2.2.2 Network Level Interactions of the Vehicle-Follower Control System

A fleet size of 72 vehicles with 33 routes was used to determine the initial performance of the network. A constant-headway spacing policy was assumed, with an operational headway of 5.0 sec. The following was observed after an hour of simulation of this sytem:

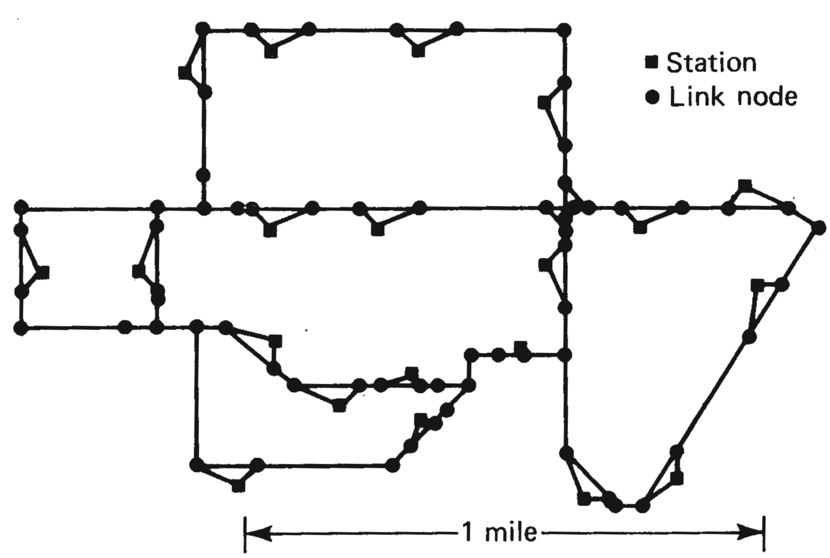

Fig. 2.2 **Network management simulation network configuration.**

1. No vehicle queuing instabilities were observed in the network as vehicle flows and service reached a steady-state condition.

2. No vehicles were denied access to their destination stations.

3. In steady state, 14% of the vehicles were in the vehicle-follower mode, 37% were in the velocity-control mode, and 49% were in station areas.

4. The maximum vehicle flow observed was 420 vehicles per hour (about 58% of maximum capacity).

5. All vehicles were able to arrive at their station stops within 30 sec of their scheduled arrival times.

The vehicle fleet size was next increased to 103 vehicles with 61 routes to determine a baseline performance of the network. The major results for one hour of simulation of this system were as follows:

1. No vehicle queuing or vehicle instabilities were observed in the network after the system reached steady state.

2. One station rejection was recorded out of 2639 station arrivals during the hour of simulation.

3. In steady state, 35% of the vehicles were in the vehicle-follower mode, 33% in the velocity-control mode, and 32% were in station areas.

4. The maximum observed link flow was 600 vehicles per hour (83% of the theoretical maximum).

5. No significant degradation in service was found; 71% of all vehicles arrived within 30 sec of their scheduled times, and 94% arrived within 90 sec.

The effects due to a congested network were examined by using a fleet size of 128 vehicles with 61 routes. The following observations were made:

1. Some vehicle bunching does occur, but no network instability in the vehicle string dynamics was found.

2. There were 43 rejections during the simulation, out of 3100 station arrivals. All of the rejections occurred in the last 45 min.

3. 38% of the vehicles were in the vehicle-follower mode, 25% were in the velocity-control mode, and 37% were in station areas.

4. The theoretical maximum link flow of 720 vehicles was observed during the simulation.

The previous runs assumed a constant-headway (constant-h) spacing policy. Another policy, constant-k (which generates spacing commands based on vehicle emergency deceleration rates) was implemented. The parameters of the constant-k policy were selected such that the spacing demanded at the maximum network speed would be equal to the spacing required by a constant-h policy. Simulation runs gave the following results:

1. When the system was not congested, system performance was almost the same under constant-k and under constant-h.

2. The constant-k policy showed significant improvement over constant-h in a congested network. 17% more vehicles arrived within 60 sec of their scheduled times under constant-k than under constant-h. Also, the constant-k system had only about half the rejections of the constant-h system.

A potential drawback to the constant-k spacing policy is that occasional violations of the spacing constraint imposed by a safety subsystem may be allowed. This aspect requires additional investigation and analysis.

2.2.3 Vehicle Management Interactions

The network management simulation was next used to assess (1) the interaction between station operations and network flows and (2) the potential benefits of using vehicle management schemes to improve system performance.

For this analysis, the network configurations in the previous runs were used with changes to station characteristics. The 103-vehicle fleet configuration, which exhibited good performance and throughput, was chosen as the baseline system. Originally, a station configuration of 4 entrance queue berths, 4 dock queue berths, and 2 exit queue berths (4/4/2) was assumed. The station configuration was changed to 2/2/1 for all but two stations, which by their nature were extremely busy and were therefore assigned a station configuration of 3/3/2. This reduction in station size stressed vehicles to follow their scheduled arrival times more closely or suffer a greater probability of station rejection.

The first run with the reduced station size system without any vehicle management scheme showed an increase in vehicle rejections from 1 to 17 in one hour of simulated operation. The majority of the rejections occurred at 2/2/1 stations located immediately downstream of merge junctions. Subsequent runs increased the number of entrance queues to 4, yet only reduced the number of rejections to 7.

A matrix reservation scheme was then used with the system. The matrix reservation logic attempted to meter the arrivals of vehicles during specified time windows at stations by delaying station departures, choosing routes, and dynamically altering routes once a vehicle is under way. There was great flexibility in selecting much of the matrix reservation logic; also, for the runs considered, the parameters were selected to provide the same level of service as the runs without the matrix reservation logic.

Using the matrix reservation logic with the 2/2/1 station configuration, station rejections were reduced from 17 to 10. When the entrance queues were increased by 2, the rejections decreased from 7 to 3. As mentioned before, the service level was about the same for all cases, with only a slight increase in the spread of origin/destination travel time distribution with the matrix reservation logic.

A final run was conducted in which the parameters of the matrix reservation logic were set to minimize station rejections. This run reduced rejections to 1, but was accompanied by a decrease in the general level of service (e.g., only 60% of all trips completed within 30 sec of nominal times versus 74% with 10 rejections).

These runs have shown (for one network configuration) that interaction between the mainline operations and the station configurations should be given careful consideration in the design of an AGRT system. A matrix reservation scheme can effectively reduce station rejections while maintaining the same level of system performance. There are trade-offs in reducing station rejections by either implementing a reservation scheme or by placing additional berths in the entrance queue of stations downstream of merge junctions.

2.2.4 Network Energy Consumption Characteristics

The detailed nature of the mainline operation of the network managment simulation allowed many vehicle characteristics to be measured. One such characteristic was energy consumption. To demonstrate the utility of the network management simulation in an energy analysis, a preliminary study of the vehicle energy consumption was done.

A simple model of the vehicle motor and drive train was used. In conjunction with the velocity commands generated by the vehicle controller, the model produced a required vehicle torque, which was then translated into power needed for the vehicle.

Some of the results of this study are presented below.

1. Energy consumption under a constant-h spacing policy was about 2.4% less than for a constant-k system.
2. There is a potential for recovering as much as 8% of the energy from vehicle operation on the mainline through the use of regenerative braking.
3. Power requirements for mainline maneuvering can be significantly different if the motor and motor controller have efficiencies that vary significantly with vehicle speed. The variation was as much as 23% in our model.

2.3 SIMULATION EFFICIENCY

The network management simulation was written in Fortran and executed on an IBM 3033 computer system. All network management simulation runs conducted for this study were for a minimum of one hour of simulated time. The ratio of the simulated time to central processing unit time required by the IBM 3033 ranged from about 70 to 140. The value of these ratios demonstrates the effectiveness of the discrete-event/time-step approach used in implementing the network management simulation so as to make it a potentially powerful tool in performing sensitivity studies of previously unaccessible network performance measures.

3.0 NETWORK MANAGEMENT SIMULATION

3.1 SIMULATION OVERVIEW

The objectives of this study require a simulation that models the dynamics of vehicle flow under vehicle-follower control while keeping computer resources (e.g., memory and central processing unit time) within reasonable limits to prevent excessive run costs. Furthermore, the simulation must be able to model the management of vehicles within a station (including berth assignment and queue-shifting logic) as well as to accurately assess energy consumption of the system.

The general simulation architecture selected to satisfy the study objectives is shown in Fig. 3.1. An event-scheduling approach characteristic of discrete-event simulations was used for primary control of the sequence of execution. Vehicle mainline operations, simulated using a continuous or time-step approach, were integrated into this event-based process by simply scheduling state update events at regular intervals that were equal to the selected time-step interval. As indicated in Fig. 3.1, the types of events that were scheduled included station-related events for vehicles in the discrete-event mode, the state-update event for vehicles in continuous mode, and state-related events to transfer vehicles from one mode to the other and to actuate logic for merge control and network-path selection.

The following aspects of the network management simulation will next be discussed in detail: the network definition, vehicle dynamics, network vehicle management, and simulation outputs.

3.1.1 Network Definition

The network management simulation was designed to model an AGRT network of moderate size. A network configuration was specified by means of six types of interconnecting links. The link types used by the network management simulation were nominal, merge, divert, station bypass, station entry, and station egress. Fig. 3.2 illustrates each link type in a generalized network. Up to 90 links can be input to the network management simulation in the definition of a network configuration. For each link, the following characteristics must be given: type, maximum speed, length, the identification of any companion merge links, the exit station identification (for station entry link), and the identification of the exit link(s).

Once the links have been specified and characterized, the stations can be defined. The network management simulation was sized for a maximum of 20 serial dock stations. The stations characteristics needed as input were the number of berths in each of the station queues and the identification of the link providing vehicle entry from the mainline.

3.1.2 Vehicle Dynamics

The network management simulation uses two approaches to handle individual vehicle dynamics—continuous and discrete. Vehicles on the main guideway or mainline function in a "continuous" manner; all other vehicles operate in a "discrete" mode.

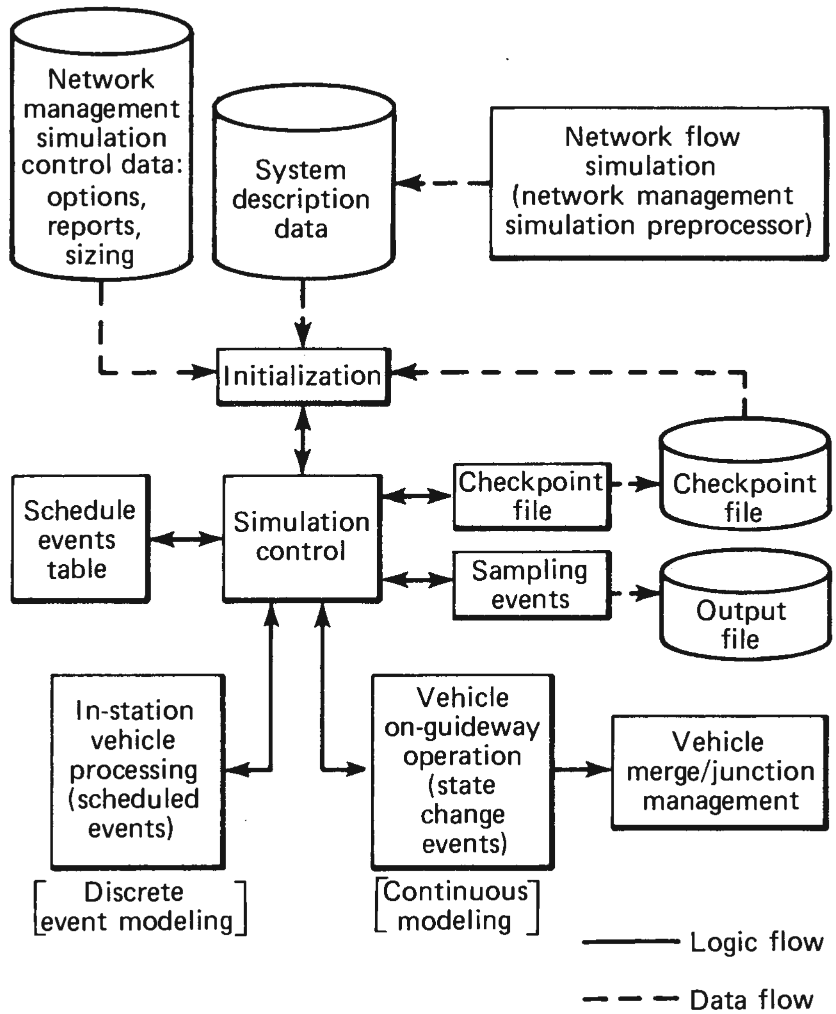

Fig. 3.1 Network management simulation architecture.

In the continuous approach, the vehicle velocity and position were updated by integrating those states based on acceleration commands of a regulation controller. The regulation controller used was derived by Pue (Ref. 3), and used a vehicle-follower, state-constrained approach. The controller generates an acceleration command based on the error between the actual spacing and a desired spacing based on kinematics and operational constraints. An ideal plant was assumed for the vehicle, so the acceleration commanded by the controller is essentially the vehicle acceleration. Figure 3.3 is a schematic of this controller design.

When vehicles are spaced far apart, vehicles reverted to an open-loop mode of operation, the velocity-command mode. Here the vehicles achieve and maintain the maximum link speeds until the vehicles either reach their destination station or overtake another vehicle, forcing them into the vehicle-follower mode again.

The computation of the acceleration and the state integrations were performed in events recursively scheduled in the network management simulation on a relatively short time scale

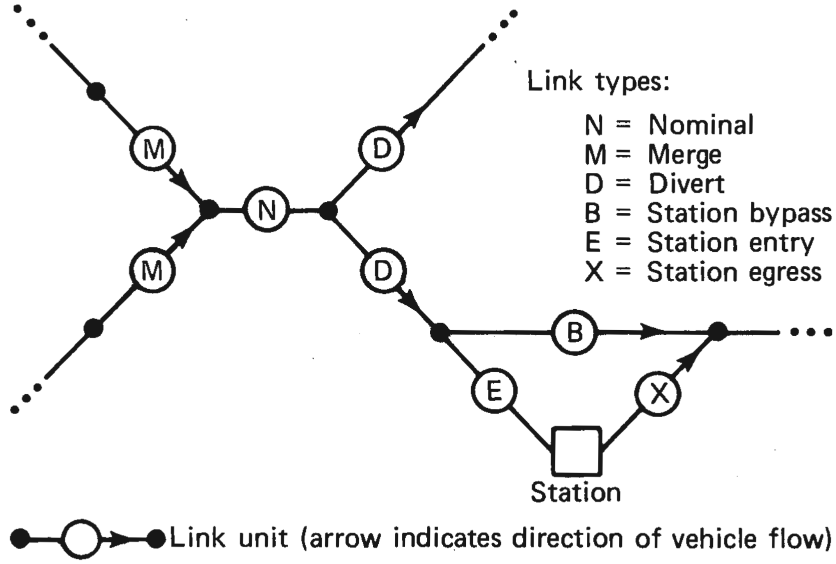

Fig. 3.2 Illustration of network management simulation.

(approximately 0.25 sec). The fidelity of the vehicle-follower dynamics was maintained in this way.

3.1.3 Vehicle Management

This simulation was designed to make individual vehicle movement almost autonomous. Once assigned a destination station, vehicles were introduced onto the main guideway. Vehicles then traveled through the network with at least a minimum spacing specified by the vehicle-follower controller. When vehicles reached a divert junction, a look-up table was used to direct vehicles to the divert branch appropriate for their destination station.

For any network configuration, as passenger demand increases, additional vehicles are required in order to serve the increased demand. With the increased fleet size, link and station flows approach the system capacity limits and a need arises to manage the level of vehicle flows to prevent saturation of system facilities.

The problem of effectively managing the vehicle flows in an AGRT network has both long-term, network-wide components and short-term, localized components. The objective of vehicle management approaches related to the long-term flow problem is to maintain steady-state vehicle flows at levels below the system guideway and station capacities. This problem was handled in the simulation by means of a network route planning and frequency allocation strategy. The approach consisted of partitioning the stations into sets, defining service routes that connect all pairs of station sets, and then allocating flows (i.e., the number of vehicles per hour) to each route such that the combined steady-state flows of all routes do not exceed system capacities. The routes and frequency allocations were generated by a network flow model used as a simulation preprocessor (Ref. 5).

This approach was implemented in the simulation by checking the route of each vehicle as it entered a station. If the station is first in the destination set of the vehicle's currently

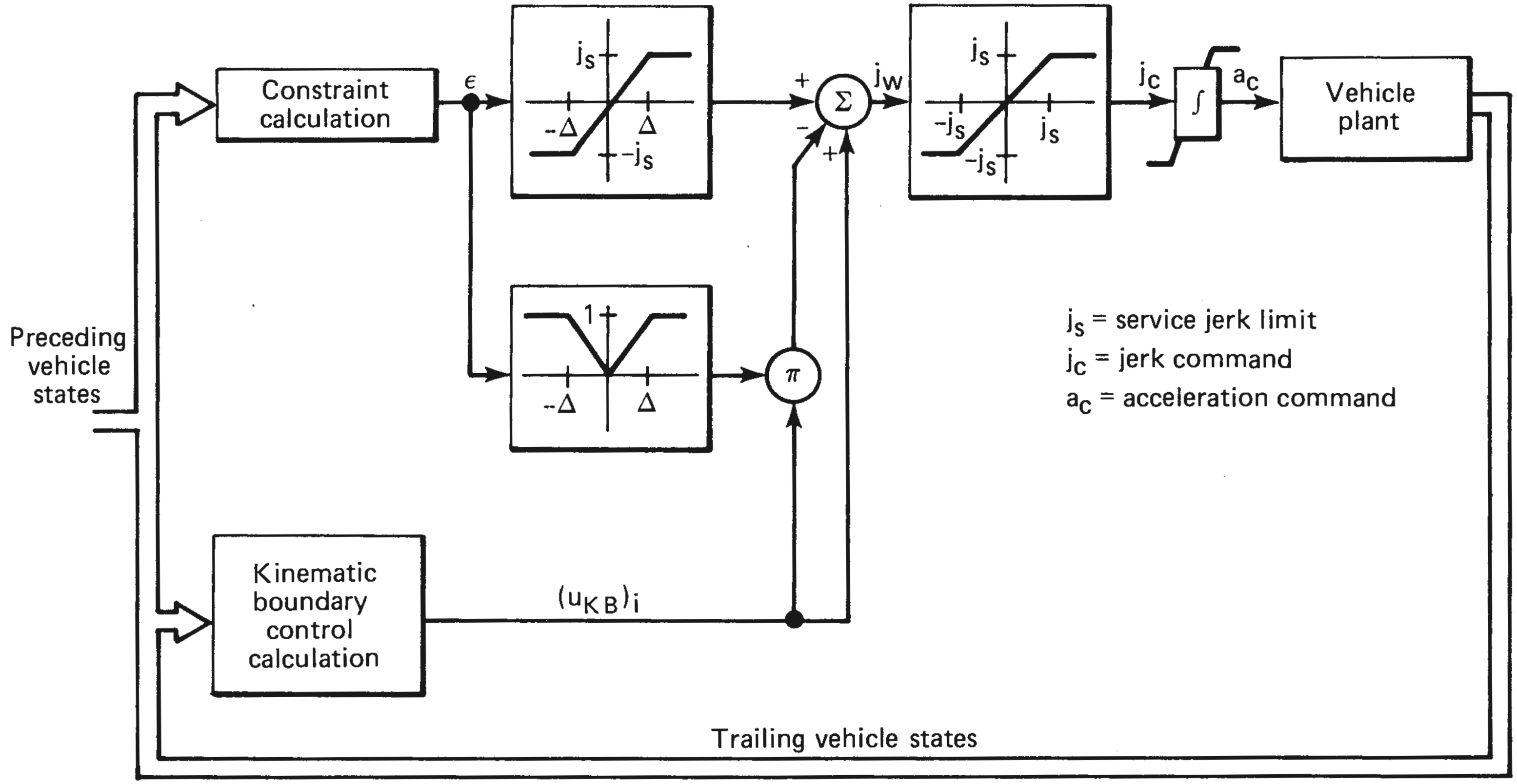

Fig. 3.3 Vehicle-follower controller structure.

assigned route, a new route is selected as the vehicle's next assignment. The new route is chosen from among all routes that originate at the current station and is the route for which a vehicle assignment is most behind schedule (as established by the route's allocated vehicle frequency of service) or least ahead of schedule. When a vehicle is dispatched from the station, that vehicle's new destination is set to the next station along the assigned service route and the vehicle is sent into the mainline as soon as it is ready to depart. This is possible with a vehicle-follower control scheme because the vehicle-follower controller automatically generates gaps in the mainline as necessary.

The short-term, localized aspect of vehicle management results from the random nature of vehicle flows under vehicle-follower control. The goal of management strategies focusing on this short-term problem is to prevent the occurrence of large strings of vehicles at merges or station entrances. Such strings can saturate the capacity of a merge and cause a vehicle backup or can overload a station and result in vehicle rejections. Vehicle debunching strategies and vehicle reservation schemes attempt to achieve more uniform spacing of vehicles or to meter the arrival rate of vehicles at merges or stations in order to avoid short-term saturation effects. A vehicle reservation scheme to meter vehicle arrivals during specified time windows at stations and merges was selected for this investigation. The basic features of the logic incorporated in the scheme are listed below:

- The reservations matrix IVRSV(I,J) contains the number of reservations that have been booked for vehicles arriving at stations (and merge junctions, if desired), I, during future time windows, J. The width of the time window (WSIZE) is specified by the user and should be on the order of the station platform cycle time, with a lower bound set by the uncertainty inherent in travel times with vehicle-follower control. A value of 30 sec is suggested as an initial window size with the dimension of J set at 40 to handle trips up to 20 min in length.

- A pointer IRP is used to store the column representing the time window containing the current time and is advanced by routine RSVUPD every WSIZE seconds. The matrix is circular with respect to time in that the window or column representing a future time is determined by starting at column IRP and advancing to the last column (40 in this case) and then looping back to column 1.

- For each station, I, the maximum number of reservations allowed in each time slot, IVTRHD(I), is computed by the following formula:

$$\text{IVTRHD(I)} = \text{THDF*(ISTA(I,2)*(WSIZE/CYCLT))} ,$$

where THDF = threshold factor between 0 and 1,
ISTA(I,2) = number of station berths,
WSIZE = window width (sec), and
CYCLT = station cycle time (dwell and access time, sec).

- Primary paths and alternate paths (if available) from all stations and divert functions to all stations and merge functions are identified by the flow model preprocessor. The nominal travel times for these paths are determined and stored in arrays PTT and ATT. If an alternate path does not exist, the times stored in each array are equal. Also, if an alternate path originating at a divert function diverges from the primary path at a downstream divert rather than at the origin junction, the times stored are equal.

• A subroutine is called prior to each vehicle dispatch in order to reserve an arrival slot at the vehicle's next destination. If the arrival slot using the primary path time is filled, a reservation based on the alternate path time is requested. If both attempts are unsuccessful, a dispatch attempt is rescheduled in 5 sec unless the station is congested (entrance queue filled and no vehicles undergoing a dwell), in which case the vehicle is assigned to the alternate path and the slot is overbooked. An option is available such that if a slot using the alternate path is available and if the station is not congested, the additional travel time incurred by using the alternate path is compared with the delay of waiting for a primary path slot. If the delay is less, no reservation is made and a dispatch attempt is rescheduled.

3.1.4 Output and Performance Measures

The primary output of the network management simulation is a printed output of various network parameters and performance measures at specified time intervals. The exact output in the listing will be discussed later. Some of the performance measures were station rejection, service adherence distributions, headway distributions, velocity distributions, travel time distributions, and energy consumption statistics.

In addition, the simulation has provision for providing data to be used for a graphic display of vehicles in the network as well as post-processing for determining other performance measures.

3.2 NETWORK MANAGEMENT SIMULATION PROGRAM DESCRIPTION

3.2.1 Network Management Simulation MAIN Program

The network management simulation is a Fortran simulation consisting of a main program MAIN and 36 subroutines. The source code list of the network management system is given in Appendix A, and the definitions of the simulation variables in Appendix D. The program MAIN can be divided into three sections: initialization, event processing, and termination. A schematic of the simulation logic flow is shown in Fig. 3.4.

The initialization phase of the simulation was developed to minimize the effort required by a user to execute a run while providing a high degree of flexibility with respect to system parameters and run control options. Default values are assigned to most variables automatically during initialization. The user can override the defaults by specifying values for those variables where changes are desired. Run control option variables, system-related parameters, and link and station characteristics can be input by the user via a run setup data file containing the appropriate job control language and data values. Service route data (route descriptions and frequencies), fleet dispatch data, point-to-point distance and travel time tables, and network path definition tables are initialized from data sets. These data sets are created by a modified version of the APL flow simulation (Ref. 5) such that it functioned as a preprocessor for the network simulation.

Run control options built into the simulation include output control options (file or printed), run time control, system modeling, and a save and restart capability. Output control option variables permit selection of an initial time and a sampling interval for vehicle-, link-,

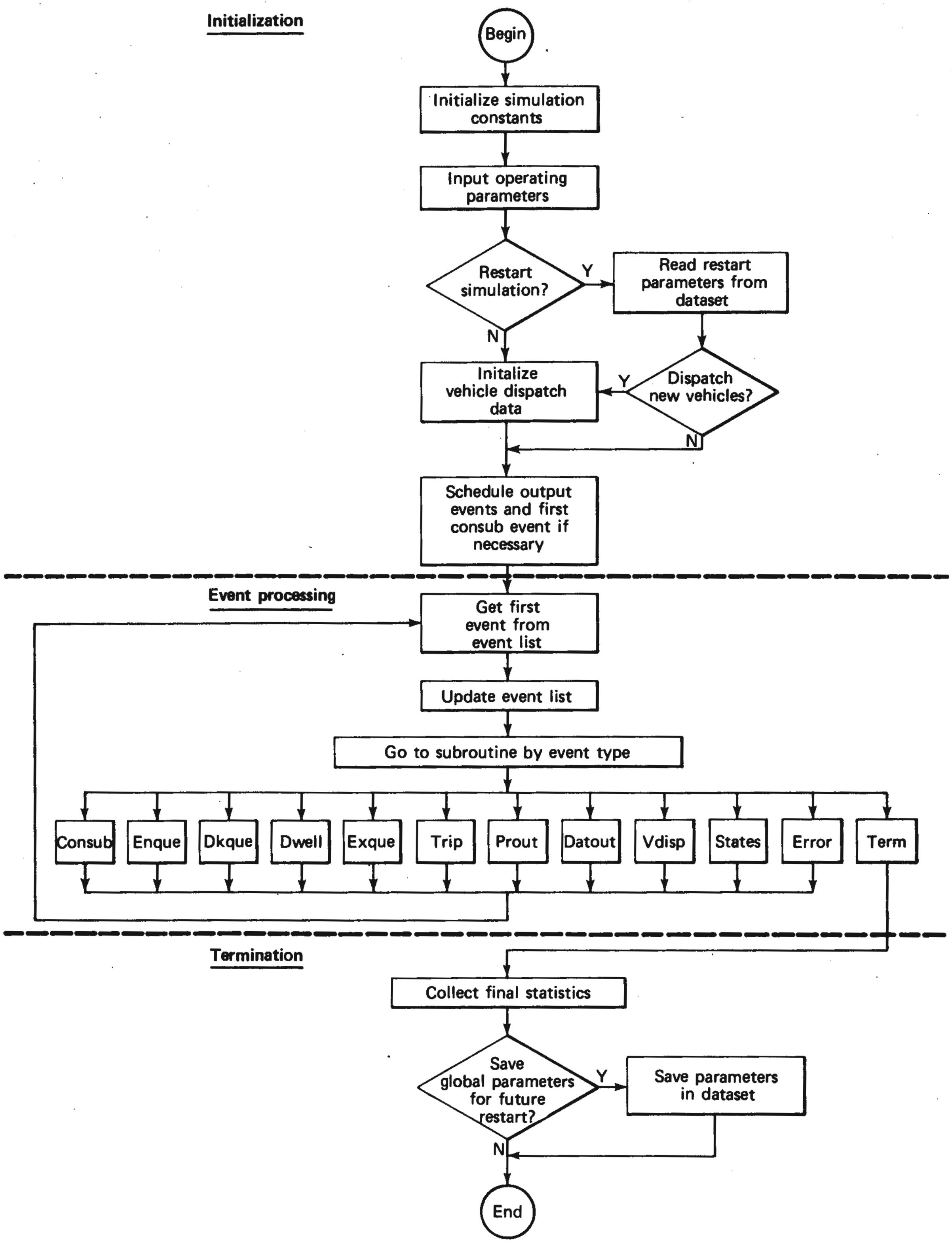

Fig. 3.4 Network management simulation logic schematic.

and station-related data. For debugging purposes, detailed vehicle and station status data can be printed by setting appropriate flags. Status data can also be written onto a disk file at selected intervals for post-processing. For example, vehicle position data has been retrieved from the file and input to a network graphics model to observe the flow dynamics of the network. This capability is a very useful debugging tool as well as an effective means of observing network performance.

The save and restart options provide the user with a capability to change one or more system parameters and restart from some point of a previous run after steady-state conditions had been established (i.e. vehicle fleet has been dispatched, distributed over the network, and assigned to service routes). The user can increase the fleet size at the beginning of a restart in order to examine the impacts of increased vehicle link and station flows.

The size of the AGRT system that can be simulated is limited by the array dimensions specified at the time of compilation. For this study these limits were set as follows: maximum vehicle fleet, 200; maximum number of berths per station, 10; maximum number of stations, 20; maximum number of links, 90; and maximum number of events in the event calendar, 500.

Last in the initialization process, events are scheduled that are needed to begin the recursive generation of events. The event terminating the simulation is also scheduled here.

The heart of the simulation is the event-processing section. As they are generated, the initialization events are put into a list and ordered according to their time of execution. Next, the event-processing section takes the event at the top of the list, updates the simulation time to the event time, executes the appropriate subroutine corresponding to the event, and sorts the event list again to begin the process over again. In many of the event subroutines, new events are generated that are entered into the event list. The events that are in the network management simulation are CONSUB, DKQUE, DWELL, EXQUE, VDISP, PROUT, STATES, DATAOUT, ERROR, and TERM. The next section will give a brief description of each of these routines.

When the TERM event is finally called, the termination portion of the network management simulation is executed. Final statistics are collected from the simulation. If a "save" option has been evoked, the parameters necessary for a "restart" are saved to a data set.

3.2.2 Event Descriptions

There are 10 events that can be called and executed from MAIN. The event VDISP is concerned with establishing the network vehicle flows. CONSUB handles the dynamics of mainline vehicles. DKQUE, DWELL, and EXQUE are events controlling the movement of station vehicles. Simulation data collection and output are done by PROUT, STATES, DATOUT, and ERROR. The termination of the network management simulation is performed by the event TERM. A more detailed description of each of these events follows.

3.2.2.1 *VDISP*

The VDISP event is used to initially dispatch vehicles into the network. When this event is called, a vehicle is given a route assignment and an initial destination station. The vehicle is then placed in the exit queue of one of several dispatch stations and allowed to merge into the mainline. Finally, if with the dispatch of this vehicle the maximum fleet size has not been reached, VDISP is rescheduled. A logic flow schematic of VDISP is shown in Fig. 3.5.

3.2.2.2 *CONSUB*

The CONSUB event updates the states of all mainline vehicles. CONSUB determines whether a mainline vehicle should be in a vehicle-follower or velocity-command mode and then generates the appropriate acceleration command. CONSUB also checks whether or not a mainline vehicle has reached its destination link and if so, changes the vehicle status from mainline to station where station events can then control its movements. Once a vehicle has left the main guideway and entered a station, the vehicle is then assigned a new route. CONSUB reschedules itself at its completion. An option in CONSUB allows energy statistics to be computed and collected on mainline vehicles. Details of the energy models can be found in Appendix B. A logic schematic of CONSUB is shown in Fig. 3.6.

3.2.2.3 *DKQUE*

The purpose of DKQUE is to handle vehicle movement into and out of the station dock queue. An input to DKQUE is the identification number of a vehicle. DKQUE will shift the input vehicle from the entrance queue to the assigned lock queue berth if the path to the dock queue is clear. If not, the vehicle is then shifted to the available berth most downstream. Once the vehicle has reached its assigned dock berth, DKQUE initiates the dwell period by scheduling the DWELL event. DKQUE then shifts a vehicle completing its dock dwell to the exit queue by scheduling an EXQUE event.

Vehicle states are not integrated in DKQUE. Instead, vehicle shift times are computed as a function of the distance traveled. These times are then used to update vehicle position in the station. The equations used to calculate vehicle shift times are based on shift time data obtained from runs of a detailed precision stop controller simulation (Ref. 6). Figure 3.7 is the logic schematic of DKQUE.

3.2.2.4 *DWELL*

DWELL is an event that signals the end of a dock dwell of an input vehicle. DKQUE is then scheduled to begin the shift of the vehicle into the exit queue. Figure 3.8 is the logic schematic of DWELL.

3.2.2.5 *EXQUE*

The function of the event EXQUE is to reinsert station vehicles waiting in the exit queue back onto the main guideway. When an input vehicle has reached the head of the exit queue, a lead vehicle is found on the main guideway. At this point, the input vehicle is now considered a mainline vehicle whose position and velocity is controlled through the CONSUB event. If the vehicle management option has been evoked, the release of the input vehicle to the mainline will be delayed until a dispatching algorithm determines a suitable time. Once the input vehicle has been dispatched, remaining vehicles in the station are shifted forward. Figure 3.9 is the logic schematic of EXQUE.

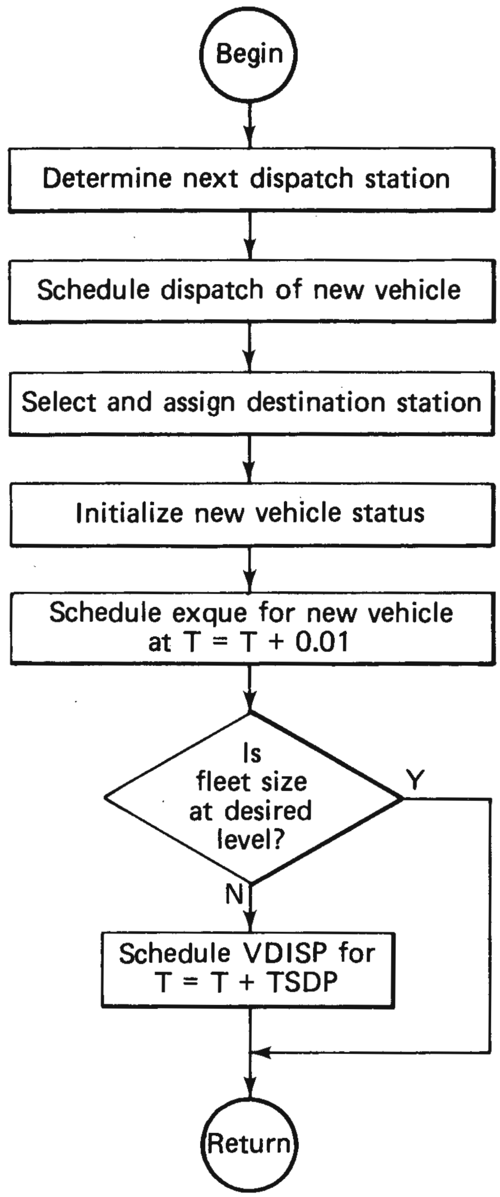

Fig. 3.5 Logic schematic for VDISP.

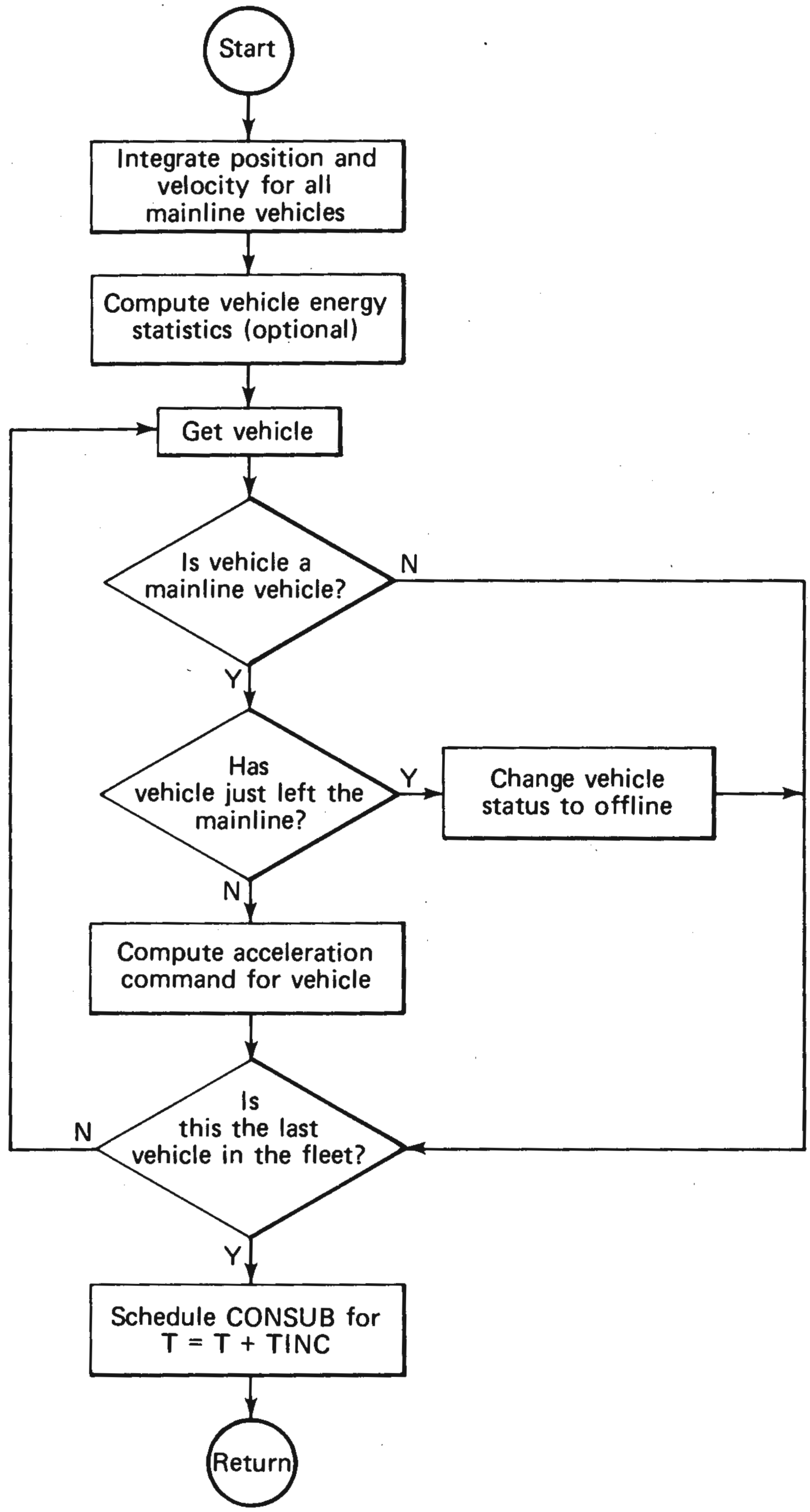

Fig. 3.6 Logic schematic for CONSUB.

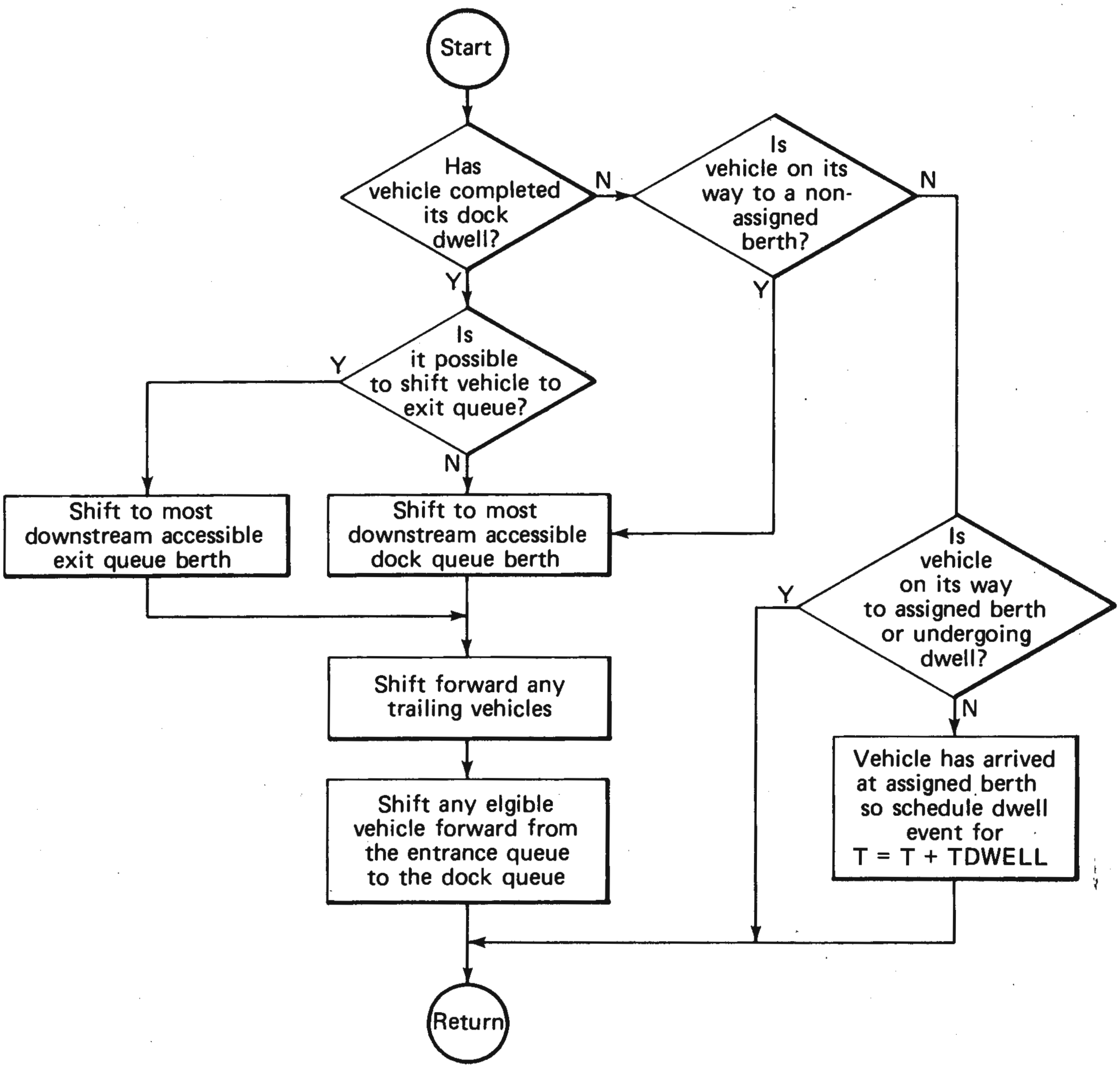

Fig. 3.7 Logic schematic for DKQUE.

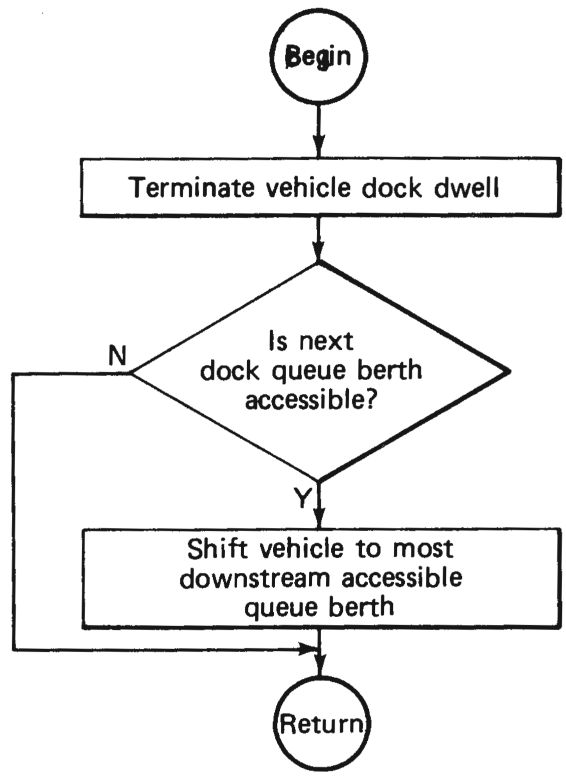

Fig. 3.8 Logic schematic for DWELL.

3.2.2.6 *PROUT*

The PROUT event produces hard copy output of various simulation statistics and performance measures. Table 3.1 lists the various PROUT outputs with their associated definitions and units. A sample of a PROUT output is shown in Appendix C. PROUT is recursively scheduled at user-defined intervals.

3.2.2.7 *DATOUT*

The purpose of DATOUT is to produce data sets that can be post-processed for evaluating network management simulation performance. The data sets are grouped by link, vehicle, and station data. Each data set is updated independently. DATOUT is recursively scheduled at user-defined intervals for each type of data set.

3.2.2.8 *STATES*

The STATES event is primarily a debugging and verification tool in which the state of every vehicle and other pertinent information are printed out at specified time intervals. It provides information to check out the vehicle dynamics on the most elementary level. Figure 3.10 is a general list of the printed vehicle variables. STATES is recursively scheduled at user-defined intervals.

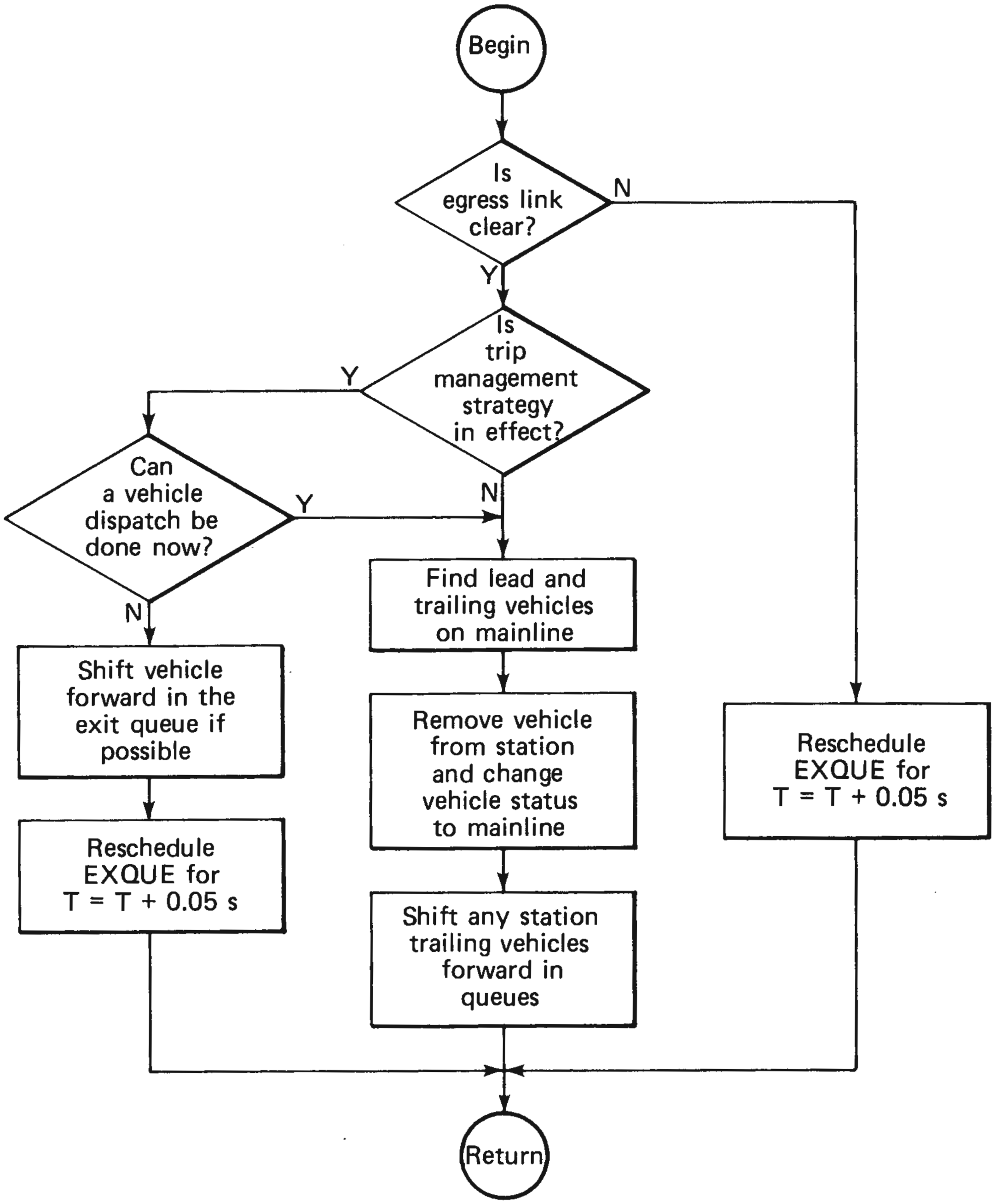

Fig. 3.9 Logic schematic for EXQUE.

Table 3.1

PROUT Outputs.

Output Name	Unit	Definition
Time	sec	The simulation time at which PROUT statistics are generated
Total vehicles	veh	The number of vehicles currently in the network
Total vehicles:		The number of vehicles currently in:
in vehicle-follower mode	veh, %	vehicle-follower mode
in velocity-command mode	veh, %	velocity-command mode
in station	veh, %	station berths
Vehicle distance	km, mi	Vehicle-miles traveled during the PROUT time interval
Total vehicle distance	km, mi	Total vehicle-miles traveled during the simulation
Average speed	mph	Average vehicle speed during PROUT time interval, including station stop delays
Station rejections	veh	Total number of vehicles denied station entry during the entire simulation run
Rejections by station	veh	Histogram of station rejections by station number
Station queue status	veh ID	Occupation of station queue berths by vehicle ID (a negative vehicle ID indicates a reserved but unoccupied berth)
Link number	veh	Number of vehicles in link
Link flow	veh/int	Number of vehicles passing through link during a PROUT interval
Link density	veh/m	Current link vehicles per link
Link speed	m/sec	Average vehicle speed on link
Link energy	kWh	Energy used on link with regenerative braking during PROUT interval
Link energy-D	kWh	Energy used on link without regenerative braking during PROUT interval
Total network energy	kWh	Total energy consumed during simulation with regenerative braking
Total direct network energy	kWh	Total energy consumed during simulation without regenerative braking
Service route schedule adherence	No. veh	Number of vehicles, by route, arriving and departing within 30 sec intervals of predefined service route schedules
Distribution of headways, by link	No. veh	Distribution of vehicle headways for each network link (5 sec bin size)
Station time data	No. veh	Distribution of total vehicle stop time, by station (30 sec bin size)
O/D trip time data	No. veh	Distribution of station-to-station trip time, for all station pairs (60 sec bin size)
Vehicle energy consumption	kWh	Energy consumed, by vehicle

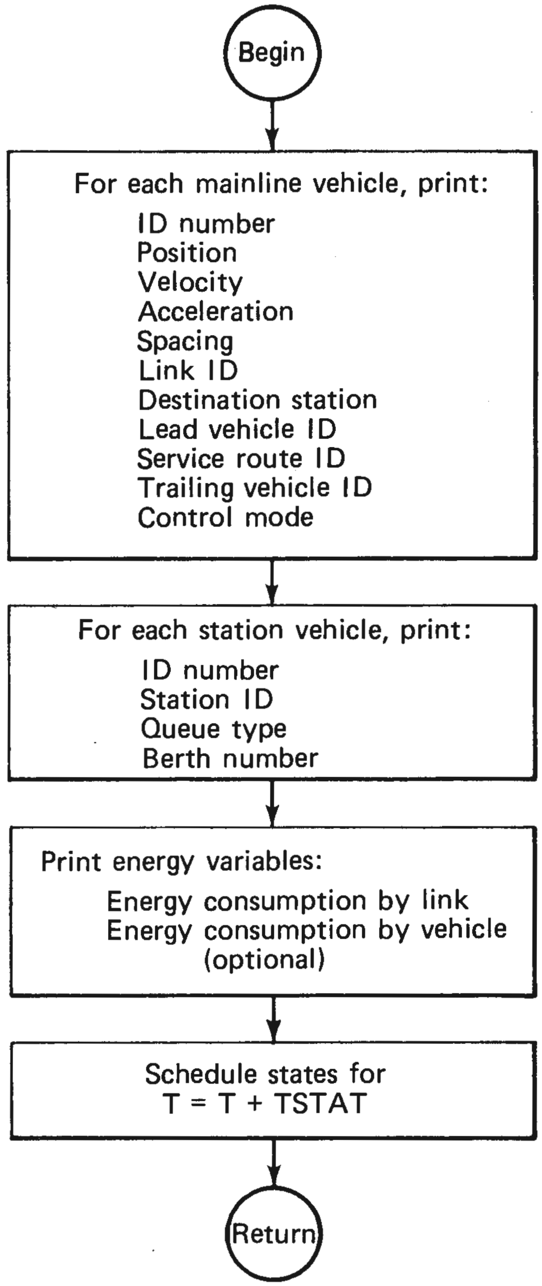

Fig. 3.10 Logic schematic for STATES.

3.2.2.9 *ERROR*

ERROR is an event used to print out diagnostic messages when fatal errors occur during a simulation run. The error message is listed and the TERM is executed, which then terminates the simulation.

3.2.2.10 *TERM*

The TERM event is called to end the running of the network management simulation. TERM generates a summary sheet of the run and then executes a global save of the simulation parameters if the SAVE option has been chosen.

3.2.3 Description of Vehicle Management Routines

A set of seven subroutines is associated with vehicle management functions. Routine RTASGN controls the assignment of vehicles to service routes. Routines DSPRSV and DVRTRV determine the path assignments and make reservations for vehicles departing stations and approaching divert functions. The remaining four routines (RSVUPD, COLUMN, ADDRSV, and CNCLRV) are utilities for updating the reservation matrix and inserting or canceling reservations. A brief description of the function of each routine is given below:

1. RTASGN(T,IDV,IDSTAT,IDRT) determines the next route assignment for a vehicle approaching the first station in the destination set of its current route. The selected route is the one most behind schedule or, if none, the route that would be least ahead of schedule.

2. DSPRSV(IDOS,IDV,T,IDFLG) reserves a slot in the vehicle reservation matrix prior to station dispatch. If the desired slot is filled, an attempt is made to reserve a slot using an alternate network path. If this slot is also filled, an EXQUE event is rescheduled for this vehicle (at T + 5.0) or, if the station is congested, the alternate path slot is overlooked and the vehicle is dispatched.

3. DVRTRV(LNK,IDV,IDS,T) reconfirms the current reservation of a vehicle as it approaches a divert junction. If it is invalid because of prior overbooking or vehicle enroute delays, a new reservation is made and the old reservation is cancelled.

4. RSVUPD(T) resets the reservation pointer, IRP, to the next time column and clears the current column. RSVUPD reschedules itself every reservation matrix time interval.

5. COLUMN(IRP,RDT,WSIZE,ICOL) converts a continuous future time into the appropriate column number of the reservations matrix.

6. ADDRSV(IRW,ICL,ISTAT) is called to place a vehicle reservation in the appropriate row (IRW) and column (ICL) of the reservation matrix. ADDRSV returns a value of 1 in INSTAT if the requested reservation is accepted; otherwise it returns a value of 0.

7. CNCLRV(IRW,ICL,T) cancels a reservation at location IRW,ICL in the reservation matrix.

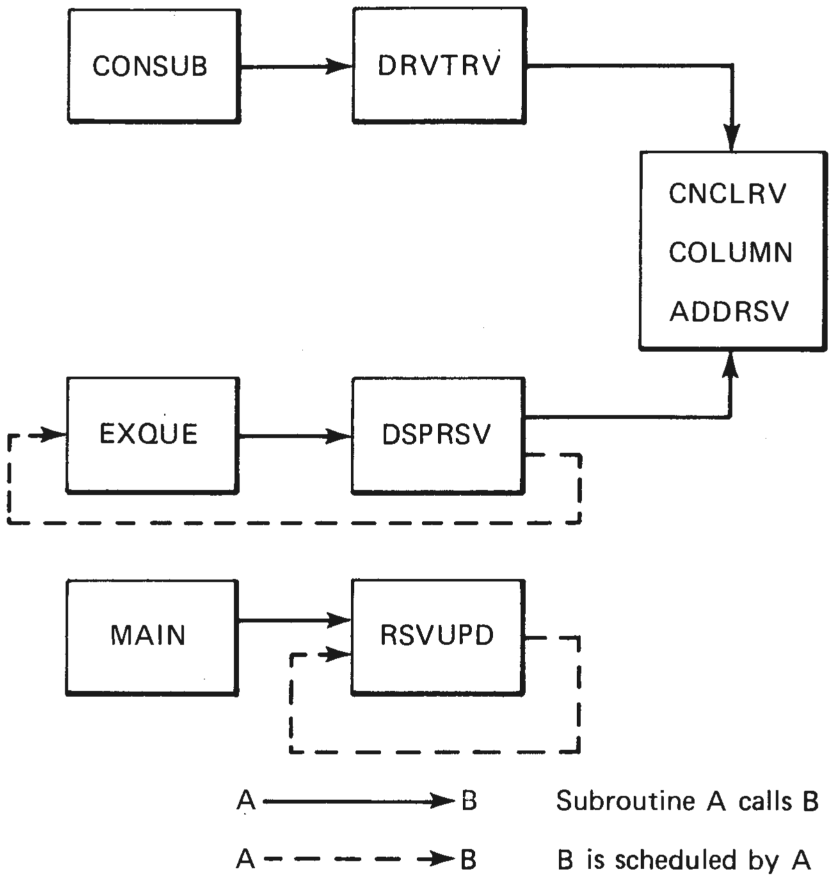

Fig. 3.11 Reservation matrix subroutine interactions.

The calling sequences among these routines are diagramed in Fig. 3.11. Routine DSPRSV is called from EXQUE while routine DVRTRV is called from CONSUB for each vehicle approaching a link crossover at a divert junction. Routines COLUMN, ADDRSV, and CNCLRV are called by DSPRSV and DVRTRV for booking and canceling reservations. RSVUPD is scheduled initially from MAIN and thereafter reschedules itself each reservation matrix time interval. RTASGN is called by REMOVE for each vehicle prior to entering a station.

3.3 SIMULATION VALIDATION

Validation of the simulation was approached in three ways. Logic to detect erroneous conditions was built into several routines. When such conditions are detected, error flags are set that, in turn, cause appropriate messages to be printed. Reasonableness checks of state data were made by means of hand computations and examination of graphical representations of vehicle flows. The final method of validation was to compare results of the simulation with results of merge or station simulations (Ref. 3) as well as steady-state flow and performance estimates of the APL network flow model (Ref. 5).

The validation process focused initially on those components of the simulation concerned with the movement of vehicles over the guideway (e.g., vehicle control, vehicle routing, and simulation control routines). These routines were exercised using only the central loop of the network configuration at first, and then on the complete network. The process continued with the addition of vehicle dispatch logic, a full complement of service routes and assignment logic, detailed station models, vehicle energy model, and finally a reservation matrix approach for traffic management. At each step in the process, results were verified at a local or detailed level by reasonableness checks (hand computations, previous simulation studies) and at the network level by comparison of such measures as link flow, station-to-station travel times, and route schedule adherence statistics with values projected by the network flow model for the same set of service routes and fleet size.

4.0 STUDY RESULTS

4.1 NETWORK SCENARIO

The network scenario selected for the study was Network Model C defined in Ref. 7. The network configuration (Fig. 4.1) consisted of approximately 8 lane-miles of unidirectional guideway and contains 20 off-line stations. The network is representative of an activity center circulation system that might be typical of initial AGRT deployments. This configuration was selected because it had most of the characteristics and complexities of larger networks (e.g., alternate paths, full intersection, and multiple service routes) but was small enough to be simulated at a reasonable cost.

The specific configuration coded as a baseline for the study included 90 links with line speeds ranging from 7 to 15 m/sec. The links immediately upstream of merge junctions represented parallel data regions within which vehicles maneuver in order to resolve merge conflicts. These links were sized according to the guidelines developed in Ref. 3 for merging with vehicle-follower control. All stations are assumed to be single channel, serial berth configurations with an entrance queue, dock queue, and exit queue. Entrance and exit ramps were sized to allow offline vehicle acceleration and deceleration to and from the mainline speed.

The initial station configuation was based on a network analysis of the Model C scenario using the analysis flow model of Ref. 7. A 4/4/2 station configuration was selected (i.e., 4 entrance queue positions, 4 berth docks, and 2 exit queue positions) on the basis of the steady-state station flows computed by the analysis model for system demand levels of 10,000 to 15,000 passengers per hour, and the theoretical station capacities estimated in Ref. 6.

4.2 NETWORK ANALYSIS PROCEDURE

The analysis procedure consisted of three sets of simulation runs, each addressing one of the principal objectives of the study. The first set focused on examining the network behavior of vehicle-follower control and establishing performance as a function of link flow as line saturation is approached. For these runs, stations were configured with excess capacity so that saturation of the network would occur on the mainline first and the practical capacity of vehicle-follower control could therefore be identified. These runs included an examination of both constant-h and constant-k implementations of vehicle-follower control.

A second set of runs focused on station operations and interactions with network operations. Station configurations were varied in order to match capacity to flow as closely as possible without causing a significant number of rejections. The benefits of a reservation matrix scheme for metering vehicle arrivals at stations was also evaluated as a means of reducing rejections. An important objective of these runs was to observe whether or not the operating capacity of various station configurations is significantly less than theoretical capacity estimates (such as derived in Ref.6, where berth assignments are made when the vehicle arrives at the station divert ramp in order to permit sufficient time for passengers to be informed of and walk to the appropriate berth for boarding).

The final set of runs examined the energy consumption and power demand characteristics of vehicle-follower control. Energy consumption under various degrees of line congestion was observed. The sensitivity of energy consumption to motor operating efficiency and

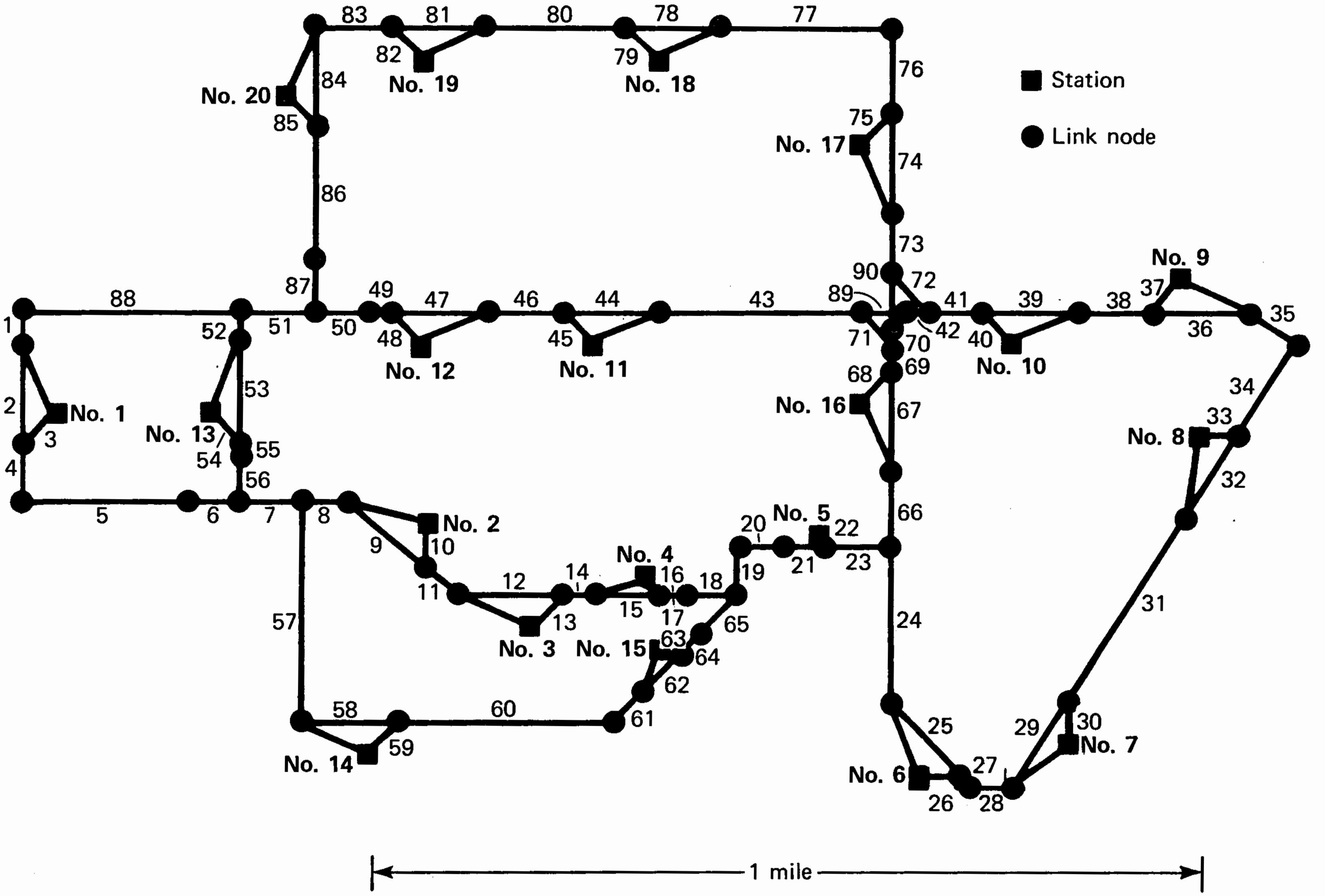

Fig. 4.1 Detailed network management simulation network configuration.

regenerative braking was examined. The objective of these runs was to study relative impacts or sensitivities as opposed to estimating the energy requirements of AGRT.

4.3 SIMULATION RESULTS

4.3.1 Network Behavior Under Vehicle-Follower Control

Under the assumed demand and network configuration presented in the previous section, the flow model preprocessor gave a required vehicle fleet size of 72 vehicles operating with 33 routes. A simulation run with this vehicle fleet size and route structure was made to determine a baseline performance measure for the network. In addition, a constant-headway vehicle-follower spacing policy was assumed where the nominal operating headway was 5.0 sec. The following results were observed from a one-hour simulation of the operation of this system:

1. Steady-state link flows were achieved within 15 min after the initial start-up time. The average vehicle speed (which included station stops) was 6.36 m/sec during the steady-state period (the last 45 min).

2. Operational steady-state flow (i.e., where vehicles achieved a constant level of service with respect to trip times) required about 30 min.

3. Averaged over the last 30 min of the simulation, 14.2% of all vehicles were in the vehicle-follower mode, 37.2% were in velocity-command mode, and 48.6% were in the station areas.

4. Examination of the headway distribution and link densities revealed no strong tendency for vehicles to form large vehicle strings after operational steady state had been reached.

5. There were no vehicles rejected for station entry out of 2456 station arrivals during the simulation hour. Snapshot examination of the network showed that entrance queues for all stations rarely had more than one vehicle at a given time.

6. The maximum observed vehicle flow was 420 vehicles per hour on the mainline.

7. During the last 30 minutes, the 20 stations processed vehicles at the rate of 2714 vehicles per hour. The maximum observed flow through the stations was 187 vehicles per hour, which is 58% of the theoretical maximum, assuming a 15 sec station dwell and an average of 47 vehicles per hour per berth (a typical value observed in Ref. 6). Furthermore, 97% of the vehicles required between 30 to 60 sec to exit from the main guideway, move through the station queues, and then begin re-entry onto the main guideway on the egress ramp. The rest of the vehicles needed 60 to 90 sec.

8. All vehicles were able to maintain their schedule of assigned station stops with a maximum delay of less than 30 sec.

The goal of the next simulation run was to obtain statistics in operating the network as close to its maximum capacity without incurring an unacceptable rate of station entry rejections. To achieve this, the original demand was increased to require a 103 vehicle fleet size by the flow model preprocessor. The number of routes then also increased from 33 to 61. The following observations were made from a simulation hour of this system:

1. Steady-state link flows were achieved by 20 min into the simulation. Operational steady-state was achieved by 30 min. The average vehicle speed for the last half hour was 7.0 m/sec.

2. An average network snapshot during the last 30 min would show that 35.3% of the vehicles were in the vehicle-follower mode, 32.7% were in the velocity-command mode, and 32.0% were in station areas.

3. The station entry rejection rate was very low. Only one station entry rejection (at station No. 2) was recorded out of 2639 arrivals during the one-hour simulation.

4. Headway and link density distributions do not suggest any significant vehicle bunching due to vehicle-follower control.

5. The maximum link flow was 600 vehicles per hour, 83% of the theoretical maximum (assuming an operational headway of 5 sec).

6. The average vehicle throughput in stations was 204 vehicles per hour. 95% of vehicle in-station time was between 30 to 60 sec. The remaining 5% was between 60 to 90 sec.

7. No significant degradation in service was observed in this run. 77% of all vehicles were within ±30 sec of their scheduled route times. 94% were within 60 sec. The service adherence distribution is plotted in Fig. 4.2.

In this second simulation run, we came very close to operating at the system's theoretical capacity without exhibiting any significant degradation in service. The results of this run were a good match to the maximum link flows, the maximum station flows, and vehicle-miles traveled per hour predicted by the flow model preprocessor. Table 4.1 shows the simulation values and the flow model values; agreement is within 10%.

The objective of the next simulation run was to examine the degradative behavior of a congested network. Demand was increased to raise the required fleet size from 103 vehicles to 128 vehicles operating under 61 routes (as determined by the flow model preprocessor). The simulation was restarted from the 103 vehicle fleet simulation and allowed to run for one hour. Vehicles were added until 128 vehicles were in the network. Also, the routes were continually readjusted until the new 61 routes were established.

The following was observed about this simulation run:

1. Vehicle flow appear to achieve steady state after 20 min. The average vehicle speed was 5.74 m/sec. Operational steady state also seem to occur after 20 min.

2. In steady state, the percentage of vehicles in the station areas rose compared to the 103 vehicle fleet run, from 32.0% to 37.7%. The percentage of vehicles in vehicle-follower mode also increased from 35.3% to 37.5%. Vehicles under velocity command dropped from 32.7% to 24.8%.

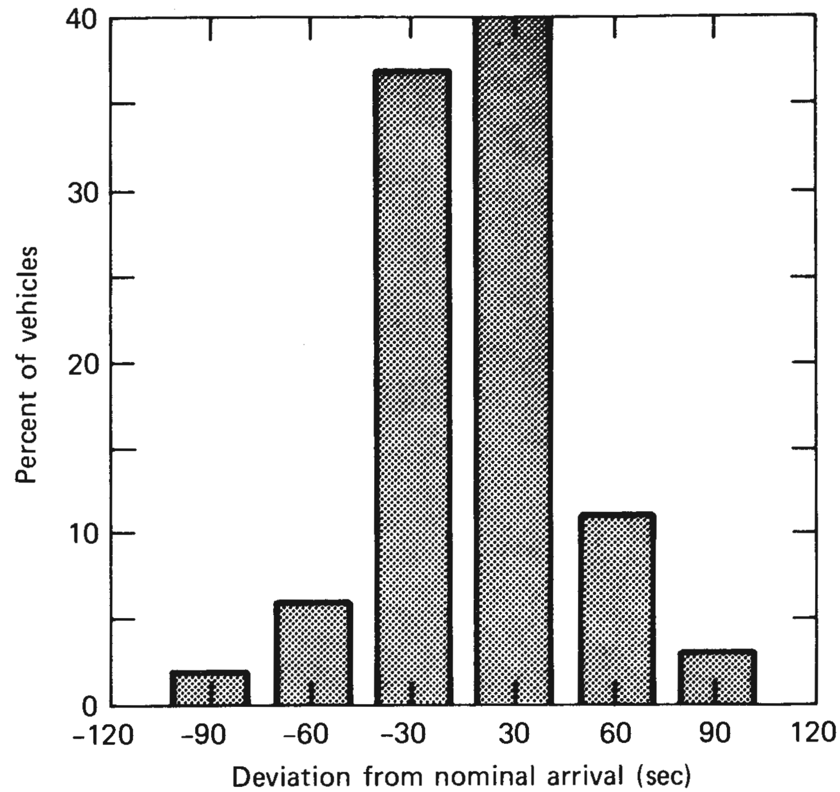

Fig. 4.2 Service adherence, 103 vehicles.

Table 4.1

Flow model versus network management simulation results, 103 vehicle fleet.

	Maximum Link Flow (vehicles/hr)	Maximum Station Flow (vehicles/hr)	Vehicle-Miles Traveled (miles)	Service Adherence (% of Trips within ±60 sec)
Flow model	600	210	1819	100
Network Management Simulation	600	204	1616	95

3. The projected maximum vehicle flow for this run was 686 vehicles per hour. During the simulation run, one link was observed to exceed this flow, and was in fact operating at the theoretical maximum flow of 720 vehicles per hour. This link (No. 19) was situated at a critical point in the network (i.e., at a merge intersection that was immediately followed by a speed reduction zone).

4. System performance was severely degraded with respect to station entry rejection. Out of 3100 station arrivals, there were 43 station entry rejections, which started to appear about 14.5 min into the simulation. All of the station rejections occurred at station No. 4, which just preceded the overutilized link No. 19. The time history of the station entry rejection is plotted in Fig. 4.3, where it can be seen that the rate is fairly constant at about 60 vehicles per hour after 20 min.

5. With respect to service adherence, only 38% of the vehicles were within ±30 sec of their scheduled trip times, and only 68% were within ±60 sec. The service adherence distribution is plotted in Fig. 4.4.

6. While the system was being initialized and before the rejections begin to occur, links No. 7 and No. 51, (both merge links) processed vehicles at a rate of 684 vehicles per hour. These links differed from No. 19 in that there was no speed reduction following these merge junctions.

The primary effect of congestion was to significantly increase the station entry rejection rate. There is evidence of vehicle bunching in the 128 vehicle system; however, there appears to be no unstable network congestion dynamics due to vehicle-follower control, since the link densities seem to approach steady-state values. One obvious effect of congestion was the reduction of average vehicle speed, particularly over the critical links of the network. The speed profile over the critical links for both the 103 and 128 vehicle systems is

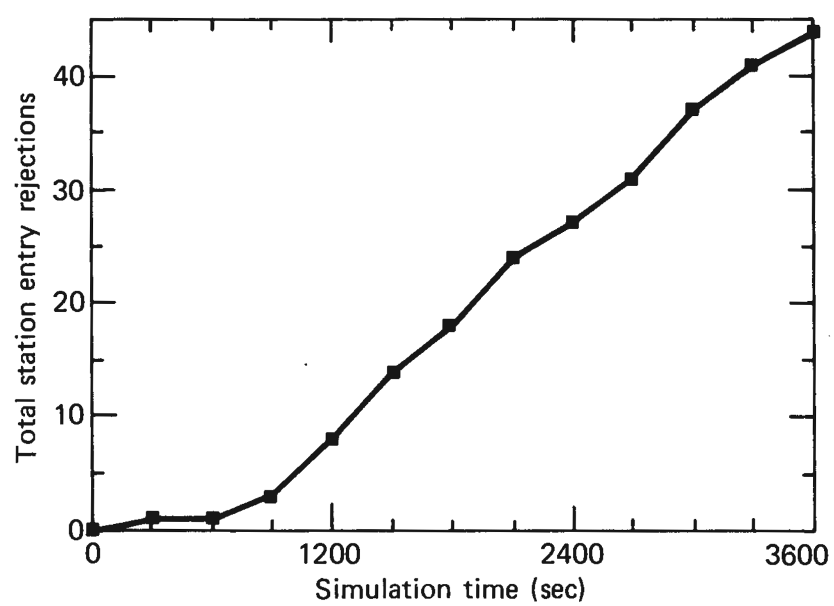

Fig. 4.3 Station entry rejection time, 120 vehicles.

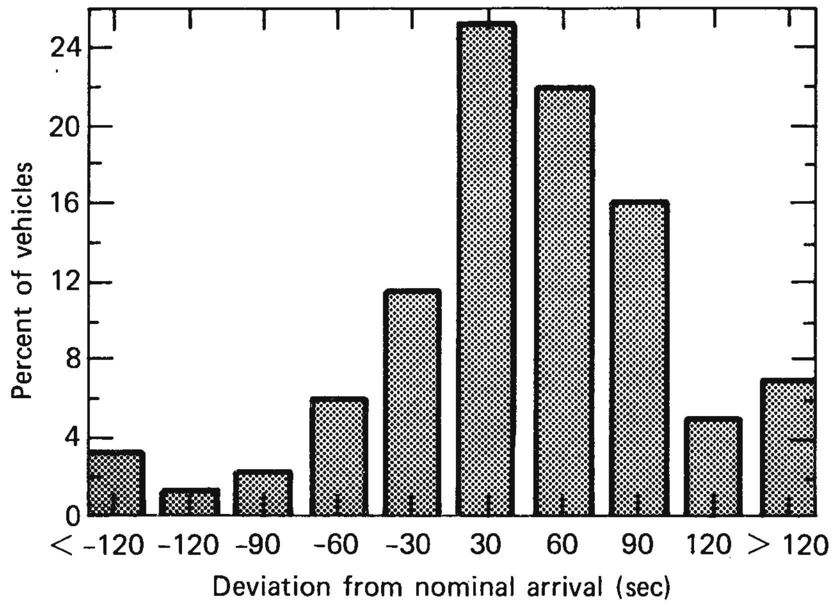

Fig. 4.4 Service adherence, 128 vehicles.

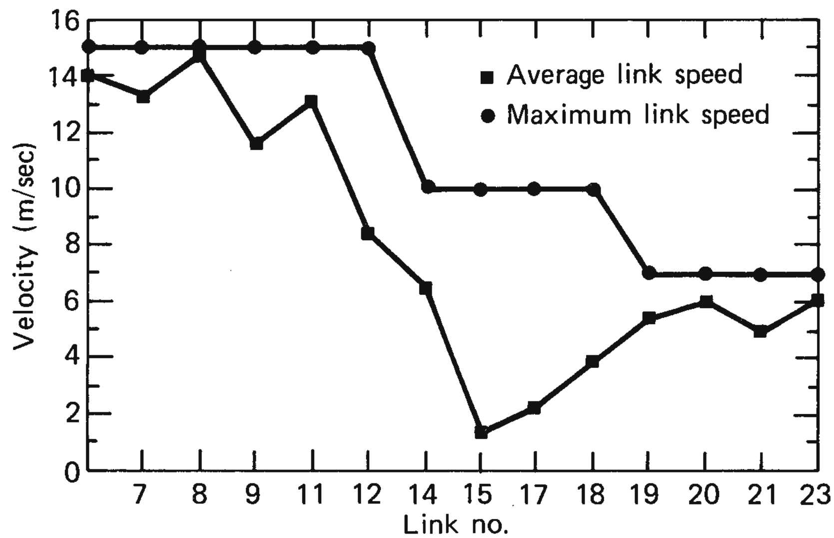

Fig. 4.5 Velocity profile, 128 vehicles.

plotted in Fig. 4.5. The largest slowdown occurs at link No. 15, where there is a merge junction followed by a speed reduction.

4.3.2 Effects of Constant-h versus Constant-k Spacing Policies

The previous runs established the system performance using a constant-h spacing policy. Under this policy, vehicles were operated so as to maintain a 5.0 sec time interval between vehicles. Another spacing policy was investigated, the constant-k policy, in which the desired spacing between vehicles is determined by the braking capability of the vehicle. The distance needed by a vehicle to brake from some velocity, v, to rest, assuming an emergency deceleration rate, a_e, and an infinite jerk limit, is given by

$$d = v^2/(2a_e) \ .$$

The constant-k policy assumes that vehicles will maintain a spacing, s, given by

$$s = kd \ ,$$

where k is a constant. For safety considerations, k was required to be greater than 1.0. For this simulation effort, k was also chosen such that the spacing when traveling at the maximum velocity, v_{max}, would be equal to the spacing commanded by the constant-h policy. Thus, vehicle flows would be comparable under both spacing policies. This means that

$$Kv_{\max}^2/(2a_e) = hv_{\max} \ .$$

Solving for k yields

$$k = 2a_e h/v_{\max} \ ,$$

which yields $k = 1.85$ for $h = 5$ sec, $a_e = 2.8$ g, and $v_{\max} = 15$ m/sec.

In Table 4.2 a comparison between the nominal 103 vehicle run for both constant-h and constant-k is summarized. Overall, marginally better performance was seen in the constant-k system. With constant-k, the average vehicle speed is higher (from 7.0 to 7.3 m/sec), and

Table 4.2

Constant-h versus constant-k, 103 vehicle fleet.

	Constant-h	Constant-k
Average Speed (m/sec)	7.00	7.30
% Vehicle in:		
Vehicle-Follower Mode	35.33	22.33
Velocity-Control Mode	32.67	40.67
Station	32.00	37.00
Total No. of Station Arrivals	2639	2770
Total Vehicle-Miles (mi)	1548.00	1618.7
Station Rejections	1	2
Service Adherence:		
% within ±30 sec	77	73
% within ±60 sec	94	97

the percentage of vehicles arriving within 60 sec of scheduled time is higher (97%, as opposed to 94% for the constant-h case). However, the constant-k system did register one more station rejection that the constant-h. It is interesting to note that the percentage of vehicles in the vehicle follower-mode decreased dramatically while operating in constant-k, allowing corresponding increases in the two other modes.

Table 4.3 gives the results comparing constant-k and constant-h performance when the system is loaded with 128 vehicles. The constant-k policy shows a distinct advantage in system performance. The number of station rejections was about half that of the constant-h run. 85% of the vehicles in constant-k arrived within 60 sec of their scheduled times, as opposed to 68% of the vehicles in the constant-h mode. This example demonstrated the strength of the constant-k policy in maintaining higher vehicle flows due to vehicles being able to operate closer to each other at lower speeds.

A revealing comparison can be made between the constant-h and constant-k policies by looking at where the station rejections occur. For the constant-h case, all the rejections occurred at station No. 2, which is at the beginning of a heavily traveled path. On the other hand, all the rejections for the constant-k case took place at station No. 4, which was at the end of the same path as station No. 2 and immediately before the merge junction. It should be noted that both station No. 2 and No. 4 had about the same level of demand. The reason why rejections occured at station No. 2 for the constant-h case is that the vehicle flow by the station was so great that the input queue to the station was saturated. In the constant-k case, vehicles were packed closer before reaching station No. 2 and thus allowing more time to enter station No. 2. However, the price of this packing became evident at station No. 4 when the packing became a detriment to vehicles attempting to leave the station. The vehicles unable to leave station No. 4 caused vehicles to be rejected at station No. 4.

Table 4.3

Constant-h versus constant-k, 128 vehicle fleet.

	Constant-h	**Constant-k**
Average Speed (m/sec)	5.74	6.92
% Vehicles in:		
Vehicle-Follower Mode	37.50	28.00
Velocity-Control Mode	24.83	33.33
Station	37.67	38.67
Total No. of Station Arrivals	3100	3641
Total Vehicle-Miles (mi)	1685.3	1994.0
Station Rejections	43	19
Service Adherence:		
% within ±30 sec	38	60
% within ±60 sec	68	85

With respect to system performance, the general effect of constant-k appeared more favorable than constant-h. However, there is a possibility that the constant-k spacing policy may violate the kinematic constraint imposed by a safety subsystem. Future work should resolve this question.

4.3.3 System Performance, Station Configuration, and Vehicle Management Interactions

The initial series of runs that investigated the network level behavior of vehicle-follower control demonstrated that steady-state vehicle flows up to 95% of the systems theoretical line capacity could be handled without unstable vehicle queuing at merges, with no significant degradation in service, and without the use of any vehicle management schemes (debunching algorithms or merge arrival metering). The purpose of the second series of runs was to investigate the interaction between station operations (with stations sized to match the steady-state flows) and network flows, and to assess the potential benefits of vehicle management with respect to this interaction. The analysis approach consisted of conducting runs with a variety of station configurations, steady-state vehicle flows, and both with and without the vehicle management logic invoked. As discussed in Section 3, vehicle management was implemented in the form of a reservation matrix that was used to meter the arrival of vehicles desiring to enter each station.

In order to isolate mainline performance of vehicle-follower control from the interaction with station operations, the initial simulation runs discussed in the previous section had all stations in the network modeled as 4/4/2 configurations (i.e., 4 entrance queue positions, 4 berths for vehicle docking, and 2 exit queue locations). This configuration provided more than adequate capacity to handle steady-state vehicle station flows projected by the flow model simulation preprocessor. These configurations were reduced for this series of runs in order to examine the interaction effects of vehicle-follower mainline control and station operations. A 3/3/2 configuration was selected for stations No. 2, 3, 4, and 5 and a 2/2/1 configuration was specified for all other stations. The first run employed a 103 vehicle fleet (prevously found to result in link flows at about 95% of the mainline capacity) and did not invoke the vehicle management reservation scheme. The effect of reducing the station configurations was an increase in the number of station rejections from 1 to 17 for one hour of simulated operation. The majority of these rejections occurred at 2/2/1-configured stations located immediately downstream of merge junctions. This result can be explained by the fact that the flow distribution of vechilces immediately downstream of a merge is less uniform, with a frequent occurrence of a mini-string of vehicles as a result of the merging process. The buffer capacity provided by two entrance queue locations was apparently insufficient to handle the closely spaced arrivals produced by the merging process. Rejections at the second or third stations downstream of merges did not occur, although the steady-state vehicle flows were as great. This can be attributed to the smoothing effect of the first station on the distribution of arrivals at successive stations.

The same run was repeated with the reservation matrix logic invoked. The effect of metering the arrival rate of vehicles at all stations was to reduce the number of rejections from 17 to 10 with no significant degradation in origin/destination travel time or schedule adherence distributions. An additional run was conducted in which the width of the reservation time window was decreased in order to achieve tighter control over the flow distribution. This reduced the number of rejections to 1, but at the cost of numerous vehicle reroutings and station dispatch delays due to the unavailability of slots in the reservation table. The additional improvement in service gained by reducing the number of station rejections substantially below 10 was offset by a decline in the general level of service (e.g., 60% of all trips completed within ±30 sec of nominal time versus 74% with 10 rejections).

The configurations of stations downstream of merge junctions where the rejections of the previous runs occurred were changed from 2/2/1 to 3/2/1 and a final pair of runs was conducted, one with and one without the reservation matrix logic invoked. Increasing the number of entrance queue berths at these stations led to a reduction of vehicle rejections from 17 to 7 without a reservation matrix, and from 7 to 3 with the reservation logic invoked. For both runs the level of service was similar, with only a slight increase in the spread of origin/desination travel time distribution with the reservation logic.

The above runs have demonstrated, for one network configuration, that interactions between mainline operations and stations must be considered in the design of an AGRT system. Stations need to be designed with consideration given to the expected steady-state flow of vehicles to be processed, the network location, the characteristics of the arriving flow distribution, and the possible implementation of vehicle management approaches. A reservation matrix form of vehicle management was found to be an effective means for reducing the number of vehicle rejections at stations. For the network considered, rejections were reduced by 40% to 50% relative to runs without the reservation matrix logic implemented. Greater reductions were possible by changing the reservation scheme parameters but at the expense of a network-wide degradation in performance (reduced trip speeds and poorer schedule adherence). A system design tradeoff of interest was the observation that the implementation of a vehicle reservation scheme reduced vehicle rejections by the same amount as achieved by adding an additional entrance queue position to stations downstream of merge junctions.

Station processing capacities ranged from 55 to 60 vehicles per berth per hour for the 4 berth configuration to 75 to 80 vehicles per berth per hour for the 2 berth configuration. These processing rates apply to stations not immediately downstream of merges. These rates were served without any vehicle rejections and with 80 to 90% of all vehicles processed in 30 to 60 sec, time from mainline exit until vehicle dispatch. The simulation logic assumed fixed dwell times of 15 sec for all vehicles.

4.3.4 Network Energy Consumption Characteristics

Energy consumption characteristics are summarized in Table 4.4. It should be emphasized that these numbers should only be used for relative comparisons between runs on this simulation model. These data do, however, seem plausible when compared to operational data taken from revenue systems. Examination of the data in column 3 of Table 4.4 shows that vehicle maneuvers (other than decelerations into stations) offer a potential for recovering 6.8% to 8.0% of the total energy consumed, depending on the vehicle spacing policy employed. Specifically, the constant-h policy has a slightly smaller energy consumption per vehicle-mile: 1.24 kWh/mi versus 1.27 kWh/mi. The data also shows that a significant amount of energy consumption takes place due to mainline maneuvering. Consequently, a significant amount of energy can theoretically be recovered. Different headway policies seem to influence the amount of mainline maneuvering and, consequently, the energy consumed.

The additional power required for maneuvering can be considerable if the motor and motor controller have efficiencies that vary significantly with vehicle speed. To get some feel

Table 4.4

Energy consumption sensitivities.

Run	Energy per Vehicle-Mile Traveled (kWh)	Regeneration Savings[1] (%)	Losses due to Variable Motor Efficiency[2] (%)
103 Vehicles, Constant-k	1.27	6.8%	15.7%
128 Vehicles, Constant-k	1.28	7.2%	15.3%
103 Vehicles, Constant-h	1.25	7.3%	17.5%
128 Vehicles, Constant-h	1.24	8.0%	23.0%

[1]Assumes 100% recovery of braking energy; includes contributions from all line maneuvers except stopping at stations
[2]Assumes a quadratic falloff in motor efficiency as vehicle speed drops below nominal line speed (15 m/sec)

for the quantitative importance of the effect, energy consumption was calculated using two values for the motor efficiency. One was held constant, the other value was varied using a simple quadractic expression. Using this variable efficiency, energy consumption for the same simulation runs increased from 15.7% up to 23% depending upon the scenario. This data agrees with the earlier results that showed larger or more frequent maneuvering occurs under the constant-h policy. Maneuvering also seems to increase with loading on the network. These results also indicate that energy consumption will be underestimated for simple motor/controller combinations having efficiencies that vary significantly with vehicle speed. Consequently, an accurate model of motor characteristics should be employed to assess the impact of main-line maneuvering on energy consumption.

Other runs changed the efficiency of the motor/drive train combination. As to be expected, power consumption was proportional to improvements in drive train efficiency.

4.4 SIMULATION EFFICIENCY

The duration of each network management simulation run performed in this study was set to a minimum of one hour of simulation time. Table 4.5 is a representative sample of the central processing unit time required for these runs executed on an IBM 3033, as well as the corresponding ratio of the simulated time to central processing unit time. The runs shown in Table 4.5 were those considered in Section 4.3.1.

From Table 4.5, the network management simulation performed with good efficiency even with a heavily congested network. The ratios of simulated time to central processing unit time ranged from 68 to 144 and were achieved without any focused optimization effort. The results indicate that the network management simulation is a cost-effective means of doing sensitivity analyses of many types of AGRT systems.

Table 4.5

Representative sample of network management simulation computational time requirements.

Run	Central Processing Unit Time Required (sec)	Ratio of Simulated Time to Central Processing Unit Time
72 Vehicle Fleet, 33 Routes	25	144
103 Vehicle Fleet, 61 Routes	41	88
128 Vehicle Fleet, 61 Routes	53	68

Note: All runs were done for one hour of simulation time.

REFERENCES

1. H. Y. Chiu, G. B. Stupp, and S. J. Brown, Jr., "Feasibility of Vehicle-Follower Controls for Short Headway AGT Systems," JHU/APL TPR 029, Oct 1974.
2. A. J. Pue, H. Y. Chiu, and S. J. Brown, Jr., "Operational Concepts and Implementation Techniques for Vehicle-Follower Control of AGT Systems," JHU/APL TPR 041, Aug 1979.
3. A. J. Pue, "Control Law Implementation for Short Headway Vehicle-Follower AGT Systems," JHU/APL TPR 045, Oct 197.
4. D. L. Kershner, "Examination of AGRT Station Design and Operational Requirements," JHU/APL BTT-062-81, 17 Aug 1981.
5. D. L. Kershner, "Network Analysis of Advanced Group Rapid Transit Systems," JHU/APL TPR 042, Apr 1979.
6. H. Y. Chiu, "A Vehicle-Follower Precision Stop Controller Design Using the State-Constrained Approach," JHU/APL BTT-056-81, 31 Sep 1981.
7. D. L. Kershner, R. C. Rand, and W. J. Roesler, "Network Model Studies or Automated Guideway Transit: Advanced Group Rapid Transit Models," JHU/APL TPR 033, Feb 1976.

APPENDIX A
NETWORK MANAGEMENT SIMULATION PROGRAM SOURCE LISTING

```
C ****************************************
C *                                      *
C *   NETWORK SIMULATION MAIN4  PROGRAM  *
C *        LAST UPDATED 9/03/82  BY HYC  *
C *              MAIN4X 9/08       PJM   *
C ****************************************
C
       INTEGER*4 RTITLE(4)/4*'    '/
       INTEGER*4 DATE(4)/4*'    '/
       INTEGER*4 FILERS(4)/4*'    '/
       INTEGER*4 FILEDO(4)/4*'    '/
       INTEGER*4 FILEGS(4)/4*'    '/
C
C   COMMON BLOCK INITIALIZATION
       COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *      IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *      IVCS(200,20),IVQCH(200),IVSB(200)
       COMMON /BLKVR/VACC(200),VVEL(200),VPOS(200),VENR(200)
       COMMON /BLKLI/LNTY(90),LNFV(90),LNTV(90),LNEL(90),LNLI(90),
     *      LNRN(90),LNLV(90)
       COMMON /BLKLR/ALMV(90),ALLL(90)
       COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
       COMMON /BLKTI/ITPS(200),ITOS(200),ITDS(200),ITVI(200),
     *       ITSB(200),ITRD(200),ITUB(200),ITQC(200),ITTD(200)
       COMMON /BLKTR/TRAT(200),TRPT(200),TRBT(200),TRCT(200)
       COMMON /BLKEVI/IETY(500),IEID(500),IEPT(500),IEES(500)
       COMMON /BLKEVR/TEVT(500)
       COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
       COMMON /BLKCN/HD,GK,DELTA,AS,XJS,CZ1,CZ2,CZ3,VLEN
       COMMON /BLKLUP/DISTL(90,90),NPNODE(90),IEXIT(90,90)
       COMMON /BLKLV2/LSTN(10)
       COMMON /BLKSCH/IELAST,IEMAX,IEHEAD,IETAIL
       COMMON /BLKSNI/INLINK,INSTAT,INVEHI
       COMMON /BLKSNR/TSLINK,TSSTAT,TSVEHI
       COMMON /BLKVD/VDT,NVDS(200),IVDSP(20),NDSP,NFLT,IDSP
       COMMON /BLKRTA/RTFRQ(80),IRTSTC(80,20),IRTSCH(80,3),NSR,NER,
     *          NSRT(80),NSEQ(80,15)
       COMMON /BLKENR/WPAX,WV,WROT,DRGCF,ARR,BRR,AF,BF,G,EGB,EM,
     *         EPCU,RCPTY
       COMMON /BLKLEN/PENGRY(90),PPWR
       COMMON /BLKXTR/ VDIST(200),JFLGE,ISRN(20)
       COMMON /BLKSA/ IRTSAT(80,10),IRTSAI(80,10)
       COMMON /BLKTRH/ THOD,THST,TAST(200),ITOD(20,20,10),ITTS(20,10)
       COMMON /BLKSTA/ MSTAR(20)
       COMMON /BLKRSV/IVRSV(25,40),PTT(25,25),ATT(25,25),IVTHRD(25),
     *          MIDL(5),MIML(5),WSIZE,IRP,JFLGTM
      COMMON /BLKENG/ PENGYD(90),DELPWR(200),POWER,POWERD,
     +            SWAUX,JFLGED
C
       NAMELIST /INP1/TEND,TINC,TDINC,AS,XJS,HD,VLEN,TDWELL,VMX
     *     ,TSLINK,TSSTAT,TSVEHI,GK,DELTA,NVEH,NLINKS,JFLGP,JFLGE,JFLGS
     *     ,JFLGR,JFLGV,TBPR,TBDO,TBST,TSST,TSPR,JFLGTM,JFLGSM,JFLGED
       NAMELIST /INPERG/WPAX,WV,WROT,DRGCF,ARR,BRR,AF,BF,G,EGB,EM,
     *        EPCU,RCPTY,SWAUX
       NAMELIST /INRSV/WSIZE,MSH,ICYCLT,STHDF,GTHDF,MIDL,MIML
       NAMELIST /INLNEL/LNEL
       NAMELIST /INLNLI/LNLI
       NAMELIST /INLNTY/LNTY
       NAMELIST /INALLL/ALLL
       NAMELIST /INALMV/ALMV
```

```
      NAMELIST /INISTA/ISTA
      NAMELIST /INISTX/ISTX
      NAMELIST /INPNOD/NPNODE
      NAMELIST /INPFP/VDT,NDSP,IVDSP
C
C    INITIALIZE VARIABLES TO ZERO
      IEHEAD=0
      IETAIL=0
      IELAST=200
      INLINK=0
      INSTAT=0
      INVEHI=0
      NFLTO=0
      PPWR=0.0
      POWER=0.0
      POWERD=0.0
      DO 8 I=1,80
      DO 8 J=1,10
      IRTSAT(I,J)=0
 8    IRTSAI(I,J)=0
      DO 10 I=1,200
      VACC(I)=0.
      VVEL(I)=0.
      VPOS(I)=0.
      VENR(I)=0.
      VDIST(I)=0.
      DELPWR(I)=0.
      IVLV(I)=0
      IVLV2(I)=0
      IVTV(I)=0
      IVSS(I)=0
      IVLS(I)=0
      IVRF(I)=0
      IVPX(I)=0
      IVSB(I)=0
      ITPS(I)=0
      ITUB(I)=0
      ITQC(I)=0
      TAST(I)=0.
 10   TRCT(I)=0.
      DO 12 I=1,500
      IETY(I)=0
      IEID(I)=0
      IEPT(I)=0
      IEES(I)=0
      TEVT(I)=0
 12   CONTINUE
      DO 20 I=1,90
      LNTY(I)=0
      LNEL(I)=0
      LNLI(I)=0
      LNLV(I)=0
      NPNODE(I)=0
      PENGRY(I)=0.0
      PENGYD(I)=0.0
 20   CONTINUE
      DO 24 I=1,20
      IVDSP(I)=0
      MSTAR(I)=0
      ISRN(I)=0
```

```
      DO 15 J=4,8
      ISTA(I,J)=0
 15   CONTINUE
      DO 16 J=1,3
      DO 16 K=1,10
      ISQS(I,J,K)=0
 16   CONTINUE
      DO 17 L=1,20
      DO 17 M=1,10
 17   ITOD(I,L,M)=0
      DO 18 N=1,10
 18   ITTS(I,N)=0
 24   CONTINUE
      DO 22 I=1,10
 22   LSTN(I)=0
C
C    DEFAULT OPERATING PARAMETER VALUES
C
C    FLAG VARIABLES: JFLGS = GLOBAL SAVE OPTION
C                    JFLGR = GLOBAL RESTART OPTION
C                    JFLGE = ENERGY STATISTICS OPTION
C                    JFLGP = COMPLETE INPUT PRINTOUT OPTION
C                    JFLGV = VEHICLE DISPATCH OPTION IN RESTART
C                    JFLGSM = DEBUG STATION ID
C                    JFLGTM = TRAFFIC MANAGEMENT OPTION(RESERV. MATRIX)
C                    JFLGED = DETAILED PRINTOUT OPTION OF ENERGY
C
C    ( DEFAULT FLAG VALUES = 0; SET FLAG = 1 TO ENABLE OPTION)
C
      JFLGS=0
      JFLGR=0
      JFLGE=0
      JFLGP=0
      JFLGV=0
      JFLGSM=0
      JFLGTM=0
      JFLGED=0
      AS=1.5
      XJS=2.0
      HD=5.0
      NLINKS=90
      VLEN=6.0
      TDWELL=30.0
      TINC=.25
      TDINC=1.0
      TSST=900.
      TEND=10.
      IEMAX=500
      VMX=15.
      GK=10.
      DELTA=10.
      NVEH=0
      IDSP=0
      VDT=20.
      TSLINK=300.
      TSSTAT=300.
      TSVEHI=10.
      NDSP=4
      WPAX=150.
      WV=12238.0
```

```
      WROT=1000.0
      DRGCF=0.03844
      ARR=0.0230
      BRR=0.0000027
      AF=0.0
      BF=0.0
      G=32.174
      EGB=0.92
      EM=0.88
      EPCU=0.98
      RCPTY=1.0
      SWAUX=1.0
      TBDO=0.0
      TBPR=0.0
      TBST=0.0
      TSPR=300.
      NFLTO=0
      THOD=60.
      THST=30.
      WSIZE=30.
      MSH=5
      ICYCLT=40
      STHDF=1.0
      GTHDF=1.0
      MIDL(1)=51
      MIDL(2)=7
      MIDL(3)=69
      MIDL(4)=41
      MIDL(5)=23
      MIML(1)=51
      MIML(2)=7
      MIML(3)=43
      MIML(4)=73
      MIML(5)=19
C
C    READ SUMMARY INFORMATION
      READ(5,33) RTITLE
 33   FORMAT( 4A4 )
      READ(5,33) DATE
      READ(5,33) FILERS
      READ(5,33) FILEDO
      READ(5,33) FILEGS
C
C    READ OPERATING PARAMETERS
      READ(5,INP1)
      WRITE(6,INP1)
C
C    SET DEBUG STATION FLAG
      ISQS(1,3,10)=JFLGSM
C
C    READ ENERGY PARAMETERS
      READ(5,INPERG)
      WRITE(6,INPERG)
C
C    READ NODE PARAMETERS
      READ(5,INPNOD)
      WRITE(6,INPNOD)
C
C    COMPUTE CONTROLLER CONSTANTS
      CZ1=(VMX/AS+AS/XJS)/2.
```

```
      CZ2=1./2./AS
      CZ3=CZ1
C
C    READ LOOK-UP TABLES
      READ(53) ((DISTL(I,J),I=1,NLINKS),J=1,NLINKS)
      READ(56) ((IEXIT(I,J),I=1,NLINKS),J=1,20)
      READ(55) ((PTT(I,J),I=1,25),J=1,25)
      READ(55) ((ATT(I,J),I=1,25),J=1,25)
C
C    IF JFLGR=0, GO TO "SET NETWORK PARAMETERS"; ELSE DO GLOBAL RESTART.
      IF( JFLGR.EQ.0 ) GO TO 26
C
C    GLOBAL RESTART
      CALL RSTART(NVEH)
      WRITE(6,9991) T
 9991 FORMAT(' T = ',F7.2,': GLOBAL RESTART EXECUTED.')
C
C   RESET STATION DEBUG FLAG
      ISQS(1,3,10) = JFLGSM
C
C    DISPATCH NEW VEHICLES IF JFLGV = 1; ELSE RETURN.
      IF( JFLGV.NE.1 ) GO TO 90
C
C   INITIALIZE VEHICLE FLEET DISPATCH DATA
      NFLTO=NFLT
      READ(5,INPFP)
      READ(54) NSR,NER,NFLT
      READ(54) (NVDS(I),I=1,200)
      READ(54) (RTFRQ(I), I=1,80)
      READ(54) (NSRT(I), I=1,80)
      READ(54) ((NSEQ(I,J),I=1,80), J=1,15)
C
C   INITIALIZE SERVICE ROUTE ARRAYS IRTSTIC,IRTSCH
      DO 45 IRT=1,80
 45   IF( IRT.LE.(NSR+NER) ) IRTSCH(IRT,1)=3600./RTFRQ(IRT)
      WRITE(6,85) IRTSTC
      WRITE(6,86)
      WRITE(6,85) IRTSCH
C
C    SCHEDULE FIRST VEHICLE INTO NETWORK
      CALL SCHED(T,1,9)
      GO TO 90
C
C    SET NETWORK PARAMETERS
C
C     LINK PARAMETERS
 26   READ(5,INLNEL)
      READ(5,INLNLI)
      READ(5,INLNTY)
      READ(5,INALLL)
      READ(5,INALMV)
C
C    CONVERT LINK LENGTHS IN ALLL FROM STANDARD UNITS TO METERS
      CCN=100./3.3
      DO 28 I=1,90
  28  ALLL(I)=ALLL(I)*CCN
C
C     STATION PARAMETERS
      READ(5,INISTX)
C
```

```
C      SET ALL ENTRANCE AND DOCK QUEUES TO 4 BERTHS
C      AND EXIT QUEUES TO 2 BERTHS
       DO 30 I=1,20
       ISTA(I,1)=4
       ISTA(I,2)=4
 30    ISTA(I,3)=2
C
C   NOW, READ IN CHANGES TO DEFAULT STATION BERTHS ALLOCATIONS
       READ(5,INISTA)
C
C   INITIALIZE VEHICLE FLEET DISPATCH DATA
       READ(5,INPFP)
       READ(54) NSR,NER,NFLT
       READ(54) (NVDS(I),I=1,200)
       READ(54) (RTFRQ(I), I=1,80)
       READ(54) (NSRT(I), I=1,80)
       READ(54) ((NSEQ(I,J),I=1,80), J=1,15)
C
C    PRINT OUT ALL NAMELIST INPUTS
       IF( JFLGP.EQ.0 ) GO TO 40
       WRITE(6,INLNEL)
       WRITE(6,INLNLI)
       WRITE(6,INLNTY)
       WRITE(6,INALLL)
       WRITE(6,INALMV)
       WRITE(6,INISTX)
       WRITE(6,INISTA)
       WRITE(6,INPFP)
 40    CONTINUE
C
C   INITIALIZE SERVICE ROUTE ARRAYS IRTSTIC,IRTSCH
      CALL RTDTI
      IF( JFLGP.EQ.0 ) GO TO 87
      WRITE(6,85) IRTSTC
 84   FORMAT(//' IRTSTC',/)
 85   FORMAT(10I6)
      WRITE(6,86)
      WRITE(6,85) IRTSCH
 86   FORMAT(//' IRTSCH,'/)
 87   CONTINUE
C
C    SCHEDULE FIRST VEHICLE INTO NETWORK AND INITIALIZE TIME
       T=0.
       CALL SCHED(0.,1,9)
C
C    SCHEDULE CONSUB AT T=0.
       CALL SCHED(0.,0,1)
       IF(JFLGTM.EQ.0) GO TO 90
C
C    READ IN RESERVATION MATRIX PARAMETERS
       READ(5,INRSV)
       WRITE(6,INRSV)
C    INITIALIZE RESERVATION MATRIX & CELL CAPACITY VECTOR
       DO 81 I=1,20
 81     IVTHRD(I) = STHDF*(ISTA(I,2)*(WSIZE/ICYCLT))
       DO 88 I=21,25
 88     IVTHRD(I) = GTHDF*(WSIZE/MSH)
       DO 89 I=1,25
        DO 89 J=1,40
 89      IVRSV(I,J) = 0
```

```
      IRP = 1
      WRITE(6,8999) (IVTHRD(I),I=1,25)
 8999 FORMAT(25I3)
C     SCHEDULE FIRST UPDATE OF RESERVATION MATRIX COLUMN POINTER
      CALL SCHED(WSIZE,0,13)
C
C
C
C   SCHEDULE FIRST PROUT AT T=T+TBPR. THIS IS ALSO THE BEGINNING OF
C   A GLOBAL RESTART.
 90   TX=T+TBPR
      IF(JFLGR.EQ.0) CALL SCHED(TX,0,7)
C
C   SCHEDULE FIRST DATAOUT AT T=TBDO
      TX=T+TBDO
      CALL SCHED(TX,1,8)
      CALL SCHED(TX,2,8)
      CALL SCHED(TX,3,8)
C
C   SCHEDULE LAST EVENT
      TX=T+TEND
      CALL SCHED(TX,0,11)
C
C   SCHEDULE STATES AT T=TBST
      TX=T+TBST
      CALL SCHED(TX,0,12)
C
C   START SIMULATION.
C
C   GET EVENT TYPE AND ID
 50   IID=IEHEAD
      IITYPE=IETY(IEHEAD)
C
C   UPDATE TIME
      IF( T.LT.TEVT(IID) ) T=TEVT(IID)
C
C   RESET EVENT LIST
      IEES(IEHEAD)=0
      IEHEAD=IEPT(IEHEAD)
C
C   GO TO APPROPIATE SUBROUTINE BY TYPE
      GO TO (100,200,300,400,500,600,700,800,900,1100,1200,1000,750),
     + IITYPE
      IID=5
      GO TO 1100
C
C   EVENT TYPES:
C        1 = CONSUB
C        2 = ENQUE
C        3 = DKQUE
C        4 = DWELL
C        5 = EXQUE
C        6 = TRIP
C        7 = PRINTOUT
C        8 = DATAOUT
C        9 = VEHICLE DISPATCH
C       10 = ERROR
C       11 = TERMINATE
C       12 = STATES
C       13 = RSVUPD
```

```
C
C     CONTINUOUS MODE
 100    CALL CONSUB(NVEH)
        GOTO 50
C
C     ENTRANCE QUEUE
 200    CALL ENQUE(IID)
        GO TO 50
C
C     DOCK QUEUE
 300    CALL DKQUE(IID)
        GO TO 50
C
C     DOCK DWELL
 400    CALL DWELL(IID)
        GO TO 50
C
C     EXIT QUEUE
 500    CALL EXQUE(IID)
        GO TO 50
C
C       TRIP
 600    CALL TRIP
        GO TO 50
C
C     PRINTOUT
 700    IF( T.LT.TBPR ) GO TO 50
        TX=T+TSPR
        CALL PROUT(TX,NVEH)
        GO TO 50
C
C     UPDATE RESERVATION MATRIX POINTER
 750    CALL RSVUPD(T)
        GO TO 50
C
C     DATAOUT
 800    IF( T.LT.TBDO ) GO TO 50
        CALL DATOUT(IID)
        GO TO 50
C
C     VEHICLE DISPATCH
 900    CONTINUE
        CALL VDISP(NVEH,NFLTO)
        GO TO 50
C
C     STATES
 1000   IF( T.LT.TBST ) GO TO 50
        WRITE(6,1010) T
 1010   FORMAT('1STATES: T=',F7.2)
        TX=T+TSST
        CALL STATES(NVEH,TX)
        GO TO 50
C
C     ERROR TRAP
 1100   CALL ERROR(IID)
C
C     TERMINATE PROGRAM. PRINT FINAL STATISTICS.
 1200   WRITE(6,1010) T
        CALL STATES(NVEH,99999.)
        CALL PROUT(99999.,NVEH)
```

```
C
C    PRINT OUT SUMMARY SHEET
       WRITE(6,2000) RTITLE,DATE
 2000  FORMAT('1RUN TITLE:',4A4,20X,'RUN DATE:',4A4/)
       WRITE(6,2010) FILERS
 2010  FORMAT(' RESTART FILE USED: ',4A4/)
       WRITE(6,2020) FILEDO
 2020  FORMAT(' DATA OUT FILE USED: ',4A4,/)
       WRITE(6,2030) FILEGS
 2030  FORMAT(' GLOBAL SAVE FILE USED: ',4A4,/)
C
C    IF JFLGS=1 THEN EXECUTE A GLOBAL SAVE OF SIMULATION VARIABLES.
       IF( JFLGS.NE.1) STOP
       CALL GSAVE(NVEH)
       WRITE(6,9990) T
 9990  FORMAT('1T = ',F7.2,': GLOBAL SAVE EXECUTED.')
C
       STOP
       END
```

```
      SUBROUTINE ADDRSV(IRW,ICL,ISTAT)
C
C   PURPOSE:  TO ADD A VEHICLE RESERVATION TO THE RESERVATION MATRIX.
C     IF THE RESERVATION IS ACCEPTABLE, ADDRSV RETURNS A VALUE OF 1
C     IN ISTAT;  OTHERWISE, RETURNS A VALUE OF 0.
C
C   CALLED BY:  DSPRSV,DVRTRV
C
C   INPUTS:  IRW-ROW OF RESERVATION MATRIX INDICATING LOCATION OF
C                 RESERVATION.
C            ICL-COLUMN OF RESERVATION MATRIX INDICATING TIME OF
C                 RESERVATION.
C            IVTRHD(I)-RESERVATION THRESHOLD FOR EACH LOCATIONI.
C
C   OUTPUTS:  ISTAT-STATUS OF THE RESERVATION ATTEMPT
C             IVRSV(IRW,ICL)-INCREMENTED BY 1 IF CELL THRESHOLD
C                  PERMITS AN ADDITIONAL VEHICLE.
C
C   LAST CHANGE 27-08-82  BY DLK
C
      COMMON /BLKRSV/IVRSV(25,40),PTT(25,25),ATT(25,25),IVTHRD(25),
     *  MIDL(5),MIML(5),WSIZE,IRP,JFLGTM
      IF (IVRSV(IRW,ICL).GE.IVTHRD(IRW)) GO TO 100
      IVRSV(IRW,ICL) = IVRSV(IRW,ICL)+1
      ISTAT = 1
      RETURN
C
C   RESERVATION CANNOT BE ACCOMMODATED.
  100 ISTAT = 0
      RETURN
      END
```

```
      SUBROUTINE BTHANC(IDV)
C
C   PURPOSE:  TO DETERMINE THE PLATFORM BERTH ASSIGNMENT AND
C      ENTRANCE QUEUE POSITION ASSIGNMENT FOR VEHICLES APPROACHING
C      A STATION DIVERT RAMP
C
C   SCHEDULES: DKQUE
C
C   CALLED FROM:  REMOVE (PRIOR TO A CALL TO RTASGN)
C
C   OUTPUT:  IF VEHICLE IS SENT DIRECTLY TO A BERTH, PLACE BERTH
C      ASSIGNMENT INTO IVSB(IDV) AND SCHEDULE DKQUE; OTHERWISE, BERTH
C      ASSIGNMENT IS STORED IN IVLV(IDV) WITH ENTRANCE QUEUE
C      POSITION STORED IN IVSB(IDV)
C
C   INPUT:  IDV--VEHICLE ID
C           VEHICLE DATA COMMON BLOCK(ITEMS: IVLV,IVSB,IVSS)
C           STATION STATUS ARRAYS: ISQS,ISTA
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
      COMMON /BLKEVR/TEVT(500)
C
C   CASE 1: ENQ OCCUPIED
C   CASE 2: ENQ UNOCCUPIED, DQUE UNOCCUPIED
C   CASE 3: ENQ UNOCCUPIED, LAST VEHICLE IN DQUEUE HAS DOCKED OR
C        COMPLETED ITS DWELL.
C   CASE 4: ENQ UNOCCUPIED, LAST VEHICLE IN DQUEUE HAS NOT YET DOCKED
C             (IE, IS WAITING ON IN THE WAY)
C
C
      LID = IVLS(IDV)
      IVLS(IDV) = IVND(IDV)
      IDS = IVLS(IDV)
      ICASE = 0
      IF(ISTA(IDS,4).GT.0)   ICASE=1
      IF(ICASE.NE.0)   GO TO 50
      IF(ISTA(IDS,5).EQ.0)   ICASE=2
      IF(ICASE.NE.0)   GO TO 50
      K = 0
      NDQP = ISTA(IDS,2)
      DO 40 I=1,NDQP
      IF(ISQS(IDS,2,I).NE.0)   K=K+1
   40 IF(K.EQ.ISTA(IDS,5))   GO TO 45
C             I IS MOST UPSTREAM BERTH THAT IS OCCUPIED
C             IDDV IS ID OF VEHICLE OCCUPING BERTH I
   45 IDDV = ISQS(IDS,2,I)
      IF(IDDV.LT.0)   IDDV=-IDDV
      IF((IVLV(IDDV).LT.0) .OR. ((IVLV(IDDV).EQ.I) .AND.
     *(ISQS(IDS,2,I).GT.0)))   ICASE=3
      IF(ICASE.EQ.0)   ICASE=4
C ************* TEMPORARY DIAGNOSTIC WRITES *******************
   50 IF (IDS.NE.ISQS(1,3,10)) GO TO 90
      WRITE(6,51) T,IDV,IDS,ICASE,I,IDDV
   51 FORMAT('  BTHANC CALL AT TIME:  ',F7.2,5(I4))
      WRITE(6,52) ISTA(IDS,4),ISTA(IDS,5),ISTA(IDS,6)
   52 FORMAT(3(I5))
```

```
C
C ******************************************************************
   90 GO TO (100,200,300,400),ICASE
C
C      CASE 1: IF VEHICLES ARE ASSIGNED TO THE ENTRANCE QUEUE, DETERMINE
C    BERTH ASSIGNMENT,I, OF LAST VEHICLE TO ENTER THE QUEUE
C
  100 NEQP = ISTA(IDS,1)
      K = 0
      DO 110 I = 1,NEQP
      IF(ISQS(IDS,1,I).NE.0)   K=K+1
  110 IF(K.EQ.ISTA(IDS,4))  GO TO 120
  120 M = 1
      IF(ISQS(IDS,1,I).LT.0)  M=-1
      LV = M*ISQS(IDS,1,I)
      LSTVAN = IVLV(LV)
C
C      IF THIS ASSIGNMENT IS TO LAST BERTH, ASSIGN VEHICLE IDV TO
C    BERTH NO. 1;  ELSE, ASSIGN TO NEXT BERTH.
C
      IF(LSTVAN.EQ.ISTA(IDS,2))    IVLV(IDV)=1
      IF(LSTVAN.LT.ISTA(IDS,2))    IVLV(IDV)=LSTVAN+1
C
C    ASSIGN VEHICLE IDV TO NEXT AVAILABLE ENTRANCE QUEUE POSITION;
C    RESERVE THIS POSITION IN THE STATION STATUS VECTOR.
C
      IVSS(IDV) = 1
      IVSB(IDV) = I+1
      ISQS(IDS,1,IVSB(IDV)) = -IDV
      ISTA(IDS,4) = ISTA(IDS,4)+1
C
C      DETERMINE TIME AT WHICH VEHICLE WOULD ARRIVE AT ASSIGNED POSITION
C    AND SCHEDULE ENQUE.
C
      CALL TDECEL(IDV,LID,TDEC)
      TEVT(IDV) = T+TDEC
      RETURN
C
C
C      CASE 2: DOCK QUEUE IS EMPTY, ASSIGN VEHICLE IDV TO FIRST BERTH.
C    RESERVE POSITION IN THE STATION STATUS VECTOR.
C
  200 IVSB(IDV) = 1
      IVLV(IDV) = 1
      IVSS(IDV) = 2
      ISQS(IDS,2,1) = -IDV
      ISTA(IDS,5) = 1
C
C      DETERMINE TIME OF VEHICLE ARRIVAL AT BERTH 1 AND SCHEDULE
C    DKQUE EVENT.
C
      CALL TDECEL(IDV,LID,TDEC)
      CALL SCHED(T+TDEC,IDV,3)
      RETURN
C
C      CASE 3: THE MOST UPSTREAM DQUEUE VEHICLE HAS DOCKED OR COMPLETED
C    ITS DWELL.  ASSIGN VEHICLE IDV TO BERTH 1 AND TARGET FOR  MOST
C    DOWNSTREAM ACCESSIBLE BERTH OR TO  ENTRANCE QUEUE POSITION IF DOCK
C    IS INACCESSIBLE.
C
```

```
  300 IF(I.EQ.ISTA(IDS,2))  GO TO 350
C              (LAST DQUEUE VEH, IDDV, IS NOT IN LAST BERTH)
      IVLV(IDV) = 1
      IVSB(IDV) = I+1
      IVSS(IDV) = 2
      ISQS(IDS,2,I+1) = -IDV
      ISTA(IDS,5) = ISTA(IDS,5)+1
      GO TO 380
C              (LAST DQUEUE VEH IS IN LAST BERTH)
  350 IVLV(IDV) = 1
      IVSB(IDV) = 1
      IVSS(IDV) = 1
      ISQS(IDS,1,1) = -IDV
      ISTA(IDS,4) = 1
C
  380 CALL TDECEL(IDV,LID,TDEC)
      TEVT(IDV) = T+TDEC
      RETURN
C
C     CASE 4: LAST VEHICLE HAS NOT YET DOCKED. ASSIGN VEHICLE IDV TO
C   NEXT DOCK POSITION AND SCHEDULE INTO MOST DOWNSTREAM BERTH OR
C   TO ENTRANCE QUEUE POSITION 1 IF DOCK IS INACCESSIBLE.
C
  400 IF(I.EQ.ISTA(IDS,2)) GO TO 450
C               (LAST DQUEUE VEH, IDDV, IS NOT IN LAST BERTH)
      IVLV(IDV) = IVLV(IDDV)+1
      IVSB(IDV) = I+1
      IVSS(IDV) = 2
      ISQS(IDS,2,I+1) = -IDV
      ISTA(IDS,5) = ISTA(IDS,5)+1
      CALL TDECEL(IDV,LID,TDEC)
      IF(IVLV(IDV).NE.IVSB(IDV)) GO TO 480
      CALL SCHED(T+TDEC,IDV,3)
      RETURN
C               (LAST VEH IN DQUEUE IS IN LAST BERTH)
  450 IVLV(IDV) = IVLV(IDDV)+1
      IF(IVLV(IDDV).EQ.ISTA(IDS,2))  IVLV(IDV)=1
      IVSB(IDV) = 1
      IVSS(IDV) = 1
      ISQS(IDS,1,1) = -IDV
      ISTA(IDS,4) = 1
C
      CALL TDECEL(IDV,LID,TDEC)
  480 TEVT(IDV) = T+TDEC
      RETURN
      END
```

```
      SUBROUTINE CNCLRV(IRW,ICL,T)
C
C   PURPOSE:  TO CANCEL A RESERVATION AT LOCATION IRW AND TIME ICL
C        OF THE VEHICLE RESERVATION MATRIX.
C
C   CALLED BY:  DVRTRV
C
C   INPUTS:  IRW-ROW OF RESERVATION MATRIX INDICATING LOCATION OF
C                RESERVATION TO BE CANCELLED.
C            ICL-COLUMN OF RESERVATION MATRIX INDICATING TIME OF
C                RESERVATION TO BE CANCELLED.
C
C   OUTPUTS:  IVRSV(IRW,ICL)- DECREMENTED BY 1
C
C   LAST CHANGE 27-08-82  BY DLK
C
      COMMON /BLKRSV/IVRSV(25,40),PTT(25,25),ATT(25,25),IVTHRD(25),
     *   MIDL(5),MIML(5),WSIZE,IRP,JFLGTM
      IF (IVRSV(IRW,ICL).LE.0)  GO TO 100
      IVRSV(IRW,ICL) = IVRSV(IRW,ICL)-1
      RETURN
  100 WRITE(6,200) IRW,ICL,IVRSV(IRW,ICL)
  200 FORMAT('  CNCLRV: RESERV. DOES NOT EXIST--(IRW,ICL,IVRSV) ',3I5)
      RETURN
      END
```

```
      SUBROUTINE CONTR(AP,AT,VP,VT,SS,VMAX,IL,ACC,MODE)
C
C    PURPOSE: TO GENERATE ACCELERATION COMMANDS AND SELECT MODE
C
C    INPUTS: AP,VP=ACCEL AND VEL OF PRECEDING VEH
C            AT,VT=ACCEL AND VEL OF TRAILING VEH
C            SS=SPACING
C            VMAX=MAXIMUM LINK SPEED
C            IL=PRECEDING VEH ID
C
C    OUTPUTS: ACC=ACCELERATION COMMAND
C             MODE=CONTROL MODE
C
C    LAST UPDATE: 21-SEP-82 BY HYC
C
C    NOTES: THIS CONTR NOW USES A CONSTANT-K CONTROL LAW.
C         (KIN. CONST. TERM, CKT,  LIMITED TO <0 VALUES)
C
C
      COMMON /BLKCN/HD,GK,DELTA,AS,XJS,CZ1,CZ2,CZ3,VLEN
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
C
      DATA HDK,AM/1.850,3.0/
C
C    START. COMPUTE ERROR STATES
      AE=AP-AT
      VE=VP-VT
C
C    IF VT > VMAX, GO TO "VELOCITY-COMMAND"
      IF( VT.GT.VMAX ) GO TO 100
C

C    IF LEAD VEH=0, GO TO "VELOCITY-COMMAND"
      IF( IL.EQ.0 ) GO TO 100
C
C    IF VT<VMIN=7.0 M/S, SET VT=7.
      VTG=VT
      IF( VT.LT.7. ) VTG=7.
C
C    COMPUTE CONTROLLER E AND U
C * CONSTANT K IMPLEMENTED WITH HDK = K-FACTOR, AM = EMERGENCY BRAKE *
      CKT = CZ1*VE+CZ2*VTG*VE
      IF(CKT.GT.0.0) GO TO 10
      E=SS+CKT-VLEN-HDK*VT*VT/AM/2.
      U=(CZ1*AE+CZ2*(VE*AT+AE*VT)-HDK*VT*AT/AM+VE)/GK
      IF( VT.LT.7.) U=(CZ3*AE-HDK*VT*AT/AM+VE)/GK
      GO TO 20
   10 E=SS-VLEN-HDK*VT*VT/AM/2.0
      U=(-HDK*VT*AT/AM+VE)/GK
C
C    IF E > DELTA, THEN GO TO "VELOCITY-COMMAND"
   20 IF( E.GT.DELTA ) GO TO 100
C
C    REGULATION MODE
      MODE=1
      E=E-GK*AT
      E=E/DELTA
      IF( ABS(E).GT.1. ) E=SIGN(1.0,E)
      UL=U-ABS(E)*U+XJS*E
      IF( ABS(UL).GT.XJS ) UL=SIGN(XJS,UL)
```

```
      ACC=AT+UL*TINC
C
C     DO NOT ALLOW VEHICLE TO GO BACKWARDS
      IF( VT.LT..001 .AND. ACC.LT.0. ) GO TO 50
C
C     SERVICE LIMIT THE ACCELERATION COMMAND
      ACC=AT+UL*TINC
      IF( ABS(ACC).LT.AS ) RETURN
      ACC=SIGN(AS,ACC)
      RETURN
   50 VT=0.0
      ACC=0.0
      RETURN
C
C     VELOCITY-COMMAND MODE
  100 VCE=VMAX-VT
      MODE=2
      UL=SIGN(XJS,VCE)
      IF( ABS(VCE).GT.0.70 ) GO TO 150
      VT=VMAX
      UL=0.
      ACC=0.
      RETURN
  150 ACC=AT+UL*TINC
      IF( ABS(ACC).LT.AS ) RETURN
      ACC=SIGN(AS,ACC)
      RETURN
      END
```

```
      SUBROUTINE COLUMN(IRP,RDT,WSIZE,ICOL)
C
C   PURPOSE:  TO CONVERT A CONTINUOUS FUTURE TIME, T+RDT, INTO THE
C        APPROPRIATE COLUMN OF THE RESERVATION MATRIX.
C
C   CALLED BY:  DSPRSV,DVRTRV
C
C   INPUTS:  IRP-POINTER TO FIRST COLUMN OF RESERVATION MATRIX
C            RDT-INTERVAL UNTIL DESIRED TIME(SEC)
C            WSIZE-WIDTH OF RESERVATION TIME WINDOW(SEC)
C
C   OUTPUTS:  ICOL-APPROPRIATE COLUMN FOR FUTURE TIME
C
C   LAST CHANGE 27-08-82  BY DLK
C
      I = (RDT/WSIZE)+1.0
C
C       (ROUND UP TO NEAREST INTEGER COLUMN)
      ICOL = IRP+I
      IF (ICOL.GT.40) ICOL=ICOL-40
C     WRITE(6,100) RDT,WSIZE,IRP,ICOL
C 100 FORMAT(' FROM COLUMN  RDT,WSIZE,IRP,ICOL: ',2F7.2,2I5)
      RETURN
      END
```

```
      SUBROUTINE CONSUB(NVEH)
C
C    PURPOSE: TO UPDATE VEHICLE STATES IN CONTINUOUS MODE
C
C    INPUT: NVEH = NUMBER OF VEHICLES IN SYSTEM
C
C   LAST CHANGE 15-09-82  BY DLK
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *     IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *     IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKVR/VACC(200),VVEL(200),VPOS(200),VENR(200)
      COMMON /BLKLI/LNTY(90),LNFV(90),LNTV(90),LNEL(90),LNLI(90),
     *     LNRN(90),LNLV(90)
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      COMMON /BLKLR/ALMV(90),ALLL(90)
      COMMON /BLKLUP/DISTL(90,90),NPNODE(90),IEXIT(90,90)
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
      COMMON /BLKLV2/LSTN(10)
      COMMON /BLKENR/WPAX,WV,WROT,DRGCF,ARR,BRR,AF,BF,G,EGB,EM,
     *        EPCU,RCPTY
      COMMON /BLKLEN/PENGRY(90),PPWR
      COMMON /BLKXTR/ VDIST(200),JFLGE,ISRN(20)
      COMMON /BLKRSV/IVRSV(25,40),PTT(25,25),ATT(25,25),IVTHRD(25),
     *    MIDL(5),MIML(5),WSIZE,IRP,JFLGTM
      COMMON /BLKSTA/MSTAR(20),LFLOW(90)
C
C    IF NVEH=0, GO TO "RESCHEDULE NEXT CONSUB..."
      IF( NVEH.EQ.0 ) GO TO 9000
C
C    +++++++++++++++++++++++++++++++++++++++++++++
     IF(JFLGE.GT.0) JFLGE=JFLGE+1
C    +++++++++++++++++++++++++++++++++++++++++++++
C
C ******************************************************************
C ******************************************************************
C
C    START VEH INTEGRATION LOOP
C
      DO 50 I=1,NVEH
C
C    IF IN DISCRETE MODE, SKIP TO NEXT VEH
      IF( IVCM(I).EQ.3 ) GO TO 50
C
C    SET UP FOR ENERGY VARIABLES
      VOLD=VVEL(I)
C
C    RECTANGULAR INTEGRATION OF VEH STATES
      VPOS(I)=VPOS(I)+(VVEL(I)+VACC(I)*TINC/2.)*TINC
      VVEL(I)=VVEL(I)+VACC(I)*TINC
C
C    COLLECT ENERGY STATISTICS IF JFLGE=1
     IF(JFLGE.LT.5) GO TO 50
      CALL VENGRY(I,IVPX(I),VOLD,VVEL(I),VACC(I))
C    END OF VEH INTEGRATION LOOP
 50   CONTINUE
C
C   +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
     IF(JFLGE.EQ.5) JFLGE=1
C   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
C
C
C    START VEH CONTROL LOOP
C
C      IF( T.GE.117. ) WRITE(6,94) T
C94    FORMAT(/,' IN CONSUB: TIME = ',F7.2)
C      IF(T.LT.117)GO TO 99
C95    FORMAT(' I=',I3,';FIRST=',I3,';LAST=',I3,';TOTAL=',I3)
C      DO 97 IX=1,90
C97    WRITE(6,95)  IX,LNFV(IX),LNLV(IX),LNTV(IX)
C99    CONTINUE
C
       DO 1000 I=1,NVEH
C
C      IF( T.GE.117. ) WRITE(6,96) I
C96    FORMAT(' IN CONSUB: VEHICLE ',I3)
C
C    IF IN DISCRETE MODE, SKIP TO NEXT VEH
       IF( IVCM(I).EQ.3 ) GO TO 1000
C
C    IF VEH HAS NOT REACHED LINKEND, GO TO "IF NO LEAD..."
       IF( VPOS(I).LE.ALLL(IVLS(I)) ) GO TO 400
C
C ********************************************************************
C
C    BEGIN LINK CROSSING UPDATE
C
C    UPDATE VEHICLE DISTANCE TRAVELED
       VDIST(I)=VDIST(I)+ALLL(IVLS(I))
       LOLD=IVLS(I)
C
C  UPDATE LINK FLOW
       LFLOW(LOLD) = LFLOW(LOLD)+1
C
C    LINK TYPES:
C       1 = NOMINAL
C       2 = PARALLEL
C       3 = STATION ENTRY
C       4 = STATION BYPASS
C       5 = STATION EGRESS
C       6 = DIVERT
C
C    CHECK OLD LINK TYPE TO FIND NEW LINK
       LTYPE=LNTY(LOLD)
       IVLO=IVLV(I)
       GO TO (100,120,100,120,120,140),LTYPE
C
C    NOMINAL, STATION ENTRY LINKS.
 100   LNEW=LNEL(LOLD)
       GO TO 160
C
C    PARALLEL, STATION EGRESS, STATION BYPASS LINKS.
C    SET 2ND LEAD VEH = 0
 120   IVLV2(I)=0
       IVLV2(IVTV(I))=0
       LNEW=LNEL(LOLD)
C
C    IF OLD LINK WAS PARALLEL, RESET LAST VEH IN REGION
       IF( LNTY(LOLD).EQ.2 .AND. LSTN(NPNODE(LOLD)).EQ.I )
     *     LSTN(NPNODE(LOLD))=0
```

```
      GO TO 160
C
C   DIVERT LINK.
C    DIVERT LINK:  CALL DVRTRV TO RECONFIRM CURRENT RESERVATION; IF
C    INVALID, CHANGE RESERVATION AND CANCEL OLD RESERVATION.
C
  140 LNEW = IEXIT(LOLD,IVND(I))
      IF(JFLGTM.EQ.0)  GO TO 160
C ****************************************************************
C         TEMPORARY DIAGNOSTIC WRITE STATEMENTS
C     IF(IVND(I).NE.2) GO TO 145
C     WRITE(6,150) T,LOLD,I
C 150 FORMAT(' CONSUB CALL TO DVRTRV(T,LOLD,I) ',F7.2,2I5)
C     WRITE(6,155) (IVRSV(2,J), J=1,40)
C 155 FORMAT(40I2)
C
C ******************************************************************
  145 CALL DVRTRV(LOLD,I,IVND(I),T)
C ******************************************************************
C      TEMPORARY DIAGNOSTIC WRITE STATEMENTS
C     IF(IVND(I).NE.2) GO TO 158
C     WRITE(6,157) LNEW,IVQCH(I)
C 157 FORMAT(' LNEW,IVQCH: ',2I5)
C     WRITE(6,155) (IVRSV(2,J), J=1,40)
C
C *******************************************************************
  158 IF (IVQCH(I).GT.0)  GO TO 160
      IF(PTT(LOLD,IVND(I)).EQ.ATT(LOLD,IVND(I)))  GO TO 160
C
C      VEHICLE IDV IS NOW ASSIGNED TO AN ALTERNATE PATH.
      IF(LNEW.EQ.LNEL(LOLD)) LNEW=LNLI(LOLD)
      IF(LNEW.NE.LNEL(LOLD)) LNEW=LNEL(LOLD)
C
C ==================================================================
C
C   FIND LEAD VEH
C
C160   IF( T.GE.115. ) WRITE(6,161) T
C161   FORMAT(' NEWLV CALLED AT T = ',F7.2/)
 160   CALL NEWLV(I,LNEW,LOLD,IVLO)
C
C   PERFORM LINK CHANGES
C      IF( T.GE.115. ) WRITE(6,361) T
C 361  FORMAT(/,' LINKUP CALLED AT T = ',F7.2/)
       IF( LNEW.NE.0 ) CALL LINKUP(I,LOLD,LNEW)
C      IF( T.LT.117 ) GO TO 299
C      DO 297 IX=1,90
C297   WRITE(6,95)  IX,LNFV(IX),LNLV(IX),LNTV(IX)
C299   CONTINUE
C
C   IF LNEW=0, THEN VEHICLE IS IN DISCRETE MODE AND WE SHOULD SKIP
C   TO THE NEXT VEHICLE.
       IF( LNEW.EQ.0 ) GO TO 1000
C
C   CHANGE CURRENT LINK ID AND VEH POS
 310   IVLS(I)=LNEW
       VPOS(I)=VPOS(I)-ALLL(LOLD)
C
C   END OF LINK CROSSING UPDATE
C
```

```
C ***********************************************************************
C
C    BEGIN VEHICLE COMMAND COMPUTATION
C
C    IF NO LEAD VEH, GO TO "VELOCITY-COMMAND MODE"
 400   IL=IVLV(I)
       IF( IL.EQ.0 ) GO TO 500
C
C    DETERMINE VEH STATES
       AP=VACC(IL)
       VP=VVEL(IL)
       AT=VACC(I)
       VT=VVEL(I)
       VMAX=ALMV( IVLS(I) )
       SSL=DISTL( IVLS(I),IVLS(IL) )
       SS=VPOS(IL)-VPOS(I)+SSL
C
C    CHECK CURRENT LINK TYPE FOR TWO LEAD VEHS
       LNKTY=LNTY(IVLS(I))
       GO TO (420,440,420,440,440,420), LNKTY
C
C    ONLY ONE LEAD VEH
C
 420   CALL CONTR( AP,AT,VP,VT,SS,VMAX,IL,VACC(I),IVCM(I) )
       VVEL(I)=VT
       GO TO 1000
C
C    IF NOT ONE LEAD VEHS, GO TO "TWO LEAD VEHS..."
 440   IL2=IVLV2(I)
       IF( IL2.NE.0 ) GO TO 445

       IF( IVLS(IL).NE.LNLI(IVLS(I)) ) GO TO 420
C
C    RECOMPUTE SPACING
       SS=ALLL(IVLS(I))-ALLL(IVLS(IL))-VPOS(I)+VPOS(IL)
       GO TO 420
C
C    TWO LEAD VEHS. IF IL IS ON COMP LINK, GO TO "COMPUTE SPACING..."
 445   IF( IVLS(IL).EQ.LNLI(IVLS(I)) ) GO TO 480
C
C    COMPUTE FIRST VEH ACC
       CALL CONTR(AP,AT,VP,VT,SS,VMAX,IL,ACC1,MODE1)
C
C    COMPUTE SECOND VEH ACC
 450   SS=VPOS(IL2)-VPOS(I)+DISTL(IVLS(I),IVLS(IL2))
       AP=VACC(IL2)
       VP=VVEL(IL2)
       CALL CONTR(AP,AT,VP,VT,SS,VMAX,IL2,ACC2,MODE2)
C
C    CHOOSE MIN ACC
       IF( ACC1.GT.ACC2 ) GO TO 460
       VACC(I)=ACC1
       IVCM(I)=MODE1
       GO TO 1000
 460   VACC(I)=ACC2
       IVCM(I)=MODE2
       GO TO 1000
C
C     COMPUTE SPACING FOR FIRST LEAD VEHICLES.
 480   SS=ALLL(IVLS(I))-ALLL(IVLS(IL))-VPOS(I)+VPOS(IL)
```

```
      CALL CONTR(AP,AT,VP,VT,SS,VMAX,IL,ACC1,MODE1)
C
C      COMPUTE SPACING FOR SECOND LEAD VEHICLE.
      GO TO 450
C
C    VELOCITY-COMMAND MODE
 500  CALL CONTR(0.0,VACC(I),0.0,VVEL(I),0.0,ALMV(IVLS(I)),0,
     *           ACCV,IVCM(I))
      VACC(I)=ACCV
C
C    END OF VEHICLE COMMAND COMPUTATION
C
C *********************************************************************
C
C    END OF VEH CONTROL LOOP
 1000 CONTINUE
C
C *********************************************************************
C *********************************************************************
C
C    RESCHEDULE NEXT CONSUB EVENT AT T=T+TINC
 9000 TX=T+TINC
      CALL SCHED(TX,0,1)
C
      RETURN
      END
```

```
      SUBROUTINE CONTR(AP,AT,VP,VT,SS,VMAX,IL,ACC,MODE)
C
C    PURPOSE: TO GENERATE ACCELERATION COMMANDS AND SELECT MODE
C
C    INPUTS: AP,VP=ACCEL AND VEL OF PRECEDING VEH
C            AT,VT=ACCEL AND VEL OF TRAILING VEH
C            SS=SPACING
C            VMAX=MAXIMUM LINK SPEED
C            IL=PRECEDING VEH ID
C
C    OUTPUTS: ACC=ACCELERATION COMMAND
C             MODE=CONTROL MODE
C
      COMMON /BLKCN/HD,GK,DELTA,AS,XJS,CZ1,CZ2,CZ3,VLEN
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
C
C    START. COMPUTE ERROR STATES
      AE=AP-AT
      VE=VP-VT
C
C    IF VT > VMAX, GO TO "VELOCITY-COMMAND"
      IF( VT.GT.VMAX ) GO TO 100
C

C    IF LEAD VEH=0, GO TO "VELOCITY-COMMAND"
      IF( IL.EQ.0 ) GO TO 100
C
C    IF VT<VMIN=7.0 M/S, SET VT=7.
      VTG=VT
      IF( VT.LT.7. ) VTG=7.
C
C    COMPUTE CONTROLLER E AND U
      E=SS+(CZ1-HD)*VE+CZ2*VTG*VE-VLEN-HD*VT
      U=((CZ1-HD)*AE+CZ2*(VE*AT+AE*VT)-HD*AT+VE)/GK
      IF( VT.LT.7.) U=((CZ3-HD)*AE-HD*AT+VE)/GK
C
C    IF E > DELTA, THEN GO TO "VELOCITY-COMMAND"
      IF( E.GT.DELTA ) GO TO 100

C    REGULATION MODE
      MODE=1
      E=E-GK*AT
      E=E/DELTA
      IF( ABS(E).GT.1. ) E=SIGN(1.0,E)
      UL=U-ABS(E)*U+XJS*E
      IF( ABS(UL).GT.XJS ) UL=SIGN(XJS,UL)
      ACC=AT+UL*TINC
C
C    DO NOT ALLOW VEHICLE TO GO BACKWARDS
      IF( VT.LT..001 .AND. ACC.LT.0. ) GO TO 50
C
C    SERVICE LIMIT THE ACCELERATION COMMAND
      ACC=AT+UL*TINC
      IF( ABS(ACC).LT.AS ) RETURN
      ACC=SIGN(AS,ACC)
      RETURN
   50 VT=0.0
      ACC=0.0
      RETURN
C
```

```
C     VELOCITY-COMMAND MODE
 100    VCE=VMAX-VT
        MODE=2
        UL=SIGN(XJS,VCE)
        IF( ABS(VCE).GT.0.70 ) GO TO 150
        VT=VMAX
        UL=0.
        ACC=0.
        RETURN
 150    ACC=AT+UL*TINC
        IF( ABS(ACC).LT.AS ) RETURN
        ACC=SIGN(AS,ACC)
        RETURN
        END
```

```
      SUBROUTINE CONTR(AP,AT,VP,VT,SS,VMAX,IL,ACC,MODE)
C
C     PURPOSE: TO GENERATE ACCELERATION COMMANDS AND SELECT MODE
C
C     INPUTS: AP,VP=ACCEL AND VEL OF PRECEDING VEH
C             AT,VT=ACCEL AND VEL OF TRAILING VEH
C             SS=SPACING
C             VMAX=MAXIMUM LINK SPEED
C             IL=PRECEDING VEH ID
C
C     OUTPUTS: ACC=ACCELERATION COMMAND
C              MODE=CONTROL MODE
C
C     LAST UPDATE: 14-SEP-82 BY HYC
C
C     NOTES: THIS CONTR NOW USES A CONSTANT-K CONTROL LAW.
C
      COMMON /BLKCN/HD,GK,DELTA,AS,XJS,CZ1,CZ2,CZ3,VLEN
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
C
      DATA HDK,AM/1.850,3.0/
C
C     START. COMPUTE ERROR STATES
      AE=AP-AT
      VE=VP-VT
C
C     IF VT > VMAX, GO TO "VELOCITY-COMMAND"
      IF( VT.GT.VMAX ) GO TO 100
C

C     IF LEAD VEH=0, GO TO "VELOCITY-COMMAND"
      IF( IL.EQ.0 ) GO TO 100
C
C     IF VT<VMIN=7.0 M/S, SET VT=7.
      VTG=VT
      IF( VT.LT.7. ) VTG=7.
C
C     COMPUTE CONTROLLER E AND U
C * CONSTANT K IMPLEMENTED WITH HDK = K-FACTOR, AM = EMERGENCY BRAKE *
      E=SS+CZ1*VE+CZ2*VTG*VE-VLEN-HDK*VT*VT/AM/2.
      U=(CZ1*AE+CZ2*(VE*AT+AE*VT)-HDK*VT*AT/AM+VE)/GK
      IF( VT.LT.7.) U=(CZ3*AE-HDK*VT*AT/AM+VE)/GK
C
C     IF E > DELTA, THEN GO TO "VELOCITY-COMMAND"
      IF( E.GT.DELTA ) GO TO 100
C
C     REGULATION MODE
      MODE=1
      E=E-GK*AT
      E=E/DELTA
      IF( ABS(E).GT.1. ) E=SIGN(1.0,E)
      UL=U-ABS(E)*U+XJS*E
      IF( ABS(UL).GT.XJS ) UL=SIGN(XJS,UL)
      ACC=AT+UL*TINC
C
C     DO NOT ALLOW VEHICLE TO GO BACKWARDS
      IF( VT.LT..001 .AND. ACC.LT.0. ) GO TO 50
C
C     SERVICE LIMIT THE ACCELERATION COMMAND
      ACC=AT+UL*TINC
```

```
      IF( ABS(ACC).LT.AS ) RETURN
      ACC=SIGN(AS,ACC)
      RETURN
 50   VT=0.0
      ACC=0.0
      RETURN
C
C   VELOCITY-COMMAND MODE
100   VCE=VMAX-VT
      MODE=2
      UL=SIGN(XJS,VCE)
      IF( ABS(VCE).GT.0.70 ) GO TO 150
      VT=VMAX
      UL=0.
      ACC=0.
      RETURN
150   ACC=AT+UL*TINC
      IF( ABS(ACC).LT.AS ) RETURN
      ACC=SIGN(AS,ACC)
      RETURN
      END
```

```
      SUBROUTINE DATOUT(ITYPE)
C
C    PURPOSE: TO OUTPUT DATA FOR POST-PROCESSING
C
C    INPUT: ITYPE = TYPE OF DATA TO BE OUTPUT
C                   1: LINK
C                   2: STATION
C                   3: VEHICLE
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *       IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *       IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKVR/VACC(200),VVEL(200),VPOS(200),VENR(200)
      COMMON /BLKLI/LNTY(90),LNFV(90),LNTV(90),LNEL(90),LNLI(90),
     *       LNRN(90),LNLV(90)
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
      COMMON /BLKSNI/INLINK,INSTAT,INVEHI
      COMMON /BLKSNR/TSLINK,TSSTAT,TSVEHI
C
C    START. CHECK DATA TYPE.
      IF(ITYPE) 100,200,300
C
C    LINK TYPE.
 100  INLINK=INLINK+1
      WRITE(81) INLINK,T,LNTV
      TX=T+TSLINK
      CALL SCHED(TX,1,8)
      RETURN
C
C    STATION DATA.
 200  INSTAT=INSTAT+1
      WRITE(82) INSTAT,T,ISTA,ISQS
      TX=T+TSSTAT
      CALL SCHED(TX,2,8)
      RETURN
C
C    VEHICLE DATA.
 300  INVEHI=INVEHI+1
      WRITE(83) INVEHI,T,IVCM,IVSS,IVSB,IVLS,IVPX,IVRF,VPOS,VVEL,VENR
      TX=T+TSVEHI
      CALL SCHED(TX,3,8)
      RETURN
C
      END
```

```
      SUBROUTINE DKQUE(IDV)
C     PURPOSE:  TO SIGNIFY VEHICLE ARRIVAL AT ITS ASSIGNED BERTH
C       AND TO SCHEDULE END OF DWELL; TO ASSIGN A NEW TARGET
C       LOCATION FOR VEHICLE SHIFTS IN DOCK QUEUE.IF VEHICLE CAN
C       BE SHIFTED FROM ENTRANCE QUEUE TO DOCK, A DEQUEUING ACTION
C       ACTION IS TAKEN VIA A CALL TO ENQUE.
C
C     SCHEDULED BY: BTHANC,ENQUE,DKQUE(VIA RESCHD)
C
C     CALLED BY:  DWELL,EXQUE
C
C     CALLS:  RESCHD,ENQUE
C
C     OUTPUT:  SCHEDULES END-OF-DWELL,DKQUE EVENTS
C              ASSIGNS NEW TARGET QUEUE POSITIONS, ARRIVAL TIMES
C              UPDATES STATION AND VEHICLE STATUS PARAMETERS
C
C     INPUT: IDV-VEHICLE ID
C            VEHICLE STATUS VECTORS
C            STATION STATUS ARRAYS
C           (PASSENGER DEBOARD/BOARD LIST)
C
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
      COMMON /BLKEVR/TEVT(500)
      IDS = IVLS(IDV)
      IBN = IVSB(IDV)
C *******************************************************************
C     CASES:                                                        *
C       1-IDV ARRIVAL AT ASSIGNED BERTH(SCHEDULED EVENT)            *
C                                                                   *
C       2-IDV IS ON WAY TO ASSIGNED BERTH OR IS UNDERGOING ITS DWELL*
C                                                                   *
C       3-IDV IS ON WAY TO OR STOPPED AT A BERTH OTHER THAN         *
C         ITS ASSIGNED BERTH AND HAS NOT COMPLETED ITS DWELL        *
C                                                                   *
C       4-IDV HAS COMPLETED ITS DWELL AND IS WAITING AT OR ON WAY   *
C         TO A DOCK QUEUE LOCATION                                  *
C                                                                   *
C *******************************************************************
C
    5 ICASE = 0
      IF(IVLV(IDV).LT.0)  ICASE=4
      IF(ICASE.GT.0)  GO TO 10
      IF(IVSB(IDV).NE.IVLV(IDV))  ICASE=3
      IF(ICASE.GT.0)  GO TO 10
      IF(TEVT(IDV).GT.T)  ICASE=2
      IF(TEVT(IDV).LE.T)  ICASE=1
C *********TEMPORARY DIAGNOSTIC WRITES ****************************
   10 IF(IDS.NE.ISQS(1,3,10)) GO TO 90
      WRITE(6,20) T,IDV,IDS,ICASE,IVLV(IDV),IVSB(IDV)
   20 FORMAT('   DKQUE CALL AT TIME: ',F7.2,5(I4))
      IBTH = IVSB(IDV)
      IF(IBTH.LT.0) IBTH =-IBTH
      WRITE(6,30) ISTA(IDS,4),ISTA(IDS,5),ISTA(IDS,6),ISQS(IDS,2,IBTH)
```

```
   30 FORMAT(4(I5))
C
C *****************************************************************
   90 GO TO(100,200,300,400),ICASE
C
C   CASE 1:  IDV ARRIVAL AT ASSIGNED BERTH
  100 ISQS(IDS,2,IVSB(IDV)) = IDV
C
C       DETERMINE NUMBER OF PASSENGER DEBOARDINGS/BOARDINGS AND
C       COMPUTE LENGTH OF DWELL
C
C   **********************************************
C   *  FUTURE LOGIC TO COMPUTE DWELL LENGTH       *
C   *                                            *
C   **********************************************
C
C       SCHEDULE END OF DWELL
      CALL SCHED(T+TDWELL,IDV,4)
      RETURN
C
C       CASE 2:  IDV IS CURRENTLY ON WAY TO ASSIGNED BERTH -- NO
C          ACTION NECESSARY
  200 RETURN
C
C       CASE 3:  IDVHAS NOT YET DOCKED AND ITS CURRENT TARGET LOCATION
C      .  IS NOT ITS ASSIGNED BERTH
C
C       DETERMINE MOST DOWNSTREAM BERTH ACCESSIBLE TO IDV
  300 NP = IVSB(IDV)-IVLV(IDV)
C           (NP=NO. OF POSITIONS FROM CURRENT TARGET TO ASSIGNED BERTH)
      LOC = IVLV(IDV)
      DO 310 I=1,NP
      IP = IVSB(IDV)-I
  310 IF(ISQS(IDS,2,IP).NE.0)  GO TO 320
      GO TO 330
  320 LOC = IP+1
  330 IF(LOC.EQ.IVSB(IDV)) RETURN
      CALL RESCHD(IDV,IDS,2,IVSB(IDV),2,LOC,IVLV(IDV))
      ISQS(IDS,2,IVSB(IDV)) = 0
      ISQS(IDS,2,LOC) = -IDV
      IVSB(IDV) = LOC
      GO TO 500
C
C       CASE 4: IDV HAS COMPLETED ITS DWELL
C
C      DETERMINE MOST DOWNSTREAM BERTH ACCESSIBLE TO IDV
  400 LOC = 1
      IF(IVSB(IDV).EQ.1) GO TO 435
      NP = IVSB(IDV)-1
      DO 410 I=1,NP
      IP = IVSB(IDV)-I
  410 IF(ISQS(IDS,2,IP).NE.0)  GO TO 420
      GO TO 430
  420 LOC = IP+1
  430 IF(LOC.GT.1) GO TO 470
C
C      LOC=1: CHECK IF IDV CAN BE SHIFTED INTO EXIT QUEUE
  435 IF(ISQS(IDS,3,ISTA(IDS,3)).NE.0)  GO TO 470
C
C      DETERMINE MOST DOWNSTREAM ACCESSIBLE EXIT QUEUE POSITION
```

```
      LOC = 1
      NP = ISTA(IDS,3)-1
      DO 440 I=1,NP
      IP = ISTA(IDS,3)-I
  440 IF(ISQS(IDS,3,IP).NE.0)  GO TO 450
      GO TO 460
  450 LOC = IP+1
  460 CALL RESCHD(IDV,IDS,2,IVSB(IDV),3,LOC,IVLV(IDV))
      ISQS(IDS,2,IVSB(IDV)) = 0
      ISQS(IDS,3,LOC) = -IDV
      ISTA(IDS,5) = ISTA(IDS,5)-1
      ISTA(IDS,6) = ISTA(IDS,6)+1
      IVSB(IDV) = LOC
      IVSS(IDV) = 4
      GO TO 500
C
C   IDV CANNOT BE SHIFTED TO EXIT QUEUE
  470 IF(LOC.EQ.IVSB(IDV)) RETURN
      CALL RESCHD(IDV,IDS,2,IVSB(IDV),2,LOC,IVLV(IDV))
      ISQS(IDS,2,IVSB(IDV)) = 0
      ISQS(IDS,2,LOC) = -IDV
      IVSB(IDV) = LOC
      IVSS(IDV) = 3
C
C      DOES IDV HAVE A TRAILING VEHICLE THAT CAN BE SHIFTED FORWARD?
C
  500 LBN = ISTA(IDS,2)
      IB = IBN
      DO 510 I=IB,LBN
      IF(ISQS(IDS,2,I).EQ.0)  GO TO 510
      IDTV = ISQS(IDS,2,I)
      IF(IDTV.LT.0)  IDTV=-IDTV
      IF(IVSB(IDTV).EQ.IVLV(IDTV)) RETURN
C           (IE, NEXT VEHICLE IS ON WAY OR IS AT ITS ASSIGNED BERTH)
C
C      RESET IDV,IBN AND LOOP BACK THRU DKQUE TO SHIFT UP VEHICLE IDTV
      IDV = IDTV
      IBN = IVSB(IDTV)
      GO TO 5
  510 CONTINUE
C
C      NO OTHER VEHICLES IN DOCK QUEUE; CALL ENQUE FOR POSSIBLE
C      SHIFT OF VEHICLES FROM ENTRANCE QUEUE TO DOCK
      IF(ISQS(IDS,2,ISTA(IDS,2)).NE.0)  RETURN
      IF(ISTA(IDS,4).EQ.0)  RETURN
      LBN = ISTA(IDS,1)
      DO 520 I=1,LBN
  520 IF(ISQS(IDS,1,I).NE.0)  GO TO 530
  530 IDV = ISQS(IDS,1,I)
      IF(IDV.LT.0)  IDV=-IDV
      CALL ENQUE(IDV)
      RETURN
      END
```

```
      SUBROUTINE DSPRSV(IDOS,IDV,T,IDFLG)
C
C   PURPOSE:  TO RESERVE SLOTS FOR A VEHICLE PRIOR TO STATION
C      DISPATCH.  IF RESERVATIONS CANNOT BE MADE, AN ALTERNATE PATH
C      IS CONSIDERED AND SLOT RESERVATIONS ARE REQUESTED.  IF THE
C      REQUEST IS REJECTED, AN EGRESS EVENT(EXQUE) IS RESCHEDULED OR,
C      IF THE ORIGIN STATION IS CONGESTED, THE ALTERNATE PATH SLOT
C      IS OVERBOOKED AND THE VEHICLE IS ALLOWED TO DEPART.
C
C   CALLED BY: EXQUE
C
C   SCHEDULES:  EXQUE
C
C   CALLS: ADDRSV,COLUMN
C
C   INPUTS:  IDOS-ID OF STATION REQUESTING A DISPATCH RESERVATION
C            IDV-ID OF VEHICLE TO BE DISPATCHED
C            T-CURRENT TIME
C            IDFLG-IF STATION IS CONGESTED =1 (ENT QUEUE FILLED & NO
C                                               VEH. UNDERGOING DWELL)
C                  OTHERWISE = 0
C
C   OUTPUTS:  IDFLG-SET TO 1 IF RESERVATION IS MADE; 0 OTHERWISE
C                   (EXQUE IS RESCHEDULED IF 0 IS RETURNED)
C             IVRSV- SLOT RESERVATION MADE IF FEASIBLE
C             IVQCH(IDV)-SET TO + ICOL IF A RESERVATION MADE USING
C                          THE PRIMARY PATH
C                        SET TO - ICOL IF A RESERVATION MADE USING
C                             THE ALTERNATE PATH
C    LAST CHANGE 27-08-82  BY DLK
C
C
      COMMON /BLKRSV/IVRSV(25,40),PTT(25,25),ATT(25,25),IVTHRD(25),
     *   MIDL(5),MIML(5),WSIZE,IRP,JFLGTM
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *  IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *  IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      IDDS = IVND(IDV)
C
C      CHECK AVAILABILITY OF RESERVATION USING PRIMARY PATH.  IF
C   UNAVAILABLE, CHECK ALTERNATE PATH.
C
      RDT = PTT(IDOS,IDDS)
      CALL COLUMN(IRP,RDT,WSIZE,ICL)
      CALL ADDRSV(IDDS,ICL,ISTAT)
      IF (ISTAT.EQ.0) GO TO 100
C
C      RESERVATION WAS MADE, SET IDFLG & IVQCH AND RETURN.
C
      IDFLG = 1
      IVQCH(IDV) = ICL
      RETURN
C
C      CHECK AVAILABILITY OF RESERVATION USING ALTERNTE PATH.
C
  100 RDTA = ATT(IDOS,IDDS)
      CALL COLUMN(IRP,RDTA,WSIZE,ICLA)
      IF (RDTA.EQ.RDT) GO TO 200
C               (AN ALTERNATE PATH IS NOT AVAILABLE)
```

```
      CALL ADDRSV(IDDS,ICLA,ISTAT)
      IF (ISTAT.EQ.0) GO TO 200
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C      RESERVATION BY ALTERNATE PATH IS AVAILABLE.  IF STATION IS NOT
C      CONGESTED AND A VEHICLE IS NOT WAITING IN FIRST BERTH TO ADVANCE
C      TO EXIT QUEUE, COMPARE ALT. & PRIMARY PATH TIMES.  IF TIME
C      DIFFERENCE IS GREATER THAN TWO WINDOW WIDTHS AND A  RESERVATION
C      IS AVAILABLE VIA PRIMARY PATH  DURING NEXT SLOT, RESCHEDULE
C      EXQUE AND CANCEL RESERVATION USING ALTERNATE PATH.
      IF(IDFLG.EQ.1) GO TO 150
      IDBV = ISQS(IDOS,2,1)
      IF(IDBV.LT.0) IDBV=-IDBV
      IF(IVLV(IDBV).LT.0) GO TO 150
      DTAP = ATT(IDOS,IDDS)-PTT(IDOS,IDDS)
      IF(DTAP.LE.(2.0*WSIZE)) GO TO 150
      ICLP= ICL+1
      IF(ICLP.GT.40)  ICLP=40-ICLP
      IF(IVRSV(IDDS,ICLP).GE.(IVTHRD(IDDS)-1)) GO TO 150
      CALL CNCLRV(IDDS,ICLA,T)
      GO TO 300
C         NOTE:  TO INVOKE ABOVE LOGIC, LABEL NEXT STATEMENT 150
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C      RESERVATION WAS MADE USING ALTERNATE PATH.  SET IDFLG & IVQCH
C   AND RETURN.
C
  150 IDFLG = 1
      IVQCH(IDV) = -ICLA
      RETURN
C
C      A RESERVATION COULD NOT BE MADE.  IF STATION IS CONGESTED,
C   OVERBOOK SLOT USING ALTERNATE PATH AND RETURN;  OTHERWISE,
C   RESCHEDULE EXQUE.
C
  200 IF (IDFLG.EQ.0) GO TO 300
      IVRSV(IDDS,ICLA) = IVRSV(IDDS,ICLA)+1
      IDFLG = 1
      IVQCH(IDV) = ICLA
      IF(RDTA.NE.RDT) IVQCH(IDV)=-ICLA
      RETURN
C
C      STATION IS NOT CONGESTED, RESCHEDULE EXQUE.
C
  300 TX = T+5.0
      CALL SCHED(TX,IDV,5)
      IDFLG = 0
      RETURN
      END
```

```
      SUBROUTINE DWELL(IDV)
C
C     PURPOSE:  TO TERMINATE DWELL AND SCHEDULE EXIT QUEUE ARRIVAL
C         FOR VEHICLE IDV AND ANY TRAILING VEHICLES THAT HAVE
C         COMPLETED DWELLS.  IF VEHICLES CAN BE SHIFTED FROM ENTRANCE
C         QUEUE TO DOCK, A DEQUEUING ACTION IS TAKEN BY A CALL TO ENQUE
C
C     SCHEDULED BY:  DKQUE
C
C     CALLS: DKQUE
C
C     OUTPUT:  TERMINATES DWELL AND CALLS DKQUE TO INITIATE A SHIFT
C                TO MOST DOWNSTREAM ACCESSIBLE POSITION.
C              UPDATES VEHICLE AND STATION STATUS ARRAYS.
C
C     INPUT:  IDV-VEHICLE ID
C             STATION STATUS ARRAYS
C             VEHICLE STATUS VECTORS
C
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
C
C     SET VARIABLE VALUES;  DETERMINE NUMBER OF TRAILING VEHICLES WITH
C   COMPLETED DWELLS.
C
      IDS = IVLS(IDV)
      IBN = IVSB(IDV)
      IVLV(IDV) = -IVLV(IDV)
C
C     IF NEXT DOCK BERTH IS ACCESSIBLE, CALL DKQUE TO SHIFT IDV
C    FORWARD.  OTHERWISE, RETURN.
C
      IF(IBN.LE.1) GO TO 10
      IF(ISQS(IDS,2,IBN-1).NE.0)  RETURN
   10 CALL DKQUE(IDV)
      RETURN
      END
```

```
      SUBROUTINE ENQUE(IDV)
C
C   PURPOSE: TO INITIATE A SHIFT WITHIN THE ENTRANCE QUEUE
C     OR TO SCHEDULE A DOCK ARRIVAL EVENT IF THE VEHICLE'S ASSIGNED
C     BERTH IS PRESENTLY ACCESSIBLE(VEH IS TARGETED TO A DOCK POSITION
C     IF ASSIGN. BERTH IS INACCESSIBLE).  ENQUE WILL SHIFT VEHICLE IDV
C     PLUS ANY TRAILING VEHICLES FORWARD IN ENTRANCE QUEUE
C     OR TO DOCK AREA.
C
C     CALLED BY:
C        DKQUE--TO DEQUEUE VEHICLES FROM ENTRANCE QUEUE AND SHIFT
C               TO DOCK AS VEHICLES LEAVE DOCK AREA FOR EXIT QUEUE
C
C     CALLS: RESCHD
C
C     OUTPUT:  SHIFTS VEHICLES FORWARD; UPDATES VEHICLE AND
C       STATION STATUS PARAMETERS; SCHEDULES BERTH ARRIVALS(VIA RESCHD)
C
C     INPUT:  IDV- ID OF FIRST VEHICLE IN ENTRANCE QUEUE
C             VEHICLE STATUS VECTORS
C             STATION STATUS ARRAYS
C
C     NOTE:  IMPLICIT TO STATION EVENT LOGIC IS THE DECISION TO SCHEDULE
C        EVENTS ONLY FOR ARRIVAL AT THE ASSIGNED BERTH, FOR END-OF-DWELL
C        AND FOR ARRIVAL AT FIRST EXIT QUEUE POSITION.  FOR OTHER TYPES
C        OF SHIFTS, THE TARGET QUEUE LOCATION IS DETERMINED AND AN
C        ARRIVAL TIME IS COMPUTED AND STORED IN TEVT(VEH. ID). TARGET
C        QUEUE LOCATIONS ARE CHANGED AND TEVT'S UPDATED WHILE VEH. ARE
C        IN MOTION VIA STATION DEQUEUING LOGIC.

      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
      COMMON /BLKEVR/TEVT(500)
C
C     INITIALIZE PARAMETERS
C
      IDS = IVLS(IDV)
      IBN = IVSB(IDV)
C
C     DETERMINE NO. OF ENQUEUED TRAILING VEHICLES
C
      NTV = 0
      IDT = 0
C **********CHECK VALUE FOR DT  SCHEDULING DELAY FOR TRAILING VEH *****
      DT = 0.1
      IF (IBN.EQ.ISTA(IDS,1))    GO TO 15
      IS = IBN+1
      IH = ISTA(IDS,1)
      DO 10 I=IS,IH
   10 IF (ISQS(IDS,1,I).NE.0)   NTV=NTV+1
C
C     DETERMINE MOST DOWNSTREAM DOCK OR ENT. QUEUE POSITION AVAILABLE.
C
      NDP = ISTA(IDS,2)
      NEP = ISTA(IDS,1)
C **********TEMPORARY DIAGNOSTIC WRITES ******************************
   15 IF(IDS.NE.ISQS(1,3,10)) GO TO 19
```

```
      WRITE(6,16) T,IDV,IDS,IVSB(IDV),IVLV(IDV),NTV,ISTA(IDS,4),
     +ISTA(IDS,5)
   16 FORMAT('   CALL TO ENQUE AT TIME= ',F7.2,7(I5))
      WRITE(6,17) (ISQS(IDS,1,I),I=1,5),ISQS(IDS,2,ISTA(IDS,2))
   17 FORMAT(5(I6),I8)
C ***********************************************************************
   19 IF(ISQS(IDS,2,NDP).NE.0)  GO TO 50
      DO 20 I=1,NDP
      LOC = NDP-I+1
   20 IF(ISQS(IDS,2,LOC).NE.0)  GO TO 25
      LOC = 0
C     SET LOC TO MOST DOWNSTREAM ACCESSIBLE BERTH
C
   25 LOC = LOC+1
C
C     CALL RESCHD TO DETERMINE ARRIVAL TIME AT NEW TARGET BERTH AND
C     TO SCHEDULE A DKQUE EVENT IF BERTH IS IDV'S ASSIGNED BERTH
      CALL RESCHD(IDV,IDS,1,IVSB(IDV),2,LOC,IVLV(IDV))
C
C     UPDATE STATION STATUS
      ISTA(IDS,4) = ISTA(IDS,4)-1
      ISTA(IDS,5) = ISTA(IDS,5)+1
      ISQS(IDS,1,IVSB(IDV)) = 0
      ISQS(IDS,2,LOC) = -IDV
      IVSB(IDV) = LOC
      IVSS(IDV) = 2
      GO TO 100
C
C     DOCK QUEUE IS INACCESSIBLE, DETERMINE NOST DOWNSTREAM ACCESSIBLE
C     ENTRANCE QUEUE POSITION
   50 LB = IVSB(IDV)-1
      DO 60 I=1,LB
      LOC = IVSB(IDV)-I
   60 IF(ISQS(IDS,1,LOC).NE.0)  GO TO 70
      LOC = 0
   70 LOC = LOC+1
C     CALL RESCHD TO DETERMINE ARRIVAL TIME AT NEW TARGET BERTH.
C
      CALL RESCHD(IDV,IDS,1,IVSB(IDV),1,LOC,IVLV(IDV))
C
C     UPDATE STATION STATUS
      ISQS(IDS,1,IVSB(IDV)) = 0
      ISQS(IDS,1,LOC) = -IDV
      IVSB(IDV) = LOC
C
C     IF ANY REMAINING TRAILING VEHICLES, REPEAT LOGIC TO SHIFT FORWARD
  100 IF(NTV.EQ.0)  GO TO 1000
C
C     DETERMINE ID OF NEXT TRAILING VEHICLE
      IB = IBN+1
      DO 200 I=IB,NEP
  200 IF(ISQS(IDS,1,I).NE.0)  GO TO 210
  210 IBN = I
      IDV = ISQS(IDS,1,IBN)
      IF(IDV.LT.0)  IDV=-IDV
      NTV = NTV-1
      GO TO 15
 1000 RETURN
      END
```

```
      SUBROUTINE ENTRY(IS,IR)
C
C     PURPOSE: TO DETERMINE WHETHER A VEHICLE CAN ENTER A
C              STATION FROM THE MAINLINE
C
C     INPUT: IS = STATION ID
C
C     OUTPUT: IR = REJECTION FLAG
C                  0: CLEAR TO ENTER STATION
C                  1: MUST BYPASS STATION (LAST QUEUE POSTION FILLED)
C
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      COMMON /BLKSTA/ MSTAR(20)
C
C     UPDATE ARRIVAL COUNTER FOR STATION IS.
      MSTAR(IS)=MSTAR(IS)+1
C
C     IF LAST BERTH IN ENTRANCE QUEUE IS OPEN, SET IR=0; ELSE, IR=1.
      IR=1
      IF( ISQS(IS,1,ISTA(IS,1)) .EQ. 0 ) IR=0
C
      RETURN
      END
```

```
      SUBROUTINE ERROR(MSG)
C
C     PURPOSE: TO PRINT AN ERROR MESSAGE, AND TERMINATE SIMULATION
C
C     INPUT: MSG = ERROR MESSAGE NUMBER
C
       COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
C
C     START. WRITE ERROR MESSAGE.
      WRITE(6,10) T
 10   FORMAT(//' *** TERMINAL ERROR OCCURRED AT T = ',F9.2,' ***')
C
C     GO TO APPROPIATE ERROR MESSAGE PRINTOUT.
      GO TO (100,200,300,400,500), MSG
      WRITE(6,20) MSG
 20   FORMAT(' ILLEGAL MSG VALUE = ',I5,/)
      RETURN
C
C     ERROR IN NEWLV
 100  WRITE(6,110) MSG
 110  FORMAT(' ERROR',I3,' IN NEWLV: ILLEGAL NEW LINK TYPE (EGRESS).',/)
      RETURN
C
C     ERROR IN LINKUP
 200  WRITE(6,210) MSG
 210  FORMAT(' ERROR',I3,' IN LINKUP: ILLEGAL LINK TYPE FOR LOLD.',/)
      RETURN
 300  WRITE(6,210) MSG
 310  FORMAT(' ERROR',I3,' IN LINKUP: IDTV ILLEGALLY EQUAL TO ZERO.',/)
      RETURN
 400  WRITE(6,210) MSG
 410  FORMAT(' ERROR',I3,' IN LINKUP: TOO MANY LOOPS IN LOLD UPDATE.',/)
      RETURN
C
C     ERROR IN MAIN
 500  WRITE(6,210) MSG
 510  FORMAT(' ERROR',I3,' IN MAIN: ILLEGAL EVENT TYPE.',/)
      RETURN
      END
```

```
      SUBROUTINE EXQUE(IDV)
C
C     PURPOSE:  TO SIGNIFY ARRIVAL OF A VEHICLE AT THE FIRST
C       EXIT QUEUE POSITION OR TO REATTEMPT A VEHICLE DISPATCH.
C       ENQUEUED TRAILING VEHICLES ARE SHIFTED FORWARD
C       AND WHEN POSSIBLE VEHICLES ARE DEQUEUED FROM DOCK QUEUE VIA A
C       CALL TO DKQUE.
C
C     SCHEDULED BY:  DKQUE(VIA RESCHD) OR EXQUE(DISPATCH RETRY)
C                    VDISP(FOR VEHICLE FLEET ADJUSTMENTS)
C
C     CALLS: DKQUE,LEADV
C
C     OUTPUT:  DISPATCHES A VEHICLE AND SHIFTS TO CONTINUOUS MODE
C              SCHEDULES EXQUE
C              CALLS DKQUE(DEQUEUING OF DOCK VEHICLES)
C              UPDATES VEHICLE AND STATION STATUS PARAMETERS
C
C     INPUT:  IDV-VEHICLE ID
C             STATION STATUS ARRAYS
C             VEHICLE STATUS AND STATE VECTORS
C             NETWORK ARRAYS
C
C   LAST CHANGE 27-08-82  BY DLK
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
      COMMON /BLKVR/VACC(200),VVEL(200),VPOS(200),VENR(200)
      COMMON /BLKLI/LNTY(90),LNFV(90),LNTV(90),LNEL(90),LNLI(90),
     *  LNRN(90),LNLV(90)
      COMMON /BLKLR/ALMV(90),ALLL(90)
      COMMON /BLKLUP/DISTL(90,90),NPNODE(10),NODE(90),IEXIT(90,90)
      COMMON /BLKCN/HD,GK,DELTA,AS,XJS,CZ1,CZ2,CZ3,VLEN
      COMMON /BLKTRH/ THOD,THST,TAST(200),ITOD(20,20,10),ITTS(20,10)
      COMMON /BLKRSV/IVRSV(25,40),PTT(25,25),ATT(25,25),IVTHRD(25),
     *   MIDL(5),MIML(5),WSIZE,IRP,JFLGTM
C
C   INITIALIZE VARIABLES
C
      IDS = IVLS(IDV)
      IBN = IVSB(IDV)
      ISQS(IDS,3,IBN) = IDV
C
C       IF EGRESS LINK IS NOT CLEAR, RESCHEDULE EXQUE
      IF(LNTV(ISTX(IDS)).EQ.0) GO TO 50
      CALL SCHED(T+5.0,IDV,5)
      RETURN
C
C ********************** DIAGNOSTIC WRITE  *************************
   50 IF (IDS.NE.ISQS(1,3,10)) GO TO 105
      WRITE(6,100) T,IDV,IDS,ISTA(IDS,6),(ISQS(IDS,3,I),I=1,5)
  100 FORMAT('   CALL TO EXQUE AT TIME= ',F7.2,3(I5),5(I5))
C
C  ******************************************************************
C  ******************************************************************
C
C   ++++++++++++++++++++++++++++++++++++++++++++++++
```

```
C   + CAN A DISPATCH BE SCHEDULED FOR VEHICLE IDV?  +
C   +++++++++++++++++++++++++++++++++++++++++++++++++
C
  105 IF(JFLGTM.EQ.0)  GO TO 300
C
C      CHECK STATUS OF STATION. IDFLG IS SET TO 1 IN CALL TO DSPRSV IF
C      STATION IS CONGESTED(ALL ENT. QUEUE POSITIONS FILLED AND NO
C      VEHICLES CURRENTLY UNDERGOING DWELLS) AND AN IMMEDIATE DISPATCH
C      WOULD ALLEVIATE THE CONGESTION.
      IDFLG = 0
C         ENTRANCE QUEUE CHECK
      NENQ = ISTA(IDS,1)-ISTA(IDS,4)
      IF (NENQ.GE.1)  GO TO 190
C
C      DOCK QUEUE CHECK.  IF ANY VEHICLES ARE CURRENTLY UNDERGOING
C      A DWELL, ENTRANCE QUEUE VEHICLES CANNOT BE SHIFTED.
      IDQ = ISTA(IDS,2)
      DO 110 I = 1,IDQ
      IDVEH = ISQS(IDS,2,I)
      IF (IDVEH.LE.0)  GO TO 110
      IF (IVLV(IDVEH).GT.0)  GO TO 190
  110 CONTINUE
      IDFLG = 1
C
C      STATION IS CONGESTED, IMMEDIATE DISPATCH WILL PERMIT FORWARD
C      SHIFT.
C ******************************************************************
C           TEMPORARY WRITE STATEMENTS
C 190 IF(IVND(IDV).NE.2) GO TO 200
C     WRITE(6,210) T,IDS,IDV,IDFLG
C 210 FORMAT(' EXQUE CALL TO DSPRSV(T,IDS,IDV,IDFLG) ',F7.2,3I5)
C     WRITE(6,220) (IVRSV(2,J), J=1,40)
C 220 FORMAT(40I2)
C
C ******************************************************************
  190 CALL DSPRSV(IDS,IDV,T,IDFLG)
C ******************************************************************
C           TEMPORARY DIAGNOSTIC WRITE STATEMENTS
C     IF(IVND(IDV).NE.2) GO TO 250
C     WRITE(6,230) IDFLG,IVQCH(IDV)
C 230 FORMAT(' IDFLG,IVQCH ',2I5)
C     WRITE(6,220) (IVRSV(2,J), J=1,40)
C
C ******************************************************************
  250 IF (IDFLG.EQ.0)  RETURN
C          (IE, NO RESERVATION WAS MADE   EXQUE RESCHEDULED IN 5 SEC)
C
C ******************************************************************
C ******************************************************************
C
C    WINDOW SEARCH. FIND EGRESSING VEH A LEAD VEH ON MAINLINE.
 300    LX=ISTX(IDS)
        LBP=LNLI(LX)
        IFOL=0
C       WRITE(6,305) IDV,LX,LBP,LNTV(LBP)
C305    FORMAT(' IN EXQUE:',4I4)
        IF( LNTV(LBP).EQ.0 ) GO TO 340
        IDLV=LNFV(LBP)
        VMAX=ALMV(LBP)
        SI=HD*VMAX-VMAX*(VMAX/AS+AS/XJS)/2.
```

```
      WINDOW=ALLL(LBP)-ALLL(LX)+SI
      SWNDW=WINDOW-HD*VMAX
 310  XMV=VPOS(IDLV)
      IF( XMV.LT.WINDOW .AND. XMV.GT. SWNDW ) GO TO 330
      IF( XMV.GT.WINDOW ) IFOL=IDLV
      IF( XMV.LT.SWNDW ) GO TO 320
      IDLV=IVTV(IDLV)
      IF ( IDLV.EQ.0 ) GO TO 325
      IF( IVLS(IDLV).NE.LBP) GO TO 325
      GO TO 310
C
C    NO VEH IN EGRESS WINDOW. ASSIGN VEH UPSTREAM OF WINDOW TO FOLLOW
C    EGRESSING VEHICLE
 320  IF( IDLV.EQ.0 ) GO TO 325
      IF(IVCM(IDLV).NE.3) IVLV(IDLV)=IDV
C
C    ASSIGN VEH TO FOLLOW FIRST VEH PAST WINDOW
 325  IF( IFOL.EQ.0 ) GO TO 340
      IVLV(IDV)=IFOL
      IDLV=IFOL
      GO TO 350
C
C    ASSIGN VEH TO FOLLOW FIRST VEH IN WINDOW
 330  IVLV(IDV)=IDLV
      GO TO 350
C
C    FIND LEADV DOWNSTREAM OF EXIT LINK
 340  LDX=LNEL(LX)
      CALL LEADV(IDV,LDX,IDLV)
      IVLV(IDV)=IDLV
      ITV = 0
      IF(IDLV.GT.0) ITV=IVTV(IDLV)
      IF(ITV.EQ.0 .AND. IDLV.GT.0) IVTV(IDLV)=IDV
C
C    SET TRAILING VEH INITIALLY TO 0
      IVTV(IDV)=0
C     WRITE(6,345) IDV,LDX,IDLV
C345  FORMAT(' IN EXQUE:',3I4)
      IF( IDLV.EQ.0 ) GO TO 400
C
C    REASSIGN TRAILING VEH TO FOLLOW EGRESS VEH
 350  IDTV=IVTV(IDLV)
      IF( IDTV.EQ.0 .OR. IDTV.EQ.IDV ) GO TO 400
      LNTYT=LNTY(IVLS(IDTV))
      IF((LNTYT.NE.5 .AND. LNTYT.NE.2 ) .AND. IVCM(IDTV).NE.3)
     +  IVLV(IDTV)=IDV
      IF( LNTYT.NE.5 .AND. LNTYT.NE.2 ) IVTV(IDV)=IDTV
      IF( LNTYT.NE.5 .AND. LNTYT.NE.2 ) IVTV(IDLV)=IDV
      IF( IVLS(IDTV).EQ.LBP ) IVLV2(IDTV)=IDLV
      GO TO 400
C
C   REMOVE VEHICLE FROM DISCRETE MODE
C
  400 IVCM(IDV) = 1
      IVSS(IDV) = 0
      IVSB(IDV) = 0
      IVLS(IDV) = ISTX(IDS)
C
C   CLEAR EXIT QUEUE STATUS
C
```

```
      ISQS(IDS,3,IBN) = 0
      ISTA(IDS,6) = ISTA(IDS,6)-1
C
C   CHANGE EGRESS LINK CHARACTERISTICS
      LNK = IVLS(IDV)
      LNLV(LNK) = IDV
      LNTV(LNK) = LNTV(LNK)+1
      IF( LNFV(LNK).EQ.0 )   LNFV(LNK)=IDV
C
C   RESET VEHICLE STATES BEFORE ENTERING MAINLINE
      VACC(IDV) = 0.0
      VVEL(IDV) = 0.0
      VPOS(IDV) = 0.0
C
C   COMPUTE VEHICLE TIME IN STATION.
      TMST = T-TAST(IDV)
C
C   INCREMENT APPROPIATE BIN IN ARRAY, ITTS.
      DO 440 I =1,9
 440  IF( TMST.LT.(THST*I) ) GO TO 450
      I = 10
 450  ITTS(IDS,I) = ITTS(IDS,I)+1
C
C
C     IF THERE IS A TRAILING VEHICLE, ATTEMPT TO SHIFT
C   FORWARD.  OTHERWISE, BRANCH TO 1200 TO CHECK FOR POSSIBLE
C   DEQUEUING OF DOCKED VEHICLES.
C
      KNTR = 0
  500 IF(ISTA(IDS,6).EQ.KNTR)  GO TO 1200
      LBN = ISTA(IDS,3)
      DO 510 I=IBN,LBN
  510 IF(ISQS(IDS,3,I).NE.0)  GO TO 520
  520 IBN = I
      IDV = ISQS(IDS,3,IBN)
      IF(IDV.LT.0)  IDV=-IDV
C
C     SHIFT ENQUEUED VEHICLE FORWARD IN QUEUE IF POSSIBLE.
C
C        DETERMINE QUEUE POSITION TO WHICH VEHICLE IDV CAN BE SHIFTED.
C
 1000 IS = IBN-1
      IF( ISQS(IDS,3,IS).NE.0 )   RETURN
      DO 1010 I=1,IS
      LOC = IBN-I
 1010 IF( ISQS(IDS,3,LOC).NE.0 )   GO TO 1020
      IVSB(IDV) = 1
      GO TO 1030
 1020 IVSB(IDV) = LOC+1
 1030 ISQS(IDS,3,IVSB(IDV)) = -IDV
      ISQS(IDS,3,IBN) = 0
C
C     CALL RESCHD TO INITIATE SHIFT TO POSITION IVSB(IDV)
C
      CALL RESCHD(IDV,IDS,3,IBN,3,IVSB(IDV),IVLV(IDV))
      KNTR = KNTR+1
      GO TO 500
C
C     CHECK FOR POSSIBLE DEQUEUING OF VEHICLES FROM DOCK QUEUE
C   TO EXIT QUEUE.
```

```
C
 1200 IF( ISTA(IDS,5).EQ.0 )   RETURN
C
C     IS LAST EXIT QUEUE POSITION OPEN?   IF NOT, RETURN.
C
      IF( ISQS(IDS,3,ISTA(IDS,3)).NE.0 )   RETURN
C
C     HAS FIRST VEHICLE IN DOCK QUEUE COMPLETED ITS DWELL?  IF
C   NOT, RETURN.
C
      IH = ISTA(IDS,2)
      DO 1210 I=1,IH
 1210 IF( ISQS(IDS,2,I).NE.0 )   GO TO 1220
C
C     DETERMINE ID OF VEHICLE
C
 1220 IDV = ISQS(IDS,2,I)
      IF(IDV.LT.0) IDV = -IDV
      IF(IVLV(IDV).GE.0)  RETURN
C
C     CALL DKQUE TO SHIFT VEHICLE IDV INTO EXIT QUEUE.
C
      CALL DKQUE(IDV)
      RETURN
      END
```

```
      SUBROUTINE GSAVE(NVEH)
C
C     PURPOSE: TO SAVE SIMULATION VARIABLES FOR RESTART.
C
C     INPUT: NVEH = NUMBER OF VEHICLES IN SYSTEM.
C
C     LAST UPDATED: 24-AUG-82 BY HYC --9/3 BY PJM
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *      IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *      IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKVR/VACC(200),VVEL(200),VPOS(200),VENR(200)
      COMMON /BLKLI/LNTY(90),LNFV(90),LNTV(90),LNEL(90),LNLI(90),
     *      LNRN(90),LNLV(90)
      COMMON /BLKLR/ALMV(90),ALLL(90)
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      COMMON /BLKTI/ITPS(200),ITOS(200),ITDS(200),ITVI(200),
     *       ITSB(200),ITRD(200),ITUB(200),ITQC(200),ITTD(200)
      COMMON /BLKTR/TRAT(200),TRPT(200),TRBT(200),TRCT(200)
      COMMON /BLKEVI/IETY(500),IEID(500),IEPT(500),IEES(500)
      COMMON /BLKEVR/TEVT(500)
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
      COMMON /BLKLV2/LSTN(10)
      COMMON /BLKSCH/IELAST,IEMAX,IEHEAD,IETAIL
      COMMON /BLKVD/VDT,NVDS(200),IVDSP(20),NDSP,NFLT,IDSP
      COMMON /BLKRTA/RTFRQ(80),IRTSTC(80,20),IRTSCH(80,3),NSR,NER,
     *         NSRT(80),NSEQ(80,15)
      COMMON /BLKENR/WPAX,WV,WROT,DRGCF,ARR,BRR,AF,BF,G,EGB,EM,
     *        EPCU,RCPTY
      COMMON /BLKLEN/PENGRY(90),PPWR
      COMMON /BLKXTR/ VDIST(200),JFLGE,ISRN(20)
      COMMON /BLKTRH/ THOD,THST,TAST(200),ITOD(20,20,10),ITTS(20,10)
      COMMON /BLKSTA/ MSTAR(20),LFLOW(90)
      COMMON /BLKRSV/IVRSV(25,40),PTT(25,25),ATT(25,25),IVTHRD(25),
     *   MIDL(5),MIML(5),WSIZE,IRP,JFLGTM
      COMMON /BLKENG/ PENGYD(90),DELPWR(200),POWER,POWERD,SWAUX,
     *   JFLGED
C
C
C     SAVE EVENT VARIABLES: BLKEVI,BLKEVR,BLKSCH
      NEV=500
      WRITE(61) T,(IETY(I),I=1,NEV),(IEID(I),I=1,NEV),
     +            (IEPT(I),I=1,NEV),(IEES(I),I=1,NEV),
     +            (TEVT(I),I=1,NEV),IELAST,IEMAX,IEHEAD,IETAIL
C
C     SAVE VEHICLE VARIABLES: BLKVI,BLKVR,BLKXTR
      NV=200
      WRITE(61) NVEH,(IVCM(I),I=1,NV),(IVSR(I),I=1,NV),(IVND(I),I=1,NV)
     +          ,(IVLV(I),I=1,NV),(IVLV2(I),I=1,NV),(IVTV(I),I=1,NV),
     +           (IVSS(I),I=1,NV),(IVLS(I),I=1,NV),(IVTP(I),I=1,NV),
     +           (IVRF(I),I=1,NV),(IVPX(I),I=1,NV),
     +           ((IVCS(I,J),I=1,NV),J=1,20),(IVQCH(I),I=1,NV),
     +           (IVSB(I),I=1,NV),(VACC(I),I=1,NV),(VVEL(I),I=1,NV),
     +           (VPOS(I),I=1,NV),(VENR(I),I=1,NV),(VDIST(I),I=1,NV)
C
C     SAVE LINK VARIABLES: BLKLI,BLKLR,BLKLV2
      NL=90
      WRITE(61) (LNTY(I),I=1,NL),(LNFV(I),I=1,NL),(LNTV(I),I=1,NL),
     +           (LNEL(I),I=1,NL),(LNLI(I),I=1,NL),(LNRN(I),I=1,NL),
     +           (LNLV(I),I=1,NL),(ALMV(I),I=1,NL),(ALLL(I),I=1,NL),
```

```
     +          (LSTN(I),I=1,10)
C
C    SAVE STATION VARIABLES: BLKSI,BLKXTR
      NS=20
      WRITE(61) ((ISTA(I,J),I=1,NS),J=1,10),
     +          ((ISTL(I,J),I=1,NS),J=1,10),
     +          (((ISQS(I,J,K),I=1,NS),J=1,3),K=1,10),
     +          (ISTX(I),I=1,NS),(ISRN(I),I=1,NS)
C
C    SAVE DISPATCH VARIABLES: BLKVD,BLKRTA
      NR=80
      WRITE(61) VDT,(NVDS(I),I=1,NV),(IVDSP(I),I=1,NS),NDSP,NFLT,IDSP,
     +          (RTFRQ(I),I=1,NR),
     +          ((IRTSTC(I,J),I=1,NR),J=1,NS),
     +          ((IRTSCH(I,J),I=1,NR),J=1,3),NSR,NER,
     +          (NSRT(I),I=1,NR),
     +          ((NSEQ(I,J),I=1,NR),J=1,15)
C
C    SAVE ENERGY VARIABLES: BLKENR,BLKLEN,BLKENG
      WRITE(61) WPAX,WV,WROT,DRGCF,ARR,BRR,AF,BF,G,EGB,EM,EPCU,RCPTY,
     +          (PENGRY(I),I=1,NL),(PENGYD(J),J=1,NL) , POWER,POWERD,
     +          (DELPWR(K),K=1,NVEH),SWAUX,JFLGED
C
C    SAVE TRIP VARIABLES: BLKTI,BLKTR,TRHIST,BLKSTA
      NT=200
      WRITE(61) (ITPS(I),I=1,NT),(ITOS(I),I=1,NT),(ITDS(I),I=1,200),
     +          (ITVI(I),I=1,NT),(ITSB(I),I=1,NT),(ITRD(I),I=1,200),
     +          (ITUB(I),I=1,NT),(ITQC(I),I=1,NT),(ITTD(I),I=1,200),
     +          (TRAT(I),I=1,NT),(TRPT(I),I=1,NT),(TRBT(I),I=1,200),
     +          (TRCT(I),I=1,NT),
     +          THOD,THST,(TAST(I),I=1,NT),
     +          (((ITOD(I,J,K),I=1,20),J=1,20),K=1,20),
     +          ((ITTS(I,J),I=1,20),J=1,10),
     +          (MSTAR(I),I=1,20)
C
C        SAVE RESERVATION MATRIX DATA
      WRITE(61)
     +          IRP,WSIZE,
     +          (MIDL(I),I=1,5),(MIML(I),I=1,5),
     +          ((IVRSV(I,J),I=1,25),J=1,40),
     +          (IVTHRD(I),I=1,25),
     +          (IVQCH(I),I=1,200)
C
C    THE FOLLOWING COMMON BLOCKS WERE NOT SAVED: BLKTI,BLKTR,BLKTIM,
C        BLKCN,BLKLUP,BLKSNI,BLKSNR
C
 100  RETURN
      END
```

```
      SUBROUTINE LEADV(ID,LNEW,IDL)
C
C    PURPOSE: TO FIND A LEAD VEHICLE FOR A VEHICLE ENTERING A NEW LINK
C
C    INPUTS: ID=VEHICLE ID
C            LNEW=NEW LINK ID
C
C    OUTPUTS: IDL=NEW LEAD VEHICLE ID
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *      IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *      IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKLI/LNTY(90),LNFV(90),LNTV(90),LNEL(90),LNLI(90),
     *      LNRN(90),LNLV(90)
      COMMON /BLKLUP/DISTL(90,90),NPNODE(90),IEXIT(90,90)
C
C    START. IF LAST VEH ON NEW LINK IS NOT 0, ASSIGN AS LEAD VEH
      IDL=0
      LOLD=LNEW
      IF( LNLV(LNEW).EQ.0 ) GO TO 100
      IDL=LNLV(LNEW)
      RETURN
C
C    FIND NEXT LINK AFTER LNEW
 100  LNEW=IEXIT(LNEW,IVND(ID))
      IF( LNLV(LNEW).EQ.0 ) RETURN
      IDL=LNLV(LNEW)
      RETURN
      END
```

```
      SUBROUTINE LEADV2(ID,LINK,IDL,IDL2)
C
C    PURPOSE: TO FIND LEAD VEH(S) FOR PARALLEL LINK VEHICLES
C
C    INPUTS: ID=VEH ID
C            LINK=LINK ID
C
C    OUTPUTS: IDL=PRIMARY LEAD VEH, I.E., TO FOLLOW AFTER PARALLEL LINK
C              IDL2=SECONDARY LEAD VEH
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *      IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *      IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKLI/LNTY(90),LNFV(90),LNTV(90),LNEL(90),LNLI(90),
     *      LNRN(90),LNLV(90)
      COMMON /BLKLUP/DISTL(90,90),NPNODE(90),IEXIT(90,90)
      COMMON /BLKLV2/LSTN(10)
C
C    START. IF LAST VEH TO ENTER THE PARALLEL REGION IS NOT 0, GO TO
C    "IF IDL IS NOT..."
      IDL=LSTN( NPNODE(LINK) )
      IF( IDL.NE.0 ) GO TO 50
C
C    FIND NEXT LINK AND FOLLOW LAST VEHICLE ON LINK
      LNEW=IEXIT(LINK,IVND(I))
      IDL=LNLV(LNEW)
      IDL2=0
      GO TO 100
C
C    IF IDL IS NOT ON SAME LINK AS ID, IDL2 IS LEAD VEH; ELSE IDL2=0
   50 IDL2=IVLV(ID)
      IF( LINK.EQ.IVLS(IDL) ) IDL2=0
      IF( IDL2.EQ.0 ) GO TO 100
C
C    IF 2ND LEAD VEH IS NOT ON EITHER PARALLEL LINK, IDL2=0
      LINK2=IVLS(IDL2)
      IF( LINK2.NE.LINK .AND. LINK2.NE.LNLI(LINK) ) IDL2=0
C
C    RESET LAST VEH IN PARALLEL REGION TO ID
  100 LSTN( NPNODE(LINK) )=ID
      RETURN
      END
```

```
      SUBROUTINE LINKUP(IDV,LOLD,LNEW)
C
C   PURPOSE: TO UPDATE LINK CHARACTERISTICS AFTER VEHICLE
C            LINK CROSSINGS
C
C   INPUTS: IDV = VEHICLE ID
C           LOLD = OLD LINK ID
C           LNEW = NEW LINK ID
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *      IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *      IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKLI/LNTY(90),LNFV(90),LNTV(90),LNEL(90),LNLI(90),
     *      LNRN(90),LNLV(90)
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
C
C    UPDATE LOLD LINK.
      IF( LNTV(LOLD).GT.1 ) GO TO 100
C
C    NO VEHICLES ON LOLD.
      LNLV(LOLD)=0
      LNFV(LOLD)=0
      LNTV(LOLD)=0
      GO TO 400
C
C    IF TRAILING VEHICLE IS NOT ON LOLD OR A COMPANION LINK, SCHEDULE
C    ERROR EVENT; ELSE UPDATE LOLD.
 100  IDTV=IVTV(IDV)
      ICOUNT=0
 110  LTRV=IVLS(IDTV)
      LTRTY=LNTY(LTRV)
      IF( LTRV.EQ.LOLD ) GO TO 300
      IF( LTRTY.NE.2 .AND. LTRTY.NE.5 .AND. LTRTY.NE.6 ) GO TO 200
      IF( LTRV.NE.LNLI(LOLD) ) GO TO 200
      IDTV=IVTV(IDTV)
      ICOUNT=ICOUNT+1
      IF( ICOUNT.GT.10 ) GO TO 230
      IF( IDTV.EQ.0 ) GO TO 220
      GO TO 110
C
C    IMPROPER TRAILING VEHICLE. SCHEDULE ERROR EVENT WITH ERROR MESSAGE
 200  CALL SCHED(T,2,10)
      GO TO 290
 220  CALL SCHED(T,3,10)
      GO TO 290
 230  CALL SCHED(T,4,10)
 290  WRITE(6,292) IDV,LOLD,LNEW,IDTV
 292  FORMAT(/,' LINKUP ERROR: IDV=',I3,';LOLD=',I3,';LNEW=',I3,
     +          ';IDTV=',I3/)
      RETURN
C
C    UPDATING LOLD.
 300  LNFV(LOLD)=IDTV
      LNTV(LOLD)=LNTV(LOLD)-1
C
C    UPDATE LNEW LINK.
 400  IF( LNEW.EQ.0 ) RETURN
      IF( LNTV(LNEW).EQ.0 ) GO TO 500
C
C    VEHICLES ALREADY ON LNEW.
```

```
      LNLV(LNEW)=IDV
      LNTV(LNEW)=LNTV(LNEW)+1
      RETURN
C
C     NO VEHICLES ON LNEW.
 500  LNLV(LNEW)=IDV
      LNFV(LNEW)=IDV
      LNTV(LNEW)=1
      RETURN
      END
```

```
      SUBROUTINE NEWLV(I,LNW,LOLD,IVLO)
C     PURPOSE:  TO DETERMINE NEW LEAD VEHICLE ASSIGNMENTSFOR
C   A VEHICLE I ENTERING A NEW LINK LNW:  TO ADJUST TRAILING VEHICLE
C   ASSIGNMENTS FOR I'S NEW AND OLD LEAD VEHICLES;  AND TO ADJUST LEAD
C   VEHICLE ASSIGNMENT OF I'S TRAILING VEHICLE IF I IS DIVERTING INTO
C   A STATION.
C
C     CALLED BY:  CONSUB
C
C     CALLS:  LEADV, LEADV2, ENTRY, REJECT, REMOVE, RJRMV, LINKUP
C
C     OUTPUT:  UPDATED IVTV,IVLV
C
C     INPUT:  I'S OLD LINK--LOLD
C             I'S NEW LINK--LNW
C             I'S OLD LEAD VEHICLE
C
C   LAST CHANGE:  09-07-82  BY DLK
C
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *IVCS(200),IVQCH(200),IVSB(200)
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
      COMMON /BLKLI/LNTY(90),LNFV(90),LNTV(90),LNEL(90),LNLI(90),
     *LNRN(90),LNLV(90)
      COMMON /BLKXTR/VDIST(200),JFLGE,ISRN(20)
      COMMON /BLKRSV/IVRSV(25,40),PTT(25,25),ATT(25,25),IVTHRD(25),
     *   MIDL(5),MIML(5),WSIZE,IRP,JFLGTM
C
C     DETERMINE LINK TYPE FOR I'S NEW LINK,LNW.
      LNKTY = LNTY(LNW)
C
C         ESTABLISH CASE:
C            CASE1--LNW IS A NOMINAL, DIVERT, OR STATION ENTRY LINK
C
C            CASE 2--LNW IS A PARALLEL DATA REGION LINK
C
C            CASE 3--LNW IS A STATION BYPASS LINK
C
      GO TO(100,200,100,300,1000,100),LNKTY
C
C   CASE 1 LOGIC
C                NEW LINK TYPE:  NOMINAL,DIVERT,STATION ENTRY
C
  100 LNEW = LNW
      CALL LEADV(I,LNEW,IL)
      IVLV(I) = IL
      IVLV2(I) = 0
C
C   RESET TRAILING VEHICLE FOR I'S NEW LEAD VEHICLE, IL
      IF(IL.EQ.0) GO TO 120
      IF(IVTV(IL) .EQ. 0) GO TO 110
      LNKTYL = LNTY(IVLS(IL))
C
C   IF LEAD VEHICLE IL IS ON A PARALLEL OR STATION BYPASS LINK, DO
C   NOT CHANGE IL'S TRAILING VEHICLE IF IT IS ON IVLS(IL)'S
C   COMPANION LINK.
      IF((LNKTYL.NE.2) .AND. (LNKTYL.NE.4)) GO TO 110
      IF(IVLS(IVTV(IL)).EQ.LNLI(IVLS(IL))) GO TO 120
```

```
  110 IVTV(IL) = I
C
C   RESET TRAILING VEHICLE FOR IVLO (I'S OLD LEAD VEHICLE)
C
  120 IF((IVLO.EQ.0) .OR. (IVLO.EQ.IL)) GO TO 140
      IF(IVTV(IVLO).NE.I) GO TO 140
C
C IF IVTV(IVLO)=I, RESET TO ZERO
      IVTV(IVLO) = 0
C
C   NO NEED TO RESET I'S TRAILING VEHICLE
C
  140 RETURN
C
C   CASE 2 LOGIC
C             NEW LINK TYPE: PARALLEL
C
  200 LNEW = LNW
      CALL LEADV2(I,LNEW,IL,IL2)
      IVLV(I) = IL
      IVLV2(I) = IL2
C
C   RESET TRAILING VEHICLE
      IF(IL .NE. 0) IVTV(IL)=I
C
C   NO NEED TO CHANGE I'S TRAILING VEHICLE, IL'S TRAILING VEHICLE,
C   OR ILVO'S LEAD VEHICLE
      RETURN
C
C     CASE 3 LOGIC
C              NEW LINK TYPE: STATION BYPASS
C
C   FIND ID OF COMPANION LINK OF LNW (IE, THE STATION EGRESS LINK)
  300 LX = LNLI(LNW)
C
C   IF VEHICLE IS AT ITS STATION DESTINATION, GO TO "ENTER STATION"
      IF(LNLI(LOLD) .EQ. IVND(I)) GO TO 500
C
      IF(IVRF(I).NE.0) GO TO 510
C         (IE, IF VEHICLE I WAS REJECTED AT PREVIOUS DESTINATION, DIVERT
C                INTO THIS STATION)
C
C   OTHERWISE, FIND LEAD VEHICLE(S)
C
C   IF THE BYPASS LINK IS NOT EMPTY, GO TO "CHECK EGRESS'
  350 IF(LNTV(LNW) .NE. 0) GO TO 360
C
C   IF THERE ARE NO VEHICLES ON THE EGRESS LINK, GO TO "CHECK NEXT LINK"
      IF(LNTV(LX) .EQ. 0) GO TO 370
C
C   FOLLOW LAST VEHICLE ON EGRESS LINK
      IDL = LNLV(LX)
      IDL2 = 0
      GO TO 400
C
C   CHECK EGRESS.  IF THERE ARE NO VEHICLES ON THE EGRESS LINK, GO TO
C   "FOLLOW BYPASS VEHICLE"
  360 IF(LNTV(LX) .EQ. 0) GO TO 380
C
C   IF THE TRAILING VEHICLE OF THE LAST BYPASS VEHICLE IS NOT ON THE
```

```
C   EGRESS LINK, GO TO"FOLLOW BYPASS VEHICLE"
      IF(IVLS(IVTV(LNLV(LNW))) .NE. LX) GO TO 380
C
C   FOLLOW EGRESS VEHICLE
      IDL = IVTV(LNLV(LNW))
      IDL2 = LNLV(LNW)
      GO TO 400
C
C   CHECK NEXT LINK IF NEXT IS EMPTY, SET THE LEAD VEHICLE TO ZERO;
C   ELSE, FOLLOW THE LAST VEHICLE.
  370 LNW2 = LNEL(LNW)
      IDL = LNLV(LNW2)
      IDL2 = 0
      GO TO 400
C
C   FOLLOW BYPASS VEHICLE
  380 IDL = LNLV(LNW)
      IDL2 = 0
      GO TO 400
C
C   CHANGE LEAD AND TRAILING VEHICLE VARIABLES
  400 IVLV(I) = IDL
      IVLV2(I) = IDL2
      IF(IDL .NE. 0) IVTV(IDL)=I
      RETURN
C
C    ENTER STATION.  IF VEHICLE IS NOT REJECTED FROM STATION,
C   GO TO "BEGIN REMOVAL"
C
  500 CALL ENTRY(IVND(I),IVRF(I))
      IF(IVRF(I) .EQ. 0) GO TO 600
C
C   VEHICLE WAS REJECTED.  UPDATE STATION REJECTION COUNTER.
C   PRINT OUT REJECTION DATA, AND GO TO "FIND LEAD VEHICLE"
      ISRN(IVND(I)) = ISRN(IVND(I))+1
      CALL REJECT(I,T,IVND(I))
      GO TO 350
C
  510 CALL ENTRY(LNLI(LOLD),IVRF(I))
      IF(IVRF(I) .EQ. 0) GO TO 600
C
C  VEHICLE WAS REJECTED AGAIN.
      ISRN(LNLI(LOLD)) = ISRN(LNLI(LOLD))+1
      CALL REJECT(I,T,LNLI(LOLD))
      GO TO 350
C
C   BEGIN REMOVAL
C
C   RESET LEAD AND TRAILING VEHICLES.
C
C   IF TRAILING VEHICLE IS NOT IN LOLD, SET TRAILING VEHICLE'S LEAD
C   VEHICLE TO ZERO.
  600 IDT = IVTV(I)
      IF(IDT .EQ. 0) GO TO 700
      IF(IVLS(IDT) .EQ. LOLD) GO TO 700
      IF(IVCM(IDT).NE.3) IVLV(IDT)=0
  650 IDT = 0
C
C   RESET LEAD VEHICLE'S TRAILING VEHICLE.
C
```

```
  700 IDLL = IVLV(I)
      IF(IDLL .EQ. 0) GO TO 750
      IF(IVLS(IDLL) .NE. LNW) IDLL=0
      IF(IVTV(IVLV(I)).EQ.I) IVTV(IVLV(I)) = IDT
  750 IF((IDT .NE. 0) .AND. (IVCM(IDT) .NE. 3)) IVLV(IDT)=IDLL
C
C   REMOVE VEHICLE I FROM CONTINUOUS MODE.
      CALL LINKUP(I,LOLD,0)
      IVLV(I) = 0
      IVTV(I) = 0
C        SET IDEST=0 IF VEHICLE I WAS REJECTED AT UPSTREAM STATION
      IDEST = 1
      IF(IVND(I).NE.LNLI(LOLD)) IDEST=0
      IF(IDEST.EQ.1)  GO TO 800
      IF(IDEST.EQ.0)  CALL RJRMV(I,LNLI(LOLD),T)
      GO TO 810
C
 800  CALL REMOVE(I,T)
 810  LNW = 0
      RETURN
C
C    CALL ERROR IF LNW IS AN EGRESS LINK
 1000 CALL SCHED(T,1,10)
      RETURN
      END
```

```
      SUBROUTINE PROUT(TX,NVEH)
C
C   PURPOSE: TO GENERATE HARDCOPY OF VARIOUS PERFORMANCE MEASURES
C
C   INPUT: TX = THE NEXT TIME PROUT IS TO BE SCHEDULED
C          NVEH = THE NUMBER OF VEHICLES IN THE SYSTEM
C
C   LAST UPDATE: 14-SEP-82 BY PJM  02:15 PM
C
C
       COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *      IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *      IVCS(200,20),IVQCH(200),IVSB(200)
       COMMON /BLKVR/VACC(200),VVEL(200),VPOS(200),VENR(200)
       COMMON /BLKLI/LNTY(90),LNFV(90),LNTV(90),LNEL(90),LNLI(90),
     *      LNRN(90),LNLV(90)
       COMMON /BLKLR/ALMV(90),ALLL(90)
       COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
       COMMON /BLKCN/HD,GK,DELTA,AS,XJS,CZ1,CZ2,CZ3,VLEN
       COMMON /BLKLUP/DISTL(90,90),NPNODE(90),IEXIT(90,90)
       COMMON /BLKRTA/RTFRQ(80),IRTSTC(80,20),IRTSCH(80,3),NSR,NER,
     *          NSRT(80),NSEQ(80,15)
       COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
       COMMON /BLKLEN/PENGRY(90),PPWR
       COMMON /BLKXTR/ VDIST(200),JFLGE,ISRN(20)
       COMMON /BLKSA/ IRTSAT(80,10),IRTSAI(80,10)
       COMMON /BLKTRH/ THOD,THST,TAST(200),ITOD(20,20,10),ITTS(20,10)
       COMMON /BLKSTA/ MSTAR(20),LFLOW(90)
       COMMON /BLKENG/ PENGYD(90),DELPWR(200),POWER,POWERD,
     +            SWAUX,JFLGED
C
       DIMENSION LHEAD(90,10),ITOT(10)
       DIMENSION DENS(90),VSPEED(90)
C
C   BEGIN. PRINTOUT OF VEHICLE DATA.
      WRITE(6,10) T
 10   FORMAT('1PRINTOUT: TIME = ',F7.2,/)
      WRITE(6,20)
 20   FORMAT(' *** VEHICLE DATA ***',/)
      PVF=0.
      PVC=0.
      PST=0.
      DO 30 I = 1,NVEH
      IMODE=IVCM(I)
      GO TO (22,24,26), IMODE
      WRITE(6,15) I,IMODE
 15   FORMAT(' IN PRINTOUT: VEHICLE ',I3,' HAS A BAD COMMAND MODE OF',
     +        I3,'!',/)
      GO TO 30
 22   PVF=PVF+1.
      GO TO 30
 24   PVC=PVC+1.
      GO TO 30
 26   PST=PST+1.
 30   CONTINUE
      NVF=PVF
      NVC=PVC
      NST=PST
      PVF=PVF/NVEH*100.
      PVC=PVC/NVEH*100.
```

```
      PST=PST/NVEH*100.
      WRITE(6,40) NVEH,NVF,PVF,NVC,PVC,NST,PST
 40   FORMAT(' TOTAL VEHICLES = ',I4,/,
     +       '   IN VEHICLE-FOLLOWER MODE: ',I3,' VEH;',F7.2,'%',/,
     +       '   IN VELOCITY-COMMAND MODE: ',I3,' VEH;',F7.2,'%',/,
     +       '           IN STATION BERTHS: ',I3,' VEH;',F7.2,'%',/)
      DKM=0.
      DO 41 I=1,NVEH
 41   DKM=DKM+VDIST(I)
      DKM=DKM/1000.
      DMI=DKM*.6214
      WRITE(6,43) DKM,DMI
 43   FORMAT(' TOTAL VEHICLE DISTANCE = ',F12.2,' KM  (',F12.2,' MI)')
      CUMDST = VDIST(199)
      CUMEGY = VDIST(200)
      DLDSTK = DKM-CUMDST
      DLDSTM = DLDSTK*.6214
      CUMDST = DKM
      AVM = ((DLDSTM/NVEH)*3600.0)/(TX-T)
      WRITE(6,44) DLDSTK,DLDSTM,AVM
   44 FORMAT(/' TOT. VEH. DIST. THIS REPORT INTERVAL=',F10.2,' KM ('
     +,F10.2,'MI)',/,' AVERAGE SPEED THIS REPORT INTERVAL=',F10.2,'M/H',
     +'   (TRIP SPEED--INCLUDES STATION STOP DELAY)')
C
C  STATION DATA (ON SAME PAGE AS VEHICLE DATA).
      WRITE(6,42)
 42   FORMAT(/,' *** STATION DATA ***',/)
      ITOTL=0
      MTOTL=0
      DO 45 I = 1,20
      MTOTL=MTOTL+MSTAR(I)
 45   ITOTL=ITOTL+ISRN(I)
      WRITE(6,47) ITOTL,MTOTL
 47   FORMAT(' REJECTIONS: TOTAL NUMBER = ',I4,
     +       ' OUT OF ',I7,' TOTAL STATION ARRIVALS.')
      WRITE(6,48) (ISRN(J),J=1,20),(MSTAR(I),I=1,20)
 48   FORMAT(/,'  REJECTIONS BY STATION:',20I5,/,
     +       /,'  TOTAL ATTEMPTS BY STATION:',20I5,/)
      WRITE(6,46)
 46   FORMAT(//'   STATION QUEUE STATUS',//,
     +'     |     ENTRANCE QUEUE     |       DOCK QUEUE        |        ',
     +'EXIT QUEUE        |',/)
      DO 49 I=1,20
 49   WRITE(6,51) I,((ISQS(I,J,6-K),K=1,5),J=1,3)
 51   FORMAT(I5,'|',15(I4,'|'))
C
C  LINK DATA (NEW PAGE).
      WRITE(6,10) T
      WRITE(6,50)
 50   FORMAT(' *** LINK DATA ***',//,132('='),/,
     +T2,' LINK',T9,'NUMBER',T16,'FLOW',T23,'DENSITY',T31,'SPEED',
     +T38,'ENERGY',T46,'ENERGY-D',T54,'*',
     +T60,' LINK',T67,'NUMBER',T74,'FLOW',T81,'DENSITY',T89,'SPEED',
     +T96,'ENERGY',T103,'ENERGY-D',/,
     +T2,' ID',T9,'OF VEH',T16,'(VEH/INT)',T26,'(VEH/M)',T33,'(M/S)',
     +T39,'(KWH)',T47,'(KWH)',T54,'*',
     +T60,' ID',T67,'OF VEH',T74,'(VEH/INT)',T81,'(VEH/M)',
     +T88,'(M/S)',T95,'(KWH)',T103,'(KWH)',/,132('='))
C
C  COMPUTE LINK SPEEDS AND DENSITY
```

```
      DO 58 I=1,90
      VSPEED(I)=0.0
      LNUM=LNTV(I)
      ANUM=LNUM
      DENS(I)=ANUM/ALLL(I)
      IF( LNUM.EQ.0 ) VSPEED(I)=ALMV(I)
      IF( LNUM.EQ.0 ) GO TO 58
      NID=LNFV(I)
      DO 54 J=1,LNUM
      VSPEED(I)=VSPEED(I)+VVEL(NID)
      IF( NID.EQ.0 ) GO TO 56
 54   NID=IVTV(NID)
 56   VSPEED(I)=VSPEED(I)/ANUM
 58   CONTINUE
C
C  BEGIN PRINTOUT
      DO 70 I = 1,45
      J=I+45
 70   WRITE(6,80) I,LNTV(I),LFLOW(I),DENS(I),VSPEED(I),PENGRY(I),
     +  PENGYD(I),J,LNTV(J),LFLOW(J),DENS(J),VSPEED(J),PENGRY(J),
     +  PENGYD(J)
 80   FORMAT(T2,I4,T9,I4,T16,I4,T23,F7.4,T30,F7.2,T37,F7.3,T47,F7.3,
     + T58,'*',
     + T60,I4,T67,I4,T74,I4,T81,F7.4,T88,F7.2,T95,F7.4,T105,F7.4)
      ETOT=0.0
      ETOTD=0.0
      DO 84 I=1,90
      ETOTD=ETOTD+PENGYD(I)
 84   ETOT=ETOT+PENGRY(I)
      WRITE(6,86) ETOT,ETOTD
 86   FORMAT(/,' TOT NETWORK ENERGY = ',F14.4,' KWH',/,1X,'TOTAL DIRECT
     +NETWORK ENERGY =',F14.4,'KWH')
      DLENGY = ETOT-CUMEGY
      CUMEGY = ETOT
      IF(DLDSTM.EQ.0.0) GO TO 89
      EPVMT = DLENGY/DLDSTM
      WRITE(6,88) DLENGY,EPVMT
   88 FORMAT(/' ENERGY CONSUMPTION THIS REPORT INTERVAL=',F10.2,
     +  ' KWH',/,' ENERGY PER VMT THIS REPORT INTERVAL=',F10.2,' KWH')
   89 VDIST(199) = CUMDST
      VDIST(200) = CUMEGY
C
C  RESET LINK FLOW VECTOR
      DO 85 I=1,90
 85   LFLOW(I)=0
C
C  ROUTE DATA (NEW PAGE)
      WRITE(6,10) T
      IBIN=30
      WRITE(6,90) IBIN
 90   FORMAT(/' *** ROUTE DATA ***'//,
     + ' ROUTE ADHERENCE - INITIATION (BIN SIZE = ',I4,' S)'//,
     + ' RT#|  1 |  2 |  3 |  4 |  5 |  6 |  7 |  8 |  9 | 10 |##|',
     + ' RT#|  1 |  2 |  3 |  4 |  5 |  6 |  7 |  8 |  9 | 10 |',/,
     + 113('='))
      DO 91 I=1,10
 91   ITOT(I) = 0
      DO 92 I=1,40
      I2=I+40
 92   WRITE(6,94) I,(IRTSAI(I,J),J=1,10),I2,(IRTSAI(I2,J),J=1,10)
```

```
      DO 93 I=1,80
       DO 93 J=1,10
 93    ITOT(J) = ITOT(J)+IRTSAI(I,J)
 94   FORMAT(11(I4,'|'),'##|',11(I4,'|'))
      WRITE(6,100) (ITOT(I),I=1,10)
      WRITE(6,10) T
      WRITE(6,95) IBIN
 95   FORMAT(/' *** ROUTE DATA ***'//,
     + ' ROUTE ADHERENCE - TERMINATION (BIN SIZE = ',I4,' S)'//,
     + ' RT#|  1 |  2 |  3 |  4 |  5 |  6 |  7 |  8 |  9 | 10 |##|',
     + ' RT#|  1 |  2 |  3 |  4 |  5 |  6 |  7 |  8 |  9 | 10 |',/,
     + 113('='))
      DO 96 I=1,40
      I2=I+40
 96   WRITE(6,94) I,(IRTSAT(I,J),J=1,10),I2,(IRTSAT(I2,J),J=1,10)
      DO 97 I=1,10
 97   ITOT(I) = 0
      DO 98 I=1,80
       DO 98 J=1,10
 98    ITOT(J) = ITOT(J)+IRTSAT(I,J)
      WRITE(6,100) (ITOT(I),I=1,10)
 100  FORMAT(//,' TOT|',10(I4,'|'))
C
C  HEADWAY DATA
C
C  ZERO LHEAD ARRAY
      DO 110 I=1,90
      DO 110 J=1,10
 110  LHEAD(I,J)=0
C
C  BEGIN HEADWAY LOOP
      DO 160 I=1,NVEH
      IF( IVCM(I).EQ.1 ) GO TO 115
      IF( IVCM(I).EQ.3 ) GO TO 160
      GO TO 117
 115  ILINK=IVLS(I)
      LHEAD(ILINK,1)=LHEAD(ILINK,1)+1
      GO TO 160
C
C  COMPUTE SPACING
 117  IL=IVLV(I)
      IF( IL.EQ.0 ) GO TO 140
      LNKTY=LNTY(IVLS(I))
      GO TO (120,130,120,130,130,120),LNKTY
 120  SS=VPOS(IL)-VPOS(I)+DISTL(IVLS(I),IVLS(IL))
      GO TO 145
 130  IF( IVLS(IL).NE.LNLI(IVLS(I)) ) GO TO 120
      SS=ALLL(IVLS(I))-ALLL(IVLS(IL))+VPOS(IL)-VPOS(I)
      GO TO 145
 140  SS=9999999.
 145  CONTINUE
C
C  COMPUTE HEADWAY TO LEADV
      IF( VVEL(I).LE.0.0001 ) GO TO 153
      HV=SS/VVEL(I)
C
C  PUT HV INTO PROPER BIN INTERVAL BY LINK
      ILINK=IVLS(I)
      DO 150 J=1,9
 150  IF( HV.LE.(HD*J) ) GO TO 155
```

```
 153   J=10
 155   LHEAD(ILINK,J)=LHEAD(ILINK,J)+1
 160   CONTINUE
C
C  PRINTOUT HEADWAY DATA

       WRITE(6,10) T
       IBIN=HD
       WRITE(6,170) IBIN
 170   FORMAT(/' *** HEADWAY DATA ***'//,
     + ' HEADWAY DISTRIBUTION BY LINK (BIN SIZE = ',I4,' S)'//,
     + ' LNK|  1 |  2 |  3 |  4 |  5 |  6 |  7 |  8 |  9 | 10 |##|',
     + ' LNK|  1 |  2 |  3 |  4 |  5 |  6 |  7 |  8 |  9 | 10 |',/,
     + 113('='))
       DO 175 I=1,45
       I2=I+45
 175   WRITE(6,180) I,(LHEAD(I,J),J=1,10),I2,(LHEAD(I2,J),J=1,10)
 180   FORMAT(11(I4,'|'),'##|',11(I4,'|'))
       DO 190 I=1,10
 190    ITOT(I) = 0
       DO 200 I=1,90
        DO 200 J=1,10
 200    ITOT(J) = ITOT(J)+LHEAD(I,J)
       WRITE(6,100) (ITOT(I),I=1,10)
C
C  PRINTOUT STATION TIME DATA.
       WRITE(6,10) T
       WRITE(6,300) THST
 300   FORMAT(/' *** STATION TIME DATA *** '//,
     + ' TIME DISTRIBUTION BY STATION (BIN SIZE = ',F7.3,' S)'//,
     + ' ST#|  1 |  2 |  3 |  4 |  5 |  6 |  7 |  8 |  9 | 10 |',/,
     + 55('='))
       DO 310 I=1,20
 310   WRITE(6,320) I,(ITTS(I,J),J=1,10)
 320   FORMAT(11(I4,'|'))
       DO 330 I=1,10
 330   ITOT(I) = 0
       DO 340 I=1,20
       DO 340 J=1,10
 340   ITOT(J) = ITOT(J)+ITTS(I,J)
       WRITE(6,100) (ITOT(I),I=1,10)
C
C  PRINTOUT O/D TRIP TIME DATA.
C  BEGIN PRINTING 5 PAGES OF DATA (4 STATIONS/PAGE).
       DO 435 I=4,20,4
       WRITE(6,10) T
       WRITE(6,400) THOD
 400   FORMAT(' *** O/D TRIP TIME DATA ***'/,
     + ' TIME DISTRIBUTION BY STATION (BIN SIZE = ',F7.3,' S)'/)
       IM1 = I-1
       IM2 = I-2
       IM3 = I-3
       WRITE(6,410) IM3,IM2
 410   FORMAT(' #',I2,
     + '|  1 |  2 |  3 |  4 |  5 |  6 |  7 |  8 |  9 | 10 |',
     + '*****  #',I2,
     + '|  1 |  2 |  3 |  4 |  5 |  6 |  7 |  8 |  9 | 10 |',/,
     + 1X,54('='),5('*'),55('='))
       DO 420 M=1,20
 420   WRITE(6,425) M,(ITOD(IM3,M,J),J=1,10),M,(ITOD(IM2,M,J),J=1,10)
```

```
 425  FORMAT(11(I4,'|'),5('*'),11(I4,'|'))
      WRITE(6,427)
 427  FORMAT(//)
      WRITE(6,410) IM1,I
      DO 430 M=1,20
 430  WRITE(6,425) M,(ITOD(IM1,M,J),J=1,10),M,(ITOD(I,M,J),J=1,10)
 435  CONTINUE
C
C  PRINT SUM OF O/D TIME BINS
      DO 440 I=1,10
 440  ITOT(I)=0
      DO 450 I=1,10
      DO 450 J=1,20
      DO 450 K=1,20
 450  ITOT(I)=ITOT(I)+ITOD(J,K,I)
      WRITE(6,100) (ITOT(I),I=1,10)
C
C  PRINT OUT DETAILED ENERGY STATISTICS  IF JFLGED=1
      IF( JFLGED.EQ.0 ) GO TO 645
      WRITE(6,10)T
      WRITE(6,636)
      WRITE(6,637)
 636  FORMAT(/,1X,4('VEH         VEH         VEH         '))
 637  FORMAT(1X,4('ID        ENERGY        POWERD      '),//)
      DO 639 I=1,197,4
      I1=I+1
      I2=I+2
      I3=I+3
      WRITE(6,638) I,VENR(I),DELPWR(I),I1,VENR(I1),DELPWR(I1),I2,
     + VENR(I2),DELPWR(I2),I3,VENR(I3),DELPWR(I3)
 638  FORMAT(1X,4(I3,1X,2E12.4,'  *  '))
      IF( I3.GE.NVEH ) GO TO 640
 639  CONTINUE
 640  DELPEL=0.0
 641  DO 642 I =1,NVEH
      DELPEL=DELPEL + DELPWR(I)*4.0*TINC/3600.
 642  CONTINUE
 643  WRITE(6,644) DELPEL
 644  FORMAT(1X,'ENERGY USED WITH A SPEED DEPENDENT EFFCIENCY',E12.4)
 645  CONTINUE
C
C  SCHEDULE NEXT PROUT
      CALL SCHED(TX,0,7)
      RETURN
      END
```

```
      SUBROUTINE QSHFT(NPS,ETUA,ETAR)
C
C     PURPOSE:  TO DETERMINE THE TIME UNTIL ARRIVAL(ETAR)OF A
C       VEHICLE AT ITS NEW TARGET QUEUE LOCATION
C
C     CALLED BY: RESCHD
C
C     OUTPUT: TIME UNTIL ARRIVAL AT NEW TARGET LOCATION
C
C     INPUT: NPS-NUMBER OF QUEUE POSITIONS FROM OLD TARGET TO NEW TARGET
C                ETUA-PREDICTED TIME UNTIL ARRIVAL AT OLD TARGET LOCATION
C
      IF(ETUA.EQ.0.0)  GO TO 200
      IF(ETUA.LT.3.6) GO TO 100
C
C   VEHICLE IS CURRENTLY TRAVELING AT STATION SPEED
      ETAR = 1.4*NPS+ETUA
      RETURN
C
C   VEHICLE HAS BEGUN ITS DECELERATION INTO OLD LOCATION
  100 ETAR = 1.4*(NPS-1)+5.35
      RETURN
C
C   VEHICLE IS CURRENTLY AT ITS OLD TARGET LOCATION
  200 ETAR = 1.4*(NPS-1)+7.15
C                 (SAME MODEL AS USED IN TSHIFT)
      RETURN
      END
```

```
      SUBROUTINE REJECT(IDV,T,LS)
C
C    PURPOSE: TO PRINT OUT DATA REGARDING STATION REJECTIONS.
C
C    INPUTS: IDV = VEHICLE ID
C              T = TIME
C             LS = STATION ID
C
C  LAST UPDATE: 09-02-82  BY DLK
C
C
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *     IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *     IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
C
C    START.
      WRITE(6,100) T,IDV,LS
      IZ=ISTA(LS,1)
      WRITE(6,200) (ISQS(LS,1,K),K=1,IZ)
      IZ=ISTA(LS,2)
      WRITE(6,300) (ISQS(LS,2,K),K=1,IZ)
      IZ=ISTA(LS,3)
      WRITE(6,400) (ISQS(LS,3,K),K=1,IZ)
 100  FORMAT(/' AT T = ',F7.2,', VEHICLE ',I3,
     +        ' WAS REJECTED AT STATION ',I3)
 200  FORMAT(' ENQUE:',5I3)
 300  FORMAT(' DKQUE:',5I3)
 400  FORMAT(' EXQUE:',5I3/)
      RETURN
      END
```

```
      SUBROUTINE REMOVE(ID,T)
C
C    PURPOSE: TO REMOVE VEH FROM MAINLINE, FIND AN ENTRANCE QUEUE BERTH,
C             AND SCHEDULE AN ARRIVAL EVENT
C
C    INPUTS: ID=VEH ID
C            T=SIMULATION TIME
C
C    LAST UPDATE: 24-AUG-82 BY HYC
C
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *      IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *      IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      COMMON /BLKRTA/RTFRQ(80),IRTSTC(80,20),IRTSCH(80,3),NSR,NER,
     *          NSRT(80),NSEQ(80,15)
      COMMON /BLKTRH/ THOD,THST,TAST(200),ITOD(20,20,10),ITTS(20,10)
      COMMON /BLKTI/ITPS(200),ITOS(200),ITDS(200),ITVI(200),
     *       ITSB(200),ITRD(200),ITUB(200),ITQC(200),ITTD(200)
      COMMON /BLKRSV/IVRSV(25,40),PTT(25,25),ATT(25,25),IVTHRD(25),
     *          MIDL(5),MIML(5),WSIZE,IRP,JFLGTM
C
C
C    START. CHANGE VEH CONTROL MODE TO DISCRETE.
      IVCM(ID)=3
C
C*** COLLECT O/D TRAVEL TIME. ***************************************
C
C    FIND ORIGIN STATION.
      ISO = ITOS(ID)
C
C    FIND DESTINATION STATION.
      ISD = IVND(ID)
C
C    COMPUTE TRIP TIME.
      TRTIM = T-TAST(ID)
C
C    BIAS THE TRIP TIME BY THE O/D TRAVEL TIME IN THE PTT ARRAY.
      TRTIM = TRTIM - PTT(ISO,ISD)
C
C    SAVE TRIP TIME IN HISTOGRAM ARRAY, ITOD.
      DO 10 I=1,9
 10   IF( TRTIM.LE.(THOD*(I-1)) ) GO TO 20
      I = 10
 20   ITOD(ISO,ISD,I) = ITOD(ISO,ISD,I) + 1
C
C    RESET STATION ENTRY TIME.
      TAST(ID) = T
C
C    RESET ORIGIN STATION
      ITOS(ID)=ISD
C
C*** END OF TRAVEL TIME COLLECTION. **********************************
C
C    CHANGE IVLS TO STATION ID
      IS = IVND(ID)
C     IVLS(ID) IS UPDATED TO IVND(ID) IN BTHANC ++++++++++++++++++++
C
C     DETERMINE BERTH ASSIGNMENT
```

```
      CALL BTHANC(ID)
      IVPX(ID) = 0
C
C
C
C    CHANGE ROUTE ASSIGNMENT
C      WRITE(6,80) T,ID,IVSB(ID),TD
C80    FORMAT(//' IN REMOVE: T = ',F7.2,'; VEH ID=',I3,'; IVSB=',I3,
C    +         '; TD=',F7.2)
C      WRITE(6,90) IVND(ID),IVSR(ID),IRTSTC(IVSR(ID),IVND(ID)),
C    +            IVCM(ID)
       CALL RTASGN(T,ID,IVND(ID),IVSR(ID))
C
C    DETERMINE ID OF NEXT STATION ALONG ROUTE IVSR(ID) AND ASSIGN TO
C    IVND(ID)
       NUMRT=NSRT(IVSR(ID))
       IDR=IVSR(ID)
       DO 100 IRTSQ=1,NUMRT
 100   IF( IVND(ID).EQ.NSEQ(IDR,IRTSQ) ) GO TO 110
       GO TO 120
 110   IVND(ID)=NSEQ(IDR,IRTSQ+1)
 120   CONTINUE
C      WRITE(6,90) IVND(ID),IVSR(ID),IRTSTC(IVSR(ID),IVND(ID)),
C    +            IVCM(ID)
C90    FORMAT('    IVND=',I3,'; IVSR=',I3,'; IRTSTC=',I3,'; IVCM=',I3)
C *****************TEMPORARY DIAGNOSTIC WRITES********************
      IF(IS.NE.ISQS(1,3,10)) GO TO 300
      WRITE(6,200) T,ID,IVLS(ID),IVSB(ID),IVLV(ID),IVSS(ID),
     +  ISQS(IVLS(ID),IVSS(ID),IVSB(ID))
  200 FORMAT('   REMOVE CALL AT TIME:  ',F7.2,6(I5))
C
C *****************************************************************
  300  RETURN
       END
```

```
      SUBROUTINE RESCHD(IDV,IDS,IQ,IP,NQ,NP,IBRAN)
C
C     PURPOSE:  TO RE-TARGET A VEHICLE'S QUEUE DESTINATION WHILE
C       ENROUTE TO ITS CURRENT ASSIGNMENT(EG, IF A DOWNSTREAM
C       POSITION BECOMES AVAILABLE WHILE VEHICLE IDV IS ENROUTE TO
C       AN ASSIGNED LOCATION, RESCHD IS CALLED TO DETERMINE THE
C       ADDITIONAL NUMBER OF POSITIONS TO SHIFT AND CALLS QSHIFT
C       TO DETERMINE THE TIME OF ARRIVAL AT THE NEW DESTINATION).
C
C     CALLED BY: ENQUE,DKQUE,EXQUE
C
C     CALLS: QSHIFT,SCHED
C
C     OUTPUT:  SCHEDULES DKQUE AND EXQUE EVENTS
C         !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++!
C         !  IF NEW TARGET IS NOT ASSIGNED BERTH OR FIRST EXIT QUEUE   !
C         !  POSITION, UPDATES TEVT(IDV) TO ARRIVAL TIME AT NEW TARGET !
C         !  LOCATION(VIA QSHFT)     +++++++++++++++++++++++++++++++++!
C
C     INPUTS:  IDV-VEHICLE ID
C              IQ,IP-CURRENT TARGET QUEUE,QUEUE POSITION
C              NQ,NP-NEW TARGET QUEUE,QUEUE POSITION
C              IDS-STATION ID
C              IBRAN-IDV'S BERTH ASSIGNMENT
C
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
      COMMON /BLKEVR/TEVT(500)
      ETUA = TEVT(IDV)-T
      IF(ETUA.LE.0.0)  ETUA=0.0
C              (VEHICLE IDV IS ENQUEUED AT A QUEUE LOCATION)
      IF(NQ.EQ.2 .AND. NP.EQ.IBRAN)  GO TO 200
      IF(NQ.EQ.3 .AND. NP.EQ.1)  GO TO 300
C
C       NEW TARGET LOCATION IS NEITHER IDV'S ASSIGNED BERTH NOR
C    THE FIRST EXIT QUEUE POSITION
  100 NQP = 0
      IF(IQ.NE.NQ)  NQP=ISTA(IDS,NQ)
      NPS = NQP-NP+IP
C     CALL QSHFT(NPS,ETUA,ETAR)  (ORIGINAL CODE)
      IF(ETUA.EQ.0.0) GO TO 150
      DXIN = 6.0*NPS
      CALL XSHFT(ETUA,DXIN,ETAR)
      GO TO 170
  150 ETAR = 1.4*(NPS-1)+5.35
  170 TEVT(IDV) = T+ETUA+ETAR
      RETURN
C
C       NEW TARGET IS VEHICLE'S ASSIGNED BERTH
  200 NQP = 0
      IF(IQ.NE.NQ) NQP=ISTA(IDS,NQ)
      NPS = NQP-NP+IP
C     CALL QSHFT(NPS,ETUA,ETAR)     (ORIGINAL CODE)
      IF(ETUA.EQ.0.0) GO TO 250
      DXIN = 6.0*NPS
      CALL XSHFT(ETUA,DXIN,ETAR)
      GO TO 270
  250 ETAR = 1.4*(NPS-1)+5.35
  270 CALL SCHED(T+ETUA+ETAR,IDV,3)
      RETURN
```

```
C
C        NEW TARGET LOCATION IS FIRST EXIT QUEUE POSITION
  300 NQP = 0
      IF(IQ.NE.NQ)  NQP=ISTA(IDS,NQ)
      NPS = NQP-NP+IP
C     CALL QSHFT(NPS,ETUA,ETAR)     (ORIGINAL CODE)
      IF(ETUA.EQ.0.0) GO TO 350
      DXIN = 6.0*NPS
      CALL XSHFT(ETUA,DXIN,ETAR)
      GO TO 370
  350 ETAR = 1.4*(NPS-1)+5.35
  370 CALL SCHED(T+ETUA+ETAR,IDV,5)
      RETURN
      END
```

```
      SUBROUTINE RJRMV(IDV,IDS,T)
C
C
C   PURPOSE: TO REMOVE FROM CONTINUOUS MODE A VEHICLE THAT WAS REJECTED
C         AT THE FIRST UPSTREAM STATION FROM IDS.  THE VEHICLE,IDV, IS
C         ASSIGNED TO THE NEXT STATION ALONG ITS ROUTE UNLESS THE
C         REJECT STATION WAS THE FIRST STATION IN THE LAST SET OF IDV'S
C         ASSIGNED ROUTE.  IF SO, IDV IS ROUTED BACK TO THE REJECT
C         STATION AFTER STOP AT IDS.
C
C   CALLED BY:   NEWLV
C
C   CALLS:  RTASGN,BTHANC
C
C   INPUTS:  IDV- VEHICLE ID
C            IDS- ID OF FIRST STATION DOWNSTREAM OF REJECT STATION
C            T- CURRENT TIME
C
C   OUTPUT:  DETERMINES NEXT DESTINATION,NEW ROUTE ASSIGNMENT, AND
C            INITIATES STATION PROCESSING FOR IDS.
C
C
C
C    LAST UPDATE: 10-SEP-82 BY DLK
C
C
       COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *      IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *      IVCS(200,20),IVQCH(200),IVSB(200)
       COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
       COMMON /BLKRTA/RTFRQ(80),IRTSTC(80,20),IRTSCH(80,3),NSR,NER,
     *          NSRT(80),NSEQ(80,15)
       COMMON /BLKTRH/THOD,THST,TAST(200),ITOD(20,20,10),ITTS(20,10)
       COMMON /BLKRSV/IVRSV(25,40),PTT(25,25),ATT(25,25),IVTHRD(25),
     *     MIDL(5),MIML(5),WSIZE,IRP,JFLGTM
       COMMON /BLKTI/ITPS(200),ITOS(200),ITDS(200),ITVI(200),
     *     ITSB(200),ITRD(200),ITUB(200),ITQC(200),ITTD(200)
C
C
      IVCM(IDV) = 3
C
C*** COLLECT O/D TRAVEL TIME. *************************************
C
C    FIND ORIGIN STATION.
       ISO = ITOS(IDV)
C
C    FIND DESTINATION STATION.
       ISD = IDS
C
C    COMPUTE TRIP TIME.
       TRTIM = T-TAST(IDV)
C
C    BIAS THE TRIP TIME BY THE O/D TRAVEL TIME IN THE PTT ARRAY.
       TRTIM = TRTIM - PTT(ISO,ISD)
C
C    SAVE TRIP TIME IN HISTOGRAM ARRAY, ITOD.
       DO 10 I=1,9
 10    IF( TRTIM.LT.(THOD*I) ) GO TO 20
       I = 10
 20    ITOD(ISO,ISD,I) = ITOD(ISO,ISD,I) + 1
```

```
C
C    RESET STATION ENTRY TIME.
       TAST(IDV) = T
C
C    RESET ORIGIN STATION
       ITOS(ID)=IDS
C  *****************END OF TRAVEL TIME COLLECTION ****************
C
      IF(IRTSTC(IVSR(IDV),IVND(IDV)).EQ.-2) GO TO 200
C
C   REJECT STATION IS NOT FIRST STATION IN LAST SET OF IDV'S ROUTE
      IF(IRTSTC(IVSR(IDV),IDS).NE.0) IVND(IDV)=IDS
      CALL RTASGN(T,IDV,IVND(IDV),IVSR(IDV))
C
C    DETERMINE ID OF NEXT STATION ALONG ROUTE IVSR(IDV) AND ASSIGN TO
C    IVND(IDV)
       NUMRT=NSRT(IVSR(IDV))
       IDR=IVSR(IDV)
       DO 100 IRTSQ=1,NUMRT
 100   IF( IVND(IDV).EQ.NSEQ(IDR,IRTSQ) ) GO TO 110
       GO TO 120
 110   IVND(IDV)=NSEQ(IDR,IRTSQ+1)
 120   CONTINUE
       GO TO 300
C
C    REJECT STATION WAS FIRST IN LAST SET
 200  IVSR(IDV) = -1
 300  KSTORE = IVND(IDV)
C  TEMPORARILY SET IVND(IDV) =IDS FOR USE BY LOGIC IN BTHANC
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      WRITE(6,400) IDV,IDS,IVRF(IDV),IVND(IDV)
 400  FORMAT(' FROM RJRMV(IDV,IDS,IVRF,IVND): ',4I5)
C
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      IVND(IDV) = IDS
      CALL BTHANC(IDV)
      IVND(IDV) = KSTORE
      RETURN
      END
```

```
      SUBROUTINE RSTART(NVEH)
C
C     PURPOSE: TO SAVE SIMULATION VARIABLES FOR RESTART.
C
C     INPUT: JFLGV = 0: NO VEHICLE DISPATCHES
C                    1: VEHICLE DISPATCHES
C
C     OUTPUT: NVEH = NUMBER OF VEHICLES IN NETWORK
C
C     LAST UPDATED: 24-AUG-82 BY HYC -- 9/7 BY PJM 08:50
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *       IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *       IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKVR/VACC(200),VVEL(200),VPOS(200),VENR(200)
      COMMON /BLKLI/LNTY(90),LNFV(90),LNTV(90),LNEL(90),LNLI(90),
     *       LNRN(90),LNLV(90)
      COMMON /BLKLR/ALMV(90),ALLL(90)
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      COMMON /BLKTI/ITPS(200),ITOS(200),ITDS(200),ITVI(200),
     *        ITSB(200),ITRD(200),ITUB(200),ITQC(200),ITTD(200)
      COMMON /BLKTR/TRAT(200),TRPT(200),TRBT(200),TRCT(200)
      COMMON /BLKEVI/IETY(500),IEID(500),IEPT(500),IEES(500)
      COMMON /BLKEVR/TEVT(500)
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
      COMMON /BLKLV2/LSTN(10)
      COMMON /BLKSCH/IELAST,IEMAX,IEHEAD,IETAIL
      COMMON /BLKVD/VDT,NVDS(200),IVDSP(20),NDSP,NFLT,IDSP
      COMMON /BLKRTA/RTFRQ(80),IRTSTC(80,20),IRTSCH(80,3),NSR,NER,
     *          NSRT(80),NSEQ(80,15)
      COMMON /BLKENR/WPAX,WV,WROT,DRGCF,ARR,BRR,AF,BF,G,EGB,EM,
     *          EPCU,RCPTY
      COMMON /BLKLEN/PENGRY(90),PPWR
      COMMON /BLKXTR/ VDIST(200),JFLGE,ISRN(20)
      COMMON /BLKTRH/ THOD,THST,TAST(200),ITOD(20,20,10),ITTS(20,10)
      COMMON /BLKSTA/ MSTAR(20),LFLOW(90)
      COMMON /BLKRSV/IVRSV(25,40),PTT(25,25),ATT(25,25),IVTHRD(25),
     *   MIDL(5),MIML(5),WSIZE,IRP,JFLGTM
      COMMON /BLKENG/ PENGYD(90),DELPWR(200),POWER,POWERD,SWAUX,
     *   JFLGED
C
      NAMELIST /INPFP/VDT,NDSP,IVDSP
C
C     READ EVENT VARIABLES: BLKEVI,BLKEVR,BLKSCH
      NEV=500
      READ(71) T,(IETY(I),I=1,NEV),(IEID(I),I=1,NEV),
     +            (IEPT(I),I=1,NEV),(IEES(I),I=1,NEV),
     +            (TEVT(I),I=1,NEV),IELAST,IEMAX,IEHEAD,IETAIL
C
C     READ VEHICLE VARIABLES: BLKVI,BLKVR,BLKXTR
      NV=200
      READ(71) NVEH,(IVCM(I),I=1,NV),(IVSR(I),I=1,NV),(IVND(I),I=1,NV),
     +          (IVLV(I),I=1,NV),(IVLV2(I),I=1,NV),(IVTV(I),I=1,NV),
     +          (IVSS(I),I=1,NV),(IVLS(I),I=1,NV),(IVTP(I),I=1,NV),
     +          (IVRF(I),I=1,NV),(IVPX(I),I=1,NV),
     +          ((IVCS(I,J),I=1,NV),J=1,20),(IVQCH(I),I=1,NV),
     +          (IVSB(I),I=1,NV),(VACC(I),I=1,NV),(VVEL(I),I=1,NV),
     +          (VPOS(I),I=1,NV),(VENR(I),I=1,NV),(VDIST(I),I=1,NV)
C
C     READ LINK VARIABLES: BLKLI,BLKLR,BLKLV2
```

```
      NL=90
      READ(71) (LNTY(I),I=1,NL),(LNFV(I),I=1,NL),(LNTV(I),I=1,NL),
     +          (LNEL(I),I=1,NL),(LNLI(I),I=1,NL),(LNRN(I),I=1,NL),
     +          (LNLV(I),I=1,NL),(ALMV(I),I=1,NL),(ALLL(I),I=1,NL),
     +          (LSTN(I),I=1,10)
C
C     READ STATION VARIABLES: BLKSI,BLKXTR
      NS=20
      READ(71) ((ISTA(I,J),I=1,NS),J=1,10),
     +          ((ISTL(I,J),I=1,NS),J=1,10),
     +          (((ISQS(I,J,K),I=1,NS),J=1,3),K=1,10),
     +          (ISTX(I),I=1,NS),(ISRN(I),I=1,NS)
C
C     READ DISPATCH VARIABLES: BLKVD,BLKRTA
      NR=80
      READ(71) VDT,(NVDS(I),I=1,NV),(IVDSP(I),I=1,NS),NDSP,NFLT,IDSP,
     +          (RTFRQ(I),I=1,NR),
     +          ((IRTSTC(I,J),I=1,NR),J=1,NS),
     +          ((IRTSCH(I,J),I=1,NR),J=1,3),NSR,NER,
     +          (NSRT(I),I=1,NR),
     +          ((NSEQ(I,J),I=1,NR),J=1,15)
C
C     READ ENERGY VARIABLES: BLKENR,BLKLEN
      READ(71) WPAX,WV,WROT,DRGCF,ARR,BRR,AF,BF,G,EGB,EM,EPCU,RCPTY,
     +          (PENGRY(I),I=1,NL),(PENGYD(J),J=1,NL),POWER,POWERD,
     +          (DELPWR(K),K=1,NVEH),SWAUX,JFLGED
C
C     READ TRIP VARIABLES: BLKTI,BLKTR,TRHIST,BLKSTA
      NT=200
      READ(71) (ITPS(I),I=1,NT),(ITOS(I),I=1,NT),(ITDS(I),I=1,200),
     +         (ITVI(I),I=1,NT),(ITSB(I),I=1,NT),(ITRD(I),I=1,200),
     +         (ITUB(I),I=1,NT),(ITQC(I),I=1,NT),(ITTD(I),I=1,200),
     +         (TRAT(I),I=1,NT),(TRPT(I),I=1,NT),(TRBT(I),I=1,200),
     +         (TRCT(I),I=1,NT),
     +         THOD,THST,(TAST(I),I=1,NT),
     +         (((ITOD(I,J,K),I=1,20),J=1,20),K=1,20),
     +         ((ITTS(I,J),I=1,20),J=1,10),
     +         (MSTAR(I),I=1,20)
C
C        READ IN RESERVATION MATRIX DATA
      READ(71)
     +         IRP,WSIZE,
     +         (MIDL(I),I=1,5),(MIML(I),I=1,5),
     +         ((IVRSV(I,J),I=1,25),J=1,40),
     +         (IVTHRD(I),I=1,25),
     +         (IVQCH(I),I=1,200)
C
 100  RETURN
      END
```

```
      SUBROUTINE RSVUPD(T)
C
C   PURPOSE:  TO RESET RESERVATION MATRIX POINTER, IRP, TO NEXT TIME
C        COLUMN AND CLEAR CURRENT COLUMN.
C
C   SCHEDULED BY:  RSVUPD
C
C   INPUTS:  IRP-CURRENT POINTER
C            WSIZE-WIDTH OF MATRIX CELL(SEC)
C            IVRSV-RESERVATION MATRIX
C            T-CURRENT CLOCK TIME
C
C   OUTPUTS:  IRP-RETURNS NEW POINTER
C
C   LAST CHANGE 27-08-82  BY DLK
C
      COMMON /BLKRSV/IVRSV(25,40),PTT(25,25),ATT(25,25),IVTHRD(25),
     *    MIDL(5),MIML(5),WSIZE,IRP,JFLGTM
      J = IRP
      IRP = IRP+1
      IF(IRP.GT.40) IRP=1
      DO 10 I=1,25
   10 IVRSV(I,J) = 0
      CALL SCHED(T+WSIZE,0,13)
      RETURN
      END
```

```
      SUBROUTINE RTASGN(T,IDV,IDSTAT,IDRT)
C
C  PURPOSE:  TO REASSIGN VEHICLE IDV TO A NEW ROUTE IF A NEW
C         ASSIGNMENT IS NECESSARY.  THE ASSIGNMENT IS RETURNED
C         VIA IDRT.
C
C
      COMMON /BLKRTA/RTFRQ(80),IRTSTC(80,20),IRTSCH(80,3),NSR,NER,
     *       NSRT(80),NSEQ(80,15)
      COMMON /BLKSA/IRTSAT(80,10),IRTSAI(80,10)
C        IRTSAT,IRTSAI RECORD INTER-DISPATCH AND -ARRIVAL SCHEDULE
C        ADHERANCE DATA IN 10 BINS(-2.5 MIN TO +2.5 MIN IN .5 INTERVALS)
C
C     DETERMINE NEED FOR REASSIGNMENT
C
      IF(IDRT.LT.0)  GO TO 12
C                                  (IE,IDV HAS NOT YET BEEN ASSIGNED TO
C                                    A ROUTE; IDSTAT ASSIGNED BY VDISP)
C
      IF(IDRT.GT.NSR)  GO TO 10
C                                  (IE, IDRT IS AN EMPTY DIST RT)
      IF(IRTSTC(IDRT,IDSTAT).EQ.-2)  GO TO 10
C                          (IE, IDV IS APPROACHING LAST SET OF IDRT)
C
C       OTHERWISE, A NEW ASSIGNMENT IS UNNECESSARY
C
      RETURN
C
C       A NEW ASSIGNMENT IS REQUIRED FOR VEHICLE IDV
C
C              UPDATE ROUTE TERMINATION SCHED. ADHR. ARRAY
   10 DLTAT = T-IRTSCH(IDRT,3)-IRTSCH(IDRT,1)
      SI = DLTAT/30.0
      I = 6.0+SI
      IF(I.LT.1) I=1
      IF(I.GT.10) I=10
C
      IRTSCH(IDRT,3) = T

C
C           DETERMINE NEXT ASSIGNMENT
C
C     DETERMINE ROUTE THAT IS MOST BEHIND SCHEDULE(IP) AND ROUTE
C         THAT WOULD BE LEAST AHEAD OF SCHEDULE(IN)
C
   12 PDIFF = 0.0
      IP = 0
      IN = 0
      DIFFN = -1000.
      NRT = NER+NSR
      DO 20 IRT = 1,NRT
           IF(IRTSTC(IRT,IDSTAT).LT.2)  GO TO 20
           DLT = T-IRTSCH(IRT,2)-IRTSCH(IRT,1)
           IF(DLT.LE.0.0)  GO TO 15
                IF(DLT.LT.PDIFF)  GO TO 20
                     PDIFF = DLT
                     IP = IRT
                     GO TO 20
   15           IF(DLT.LT.DIFFN)  GO TO 20
                DIFFN = DLT
```

```
                IN = IRT
   20        CONTINUE
      IF(IP.EQ.0)  GO TO 30
C
C   ELSE, NEW ASSIGNMENT IS TO ROUTE IP
C
      IDRT = IP
      GO TO 40
C
C   NEW ASSIGNMENT IS TO ROUTE IN
C
   30 IDRT = IN
C
C    UPDATE ROUTE ASSIGNMENT SCHED.ADHR. ARRAY
   40 DLTAI = T-IRTSCH(IDRT,2)-IRTSCH(IDRT,1)
      IRTSCH(IDRT,2) = T
      SI = DLTAI/30.0
      I = 6.0+SI
      IF (I.LT.1) I=1
      IF (I.GT.10) I=10
      IRTSAI(IDRT,I) = IRTSAI(IDRT,I)+1
      RETURN
      END
```

MEMBER RTSTI   DOES NOT EXIST.

```
      SUBROUTINE RTDTI
C
C  PURPOSE: TO INITIALIZE SERVICE ROUTE ASSIGNMENT DATA
C       ARRAYS IRTSTIC & IRTDCH
C
C
      COMMON /BLKRTA/RTFRQ(80),IRTSTC(80,20),IRTSCH(80,3),NSR,NER,
     *      NSRT(80),NSEQ(80,15)
C
C
      DO 10 I=1,80
          DO 10 J=1,20
   10     IRTSTC(I,J)=0
      NRT=NSR+NER
      DO 30 IRT = 1,NRT
          ISTL=NSRT(IRT)
          DO 20 IST = 1,ISTL
               IF(IST.EQ.1)  IRTSTC(IRT,NSEQ(IRT,1))=2
   20      IF(IST.GT.1)  IRTSTC(IRT,NSEQ(IRT,IST))=1
           LST = NSRT(IRT)+1
   30     IRTSTC(IRT,NSEQ(IRT,LST))=-2
      DO 40 IRT = 1,80
          IRTSCH(IRT,1)=0
          IRTSCH(IRT,2)=0
          IRTSCH(IRT,3)=0
   40 IF(IRT.LE.(NSR+NER))  IRTSCH(IRT,1)=3600.0/RTFRQ(IRT)
      RETURN
      END
```

```
      SUBROUTINE SCHED(T,ID,ITYPE)
C
C     PURPOSE: TO SCHEDULE VEHICLE EVENTS
C
C     INPUTS: T=EVENT TIME
C             ID=VEHICLE/TRIP ID
C             ITYPE=EVENT TYPE
C
      COMMON /BLKEVR/TEVT(500)
      COMMON /BLKEVI/IETY(500),IEID(500),IEPT(500),IEES(500)
      COMMON /BLKSCH/IELAST,IEMAX,IEHEAD,IETAIL
C
C     START.  IF EVENT IS NOT A VEH TYPE, GO TO "FIND EVENT NO..."
C     ELSE, SET EVENT NO. TO VEH ID
      IF( ITYPE.EQ.10 ) GO TO 50
      IF( ITYPE.EQ.1 .OR. ITYPE.GT.6 ) GO TO 100
 50   IE=ID
      GO TO 120
C
C     FIND EVENT NO. FOR NON-VEHICLE EVENT
 100  IE=IELAST
 110  IE=IE+1
      IF( IE.EQ.IELAST ) GO TO 900
      IF( IE.GT.IEMAX ) IE=201
      IF( IEES(IE).EQ.1 ) GO TO 110
      IELAST=IE
C
C     SET UP EVENT.
 120  IETY(IE)=ITYPE
      IEID(IE)=ID
      IEES(IE)=1
      TEVT(IE)=T
C
C     RESHUFFLE EVENT POINTERS.
      I=IEHEAD
      IF( I.EQ.0 ) GO TO 300
      IF( T.LT.TEVT(I) ) GO TO 400
 150  J=IEPT(I)
      IF( J.EQ.0 ) GO TO 500
      IF( T.LT.TEVT(J) ) GO TO 600
      I=J
      GO TO 150
C
C     ONLY EVENT IN LIST
 300  IEPT(IE)=0
      IEHEAD=IE
      IETAIL=IE
      RETURN
C
C     FIRST EVENT IN LIST
 400  IEPT(IE)=I
      IEHEAD=IE
      RETURN
C
C     LAST EVENT IN LIST
 500  IEPT(IE)=0
      IEPT(I)=IE
      IETAIL=IE
      RETURN
C
```

```
C     EVENT IN MIDDLE OF LIST
 600    IEPT(IE)=J
        IEPT(I)=IE
        RETURN
C
C     MAXIMUM NUMBER OF EVENTS EXCEEDED. SCHEDULE ERROR.
 900    IE=IEHEAD
        IETY(IE)=10
        TEVT(IE)=0.
        IEES(IE)=1
        IEID(IE)=ID
        RETURN
        END
```

```
      SUBROUTINE STATES(NVEH,TX)
C
C     PURPOSE: TO PRINT VEHICLE STATES IN CONTINUOUS MODE
C
C     INPUT: NVEH = NUMBER OF VEHICLES IN SYSTEM
C              TX = NEXT PRINT TIME
C     LAST UPDATE: 9/09/82 -- BY  PJM 02:20 AM
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *     IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *     IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKVR/VACC(200),VVEL(200),VPOS(200),VENR(200)
      COMMON /BLKLI/LNTY(90),LNFV(90),LNTV(90),LNEL(90),LNLI(90),
     *     LNRN(90),LNLV(90)
      COMMON /BLKLR/ALMV(90),ALLL(90)
      COMMON /BLKLUP/DISTL(90,90),NPNODE(90),IEXIT(90,90)
C     COMMON /BLKLEN/PENGRY(90),PPWR
      COMMON /BLKENG/PENGYD(90),DELPWR(200),POWER,POWERD,
     *     SWAUX,JFLGED
      COMMON /BLKXTR/ VDIST(200),JFLGE,ISRN(20)
C
C     IF NVEH=0, GO TO "RESCHEDULE NEXT STATES..."
      IF( NVEH.EQ.0 ) GO TO 900
C
C     START VEH LOOP
      DO 500 I=1,NVEH
C
C
C     IF IN DISCRETE MODE, GO TO "PRINT STATION..."
    5 IF( IVCM(I).EQ.3 ) GO TO 200
C
C     COMPUTE SPACING
      IL=IVLV(I)
      IF( IL.EQ.0 ) GO TO 50
      LNKTY=LNTY(IVLS(I))
      GO TO (20,40,20,40,40,20),LNKTY
   20 SS=VPOS(IL)-VPOS(I)+DISTL(IVLS(I),IVLS(IL))
      GO TO 60
   40 IF( IVLS(IL).NE.LNLI(IVLS(I)) ) GO TO 20
      SS=ALLL(IVLS(I))-ALLL(IVLS(IL))+VPOS(IL)-VPOS(I)
      GO TO 60
   50 SS=0.0
   60 CONTINUE
C
C     PRINT VEHICLE STATES
      WRITE(6,100) I,VPOS(I),VVEL(I),VACC(I),SS,IVLS(I),IVND(I)
     +             ,IVLV(I),IVSR(I),IVTV(I),IVCM(I)
  100 FORMAT('  I=',I3,'; POS=',F7.2,'; VEL=',F7.2,'; ACC=',F7.2,
     +       '; SPC=',F7.2,
     +       '; LINK=',I3,'; DEST=',I3,'; LV1=',I3,'; SRT=',I3,
     +       '; TV=',I3,'; MODE=',I3)
      GO TO 500
C
C     PRINT STATION ID OF DISCRETE VEHS
  200 WRITE(6,210) I,IVLS(I),IVSS(I),IVSB(I)
  210 FORMAT('  I=',I3,'; STAT ID=',I3,'; QUEUE=',I3,'; BERTH=',I3)
C
  500 CONTINUE
C
C     PRINT ENERGY VARIABLES
```

```
      WRITE(6,510) POWER,POWERD
 510  FORMAT(/,'   POWER=',F14.2,5X,'POWERD=',F14.2,//)
C     WRITE(6,520) (PENGRY(I),I=1,88)
C520  FORMAT(10F10.2)
      IF( JFLGED.EQ.0) GO TO 640
C   CALCULATE INDIVIDUAL VEHICLE ENERGY PER METER

      WRITE(6,636)
      WRITE(6,637)
  636 FORMAT(1X,4('VEH         VEH          VEH  ** '))
  637 FORMAT(1X,4('ID          ENERGY/M     POWERD ** '),//)
      DO 639  I=1,196,4
      I1=I+1
      I2=I+2
      I3=I+3
      IF(VDIST(I1) .EQ. 0.) GO TO 640
      IF(VDIST(I2) .EQ. 0.) GO TO 640
      IF(VDIST(I3) .EQ. 0.) GO TO 640
      VEPM1=VENR(I)/VDIST(I)
      VEPM2=VENR(I1)/VDIST(I1)
      VEPM3=VENR(I2)/VDIST(I2)
      VEPM4=VENR(I3)/VDIST(I3)
      WRITE(6,638) I,VEPM1,DELPWR(I),I1,VEPM2,DELPWR(I1),I2,VEPM3,
     + DELPWR(I2),I3,VEPM4,DELPWR(I3)
  638 FORMAT(1X,4(I3,E12.5,1X,E12.5, '   * '))
      IF (I3.GE.NVEH) GO TO 640
  639 CONTINUE
  640 CONTINUE

C
C   RESCHEDULE NEXT STATES EVENT AT T=TX
 900  CALL SCHED(TX,0,12)
C
      RETURN
      END
```

```
      SUBROUTINE TDECEL(IDV,LID,TDEC)
C
C    PURPOSE: TO COMPUTE THE TIME TO DECELERATE FROM LINE SPEED
C             TO ZERO USING THE PRECISION STOP CONTROLLER.
C
C    INPUT: IDV=VEHICLE ID
C           LID=ID OF STATION APPROACH LINK
C
C    OUTPUT: TDEC=TIME TO DECELERATE TO ZERO
C
C  LAST UPDATE: 09-14-82  BY DLK
C
C
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *      IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *      IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKVR/VACC(200),VVEL(200),VPOS(200),VENR(200)
      COMMON /BLKLR/ALMV(90),ALLL(90)
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      COMMON /BLKCN/HD,GK,DELTA,AS,XJS,CZ1,CZ2,CZ3,VLEN
C
C    START. DEFINE RAMP FUNCTION WHICH COMPUTES THE DISTANCE NEEDED TO
C    DECELERATE TO ZERO FROM V ON SERVICE LIMITS.
      RAMP(V)=V*( V/AS + AS/XJS )
C
C    ESTABLISH PARAMETERS.
      VEL=VVEL(IDV)
      IDS=IVND(IDV)
      VMAX=ALMV(LID)
C
C    FIND NUMBER OF BERTHS TO TRAVEL IN STATION.
      ISH=ISTA(IDS,1)-IVSB(IDV)
      IF( IVSS(IDV).EQ.2 ) ISH=ISTA(IDS,1)+ISTA(IDS,2)-IVSB(IDV)
C
C    COMPUTE TOTAL DISTANCE TO TRAVEL
      DTOT=RAMP(VMAX)+VLEN*ISH
C
C    COMPUTE TDEC
      DXTRA=VPOS(IDV)-ALLL( LID )
      TDEC=( DTOT-RAMP(VEL)-DXTRA )/VEL + 0.547*VEL + 5.632
C
      RETURN
      END
```

```
      SUBROUTINE VDISP(IVP,NFLTO)
C
C  PURPOSE:  TO SCHEDULE AND ASSIGN DESTINATIONS FOR
C         INITIAL DISPATCH OF VEHICLE FLEET
C
C  INPUT DATA:
C             NVDS(I)--DESTINATION STATION FOR ITH VEHICLE DISPATCHED
C             IVDSP(I)-- VECTOR OF DISPATCH STATION ID'S
C             NDSP--NO. OF DISPATCH STATIONS
C             NFLT--ALLOCATED FLEET SIZE
C             VDT--STATION INTER-VEHICLE DISPATCH TIME
C             IVP--LAST ASSIGNED VEHICLE ID
C
C  LAST UPDATE: 22-AUG-82 BY HYC
C
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *     IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *     IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKVD/VDT,NVDS(200),IVDSP(20),NDSP,NFLT,IDSP
      COMMON /BLKSI/ISTA(20,10),ISTL(20,20),ISQS(20,3,10),ISTX(20)
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
      COMMON /BLKEVR/TEVT(500)
      COMMON /BLKTI/ITPS(200),ITOS(200),ITDS(200),ITVI(200),
     *  ITSB(200),ITRD(200),ITUB(200),ITQC(200),ITTD(200)
      COMMON /BLKTRH/THOD,THST,TAST(200),ITOD(20,20,10),ITTS(20,10)
C
C   DETERMINE NEXT DISPATCH STATION ****************************
C
C       INCREMENT DISPATCH POINTER
C
      IDSP = IDSP+1
      IF(IDSP.GT.NDSP)  IDSP=1
C
C        CHECK EXIT QUEUE STATUS OF STATION IVDSP(IDSP)
C
C
      IF(ISQS(IVDSP(IDSP),3,ISTA(IVDSP(IDSP),3)).EQ.0)  GO TO 20
C
C         EXIT QUEUE OCCUPIED--CHECK OTHER DISPATCH STATIONS
C
      DO 10 I=1,NDSP
           IDSP = IDSP+1
           IF(IDSP.GT.NDSP)  IDSP=1
      IF(ISQS(IVDSP(IDSP),3,ISTA(IVDSP(IDSP),3)).EQ.0)  GO TO 20
   10 CONTINUE
      GO TO 30
C
C     SCHEDULE DISPATCH FROM STATION IVDSP(IDSP) *******************
C
C         ASSIGN NEXT VEHICLE ID
C
   20 IVP = IVP+1
      ITOS(IVP) = IVDSP(IDSP)
      TAST(IVP) = T
C
C     SELECT & ASSIGN DESTINATION STATION ***************************
C
      IVND(IVP) = NVDS(IVP-NFLTO)
C
```

```
C     INITIALIZE VEHICLE STATUS:
C             CURRENT STATION,STATION STATUS,CONTROL MODE,SERVICE ROUTE
C
      IVLS(IVP) = IVDSP(IDSP)
      IVSS(IVP) = 4
      IVCM(IVP) = 3
      IVSR(IVP) = -1
C
C      SCHEDULE EXIT QUEUE ARRIVAL ************************************
C
      TSCH = T+0.01
      IF(ISTA(IVDSP(IDSP),6).NE.0) GO TO 25
      IVSB(IVP) = 1
      ISTA(IVDSP(IDSP),6) = ISTA(IVDSP(IDSP),6)+1
      ISQS(IVDSP(IDSP),3,1) = -IVP
      CALL SCHED(TSCH,IVP,5)
      GO TO 30
   25 TEVT(IVP) = TSCH
C
C      UPDATE STATION QUEUE STATUS ************************************
      ISTA(IVDSP(IDSP),6) = ISTA(IVDSP(IDSP),6)+1
      IVSB(IVP) = ISTA(IVDSP(IDSP),3)
      ISQS(IVDSP(IDSP),3,IVSB(IVP))= -IVP
C
C      RESCHEDULE DISPATCH EVENT IF FLEET IS LESS THAN DESIRED LEVEL ***
C
   30 TDSP = T+(VDT/NDSP)
      IF((IVP-NFLTO).LT.NFLT)  CALL SCHED(TDSP,IVP,9)
      RETURN
      END
```

```
      SUBROUTINE VENGRY(I,NVPX,VOLD,VP,ACC)
C
C  PURPOSE: COMPUTES ELECTRICAL PROPULSIVE POWER REQUIRED
C         BY VEHICLE IVEH DURING PREVIOUS TIME INTERVAL(OR
C         RETURNED TO LINE VIA REGENERATIVE BRAKING)

C  CALLED FROM; CONSUB  (FOR EACH VEHICLE IN CONTINUOUS MODE)
C
C  INPUTS:
C       I=IVEH- VEHICLE ID
C         NVPX- NO. OF PASSENGERS ON BOARD
C         VOLD,VP- VELOCITY AT BEGINNING & END OF PREVIOUS INTERVAL(M/S)
C         ACC- ACCELERATION COMMAND DURING PREVIOUS INTERVAL(M/S*S)
C         WPAX- AVG PASS. WEIGHT(LB)
C         WV- VEHICLE EMPTY WEIGHT(LB)
C         WROT- EQUIVALENT WEIGHT OF ROTATING EQUIPMENT ON VEHICLE(LB)
C              WROT=((4*IWHEELS+IMOTOR*(GEAR RATIO)**2)/(RWHEELS**2)
C         DRGCF- DRAG FORCE COEFF.(1/2 * RHO * CD * A    SLUGS/FT)
C         ARR,BRR- ROLLING RESISTANCE FORCE COEFF.(MU LB/LB;
C                    MU*CC  LB*SEC/LB*FT)
C         AF,BF- FRICTIONAL FORCE COEFF.(GUIDE WHEELS ETC. LB,LB*SEC/FT)
C         G- ACC. DUE TO GRAVITATIONAL FORCE (FT/SEC*SEC)
C         EGB,EM,EPCU- GEAR BOX,MOTOR,PCU SYSTEM EFFICIENCIES
C                (NOTE: HIGHER FIDELITY MODEL MAY REQUIRE EM=F(V) )
C         RCPTY- ESTIMATED AVG LINE RECEPTIVITY FOR REGENERATED POWER)
C
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *      IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *      IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKVR/VACC(200),VVEL(200),VPOS(200),VENR(200)
      COMMON /BLKTIM/T,TEND,TINC,TDINC,TDWELL
      COMMON /BLKENR/ WPAX,WV,WROT,DRGCF,ARR,BRR,BF,AF,G,EGB,EM,EPCU,
     +            RCPTY
      COMMON /BLKLEN/PENGRY(90),PPWR
      COMMON /BLKENG/ PENGYD(90),DELPWR(200),POWER,POWERD,
     +            SWAUX,JFLGED
C
C    COMPUTE AVERAGE VELOCITY DURING PREVIOUS INTERVAL
C
      VAVG = ((VOLD+VP)*3.281)/2.0
C
C     COMPUTE THRUST FORCE,TF, REQUIRED TO ACHIEVE COMMANDED
C     ACCELERATION(ASSUME ZERO WIND AND GRADE)
C
      ACL = ACC*3.281
      WG = WV+NVPX*WPAX
      WGEQ = WG+WROT
      FDRAG = DRGCF*(VAVG*VAVG)
      FRR = (ARR+BRR*VAVG)*WG
      FF = AF+BF*VAVG
      FINRT = (WGEQ*ACL)/G
      TF = FDRAG+FRR+FF+FINRT
C
C    DETERMINE IF MOTORING OR BRAKING
C
      IF (TF.LT.0.0)   GO TO 100

C
C     COMPUTE REQUIRED MECHANICAL POWER TO ACCELERATE VEHICLE
```

```
C
      PMECH = (TF*VAVG)/737.6
C
C    DETERMINE REQUIRED ELECTRICAL POWER
C
      PEL = PMECH/(EGB*EM*EPCU)
C
      EMV =EM*(1.-(VAVG/49.2-1.)**2)
      IF(VAVG .LT. 5.0)EMV=0.2
      PELV = PMECH/(EMV*EGB*EPCU)
      GO TO 105
  100 IF(RCPTY.GT.0.001)   GO TO 110
C
C    LINE RECEPTIVITY IS 0.0, NO POWER RETURNED VIA REGENERATION
C
      PEL = 0.0
      RETURN
C
C    LINE RECEPTIVITY > 0.0; DETERMINE AMOUNT OF POWER RETURNED TO LINE
C
C       COMPUTE AVAILABLE MECHANICAL POWER
C
  110 PMECH = ((TF*VAVG)/737.6)*EGB
C
C   ELECTRICAL POWER AVAILABLE TO PUT INTO LINE
C
      PAVAIL = PMECH*EM*EPCU  -SWAUX*6.0
C
C    AMOUNT OF POWER ACTUALLY RETURNED TO LINE
C
      PEL = PAVAIL*RCPTY

C    ACCUMULATE ENERGY CONSUMPTION BY LINK AND BY VEHICLE
 105  PENGRY(IVLS(I))=PENGRY(IVLS(I))+(PEL*4.0*TINC)/3600.
      IF(PEL .LT.0.) GO TO 120
      VENR(I)=VENR(I)+(PEL*4.0*TINC)/3600.
      PENGYD(IVLS(I))=PENGYD(IVLS(I))+(PEL*4.0*TINC)/3600.
      DELPWR(I)=  PELV +DELPWR(I)
      POWERD=POWERD+PEL
  120 CONTINUE
      POWER=POWER+PEL

      RETURN
      END
```

```
      SUBROUTINE XSHFT(DTIN,DXN,DTOUT)
C
C   PURPOSE: TO COMPUTE THE EXTRA TIME NEEDED TO ARRIVE AT A NEW
C            DESTINATION
C
C   INPUT: DTIN = THE TIME REMAINING TO REACH THE OLD DESTINATION
C          DXN  = THE DISTANCE FROM THE OLD TO THE NEW DESTINATION
C
C   OUTPUT: DTOUT = THE EXTRA TIME NEEDED TO REACH THE NEW DESTINATION
C
C   COMMENTS: THIS SUBROUTINE USES DATA FROM THE PRECISION STOP
C             SIMULATION. A MAXIMUM STATION SPEED, VSTAT, IS ASSUMED.
C
      COMMON /BLKVI/IVCM(200),IVSR(200),IVND(200),IVLV(200),IVLV2(200),
     *      IVTV(200),IVSS(200),IVLS(200),IVTP(200),IVRF(200),IVPX(200),
     *      IVCS(200,20),IVQCH(200),IVSB(200)
      COMMON /BLKCN/HD,GK,DELTA,AS,XJS,CZ1,CZ2,CZ3,VLEN
      DATA AV,BV,P0,P1,P2,P3/9.125E-3,.79,0.,.0273,-.0199,7.699E-3/
      DATA DX3,DT3/4.234,7.2740/
      DATA VSTAT/3.0/
C
C   THERE ARE TWO CASES:
C      CASE 1 - SHIFTING DESTINATION FROM THE MAINLINE (IVPX=0)
C      CASE 2 - SHIFTING DESTINATION FROM A PREVIOUS SHIFT (IVPX=1)
C
      IF( IVPX(IDV).EQ.1 ) GO TO 100
C
C   CASE 1.
C
C   FIND INITIAL VELOCITY AND DISTANCE REMAINING TO OLD DEST.
C   LIMIT VELOCITY TO 10 M/S.
      IF( DTIN.GT.5.4 ) GO TO 10
      V = AV*( EXP(BV*DTIN) ) + 0.7
      DX = ( (P3*DTIN + P2)*DTIN +P1)*DTIN + P0
      GO TO 20
 10   TDL=DTIN-5.4
      V = AS*TDL + 1.35
      DX = AS*TDL*TDL + 0.78
      V = MIN1(V,10.001)
      IF( V.GE.10.0 ) DX = 50.72+10.0*(TDL-5.77)
C
C   COMPUTE TIME AND DISTANCE TO ACCEL (DECEL) TO VSTAT.
 20   DT1 = 0.
      DX1 = 0.
      VDL = VSTAT-V
      IF( VDL.LT.AS*AS/XJS ) GO TO 30
      DT1 =  ABS(VDL)/AS + AS/XJS
      DX1 = AS*( (VSTAT+V)/XJS + (VSTAT*VSTAT - V*V)/AS )/2.
C
C   COMPUTE TIME AND DISTANCE AT VSTAT.
 30   DX2 = DXN + DX - DX1 - DX3
      IF( DX2.GT.0. ) GO TO 40
      DT2 = 0.
      GO TO 50
 40   DT2 = DX2/VSTAT
C
C   COMPUTE EXTRA TIME NEEDED TO REACH NEW DESTINATION.
C   (NOTE: DT3 AND DX3 ARE THE TIME AND DISTANCE TO DECEL TO ZERO FROM
C          VSTAT USING THE PRECISION STOP CONTROLLER.)
 50   DTOUT = DT1 + DT2 + DT3 - DTIN - 1.5
```

```
      DTM = DXN/VSTAT
      IF( DTOUT.LT.DTM ) DTOUT = DTM
      GO TO 200
C
C    CASE 2.
C
C    ASSUME THE DXN WILL BE TRAVELED AT VSTAT.
 100  DTOUT = DXN/VSTAT
C
C    SET IVPX = 1 TO SIGNIFY XSHIFT OCCURRED.
 200  IVPX(IDV) = 1
C
C    LIMIT DTOUT TO 3*DTM TO ACCOUNT FOR ANOMOLOUS SITUATIONS.
      DTM = DXN*3.0/VSTAT
      IF( DTOUT.GT.DTM ) DTOUT=DTM
      RETURN
      END
```

APPENDIX B
ENERGY MODELS USED IN THE NETWORK MANAGEMENT SIMULATION

Energy consumption calculations are performed mostly within the subroutine VENGRY. Some cumulative statistics are tabulated for convenience in two other subroutines, PROUT and STATES. The sequence of calculations follow the flow shown in Fig. B-1. A velocity command is calculated for each vehicle and each time step in the subroutine CONSUB. This command is passed to VENGRY every fourth time step (every 1 sec) along with the old value of velocity. The acceleration (in ft/sec^2) and force (in ft-lb) required to achieve this change in velocity is calculated, assuming zero wind velocity and zero grade. The mechanical power in the motor required to produce this tractive force at the wheels is then determined. The drive train consists of a power conditioning unit with an efficiency of 0.98 and a gear set with an efficiency of 0.92. Two values are used for the efficiency of the motor: EM, a constant equal to 0.89; and EMV, a value that varies with speed according to the relationship

$$\eta(\nu) = \begin{cases} \eta_0 \left[1. - \left(\dfrac{\nu}{49.5} - 1\right)^2\right] & \nu \geq 5 \text{ ft/sec} \\ 0.2 & \nu < 5 \text{ ft/sec.} \end{cases}$$

The required power is converted in kW and is tabulated for each vehicle, each link, and the whole network. Energy consumption is also separately kept for vehicles that are decelerating, ΔVLO. This calculation assumes that each vehicle recovers 100% of its braking energy via regeneration. On a network basis this last value is kept by the program variable POWER. The variable POWERD is never decremented for regeneration. Consequently the difference POWERD - POWER returns the power saved from 100% recovery. Similarly the energy difference is kept via ETOTD - ETOT. This difference can be multiplied by any fraction to represent (constant) losses during regeneration.

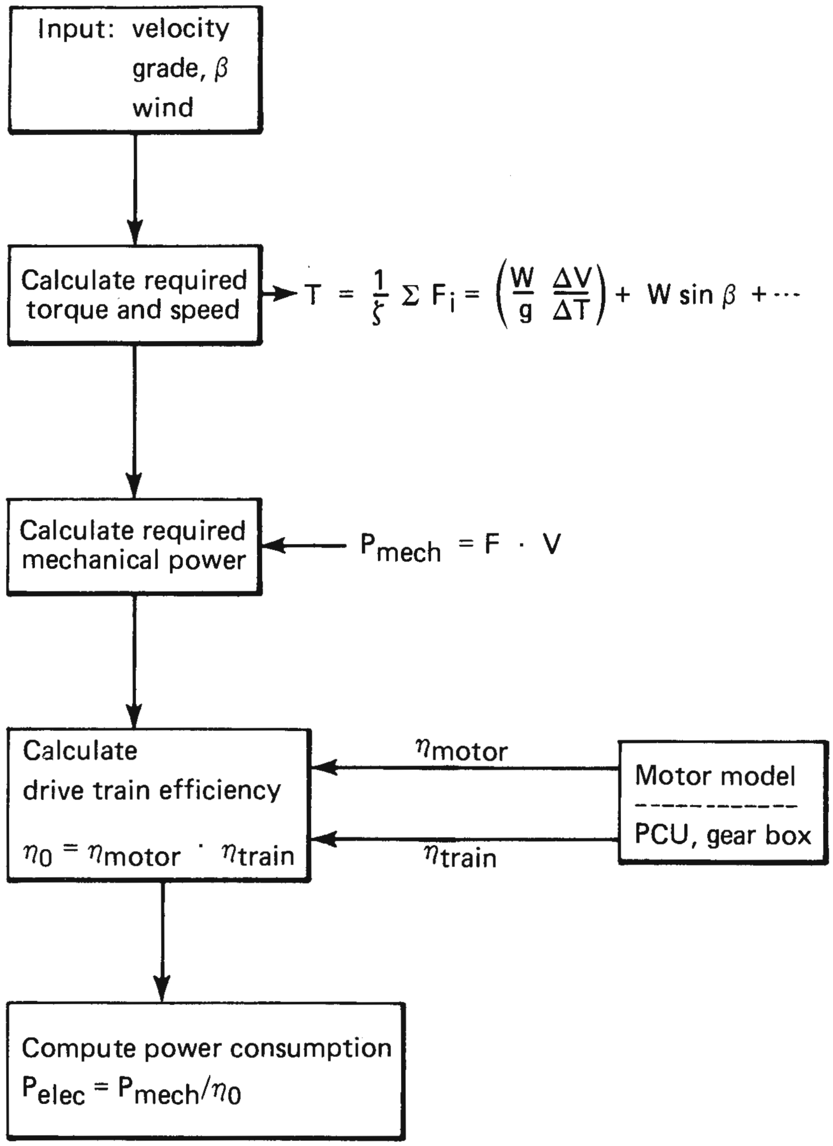

Fig. B-1 Overview of power calculations in subroutine VENGRY.

APPENDIX C
SAMPLE OUTPUT FROM SUBROUTINE PROUT

PRINTOUT: TIME = 3600.00

*** VEHICLE DATA ***

TOTAL VEHICLES = 103
IN VEHICLE-FOLLOWER MODE: 29 VEH; 28.16%
IN VELOCITY-COMMAND MODE: 41 VEH; 39.81%
IN STATION BERTHS: 33 VEH; 32.04%

TOTAL VEHICLE DISTANCE = 2604.93 KM (1618.71 MI)

TOT. VEH. DIST. THIS REPORT INTERVAL= 225.66 KM (140.23MI)
AVERAGE SPEED THIS REPORT INTERVAL= 0.05M/H (TRIP SPEED--INCLUDES STATION STOP DELAY)

*** STATION DATA ***

REJECTIONS: TOTAL NUMBER = 2 OUT OF 2770 TOTAL STATION ARRIVALS.

REJECTIONS BY STATION: 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

TOTAL ATTEMPTS BY STATION: 68 195 178 175 154 153 150 148 144 106 150 147 154 114 97 137 144 141 116 99

STATION QUEUE STATUS

	ENTRANCE QUEUE					DOCK QUEUE					EXIT QUEUE				
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	45
3	0	0	0	0	0	0	0	-8	46	0	0	0	0	0	-58
4	0	0	0	-28	-55	0	52	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	-65	51	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	19	0	0	0	0	50
7	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0
8	0	0	0	0	0	0	0	31	33	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	-26	49	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	98	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	-15	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	-21	0	0	0	0	0
13	0	0	0	0	0	0	0	0	-48	69	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	-74	37	0	0	0	0	0
16	0	0	0	0	0	0	0	0	-84	-13	0	0	0	0	-17
17	0	0	0	0	0	0	0	87	16	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	38	90	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	-42	0	0	0	0	-60
20	0	0	0	0	0	0	0	0	59	0	0	0	0	0	0

PRINTOUT: TIME = 3600.00

*** LINK DATA ***

LINK ID	NUMBER OF VEH	FLOW (VEH/INT)	DENSITY (VEH/M)	SPEED (M/S)	ENERGY (KWH)	ENERGY-D (KWH)	*	LINK ID	NUMBER OF VEH	FLOW (VEH/INT)	DENSITY (VEH/M)	SPEED (M/S)	ENERGY (KWH)	ENERGY-D (KWH)
1	0	6	0.0	15.00	2.390	2.390	*	46	2	36	0.0141	14.93	30.2154	33.6845
2	0	0	0.0	15.00	0.0	0.0	*	47	2	22	0.0105	14.80	27.6033	31.0318
3	0	7	0.0	15.00	18.143	18.143	*	48	0	15	0.0	15.00	43.0239	43.0329
4	0	7	0.0	15.00	4.473	4.473	*	49	0	37	0.0	15.00	8.6302	8.9784
5	1	8	0.0031	15.00	12.795	12.795	*	50	1	37	0.0100	14.05	17.9970	22.1869
6	0	8	0.0	15.00	1.612	3.556	*	51	3	52	0.0198	12.88	61.2592	61.4958
7	2	51	0.0165	13.56	48.287	48.988	*	52	2	44	0.0132	14.12	45.6172	50.6214
8	1	41	0.0110	15.00	24.598	28.606	*	53	2	30	0.0105	13.39	47.0266	51.8287
9	1	25	0.0049	10.16	39.316	44.870	*	54	0	16	0.0	15.00	35.2753	35.3035
10	1	19	0.0124	10.13	46.541	46.605	*	55	1	45	0.0849	11.89	3.6497	3.6610
11	1	43	0.0131	15.00	20.143	22.826	*	56	1	44	0.0100	15.00	31.3832	32.6027
12	2	23	0.0098	12.81	23.714	30.974	*	57	1	9	0.0028	15.00	26.0595	26.0595
13	0	19	0.0	15.00	33.904	34.499	*	58	0	0	0.0	15.00	0.0	0.0
14	1	43	0.0144	11.06	0.567	12.895	*	59	1	11	0.0124	0.0	30.4241	30.4241
15	0	25	0.0	10.00	19.954	22.596	*	60	1	10	0.0024	15.00	27.4218	27.4348
16	1	17	0.0270	0.38	21.761	21.761	*	61	1	10	0.0060	10.00	0.1524	7.0000
17	1	41	0.0177	10.00	14.366	14.456	*	62	0	1	0.0	10.00	0.7441	0.7819
18	1	41	0.0100	9.87	15.053	17.107	*	63	0	9	0.0	10.00	10.4735	10.4735
19	2	52	0.0220	7.00	23.172	28.366	*	64	0	10	0.0	10.00	2.9271	2.9279
20	3	51	0.0330	7.00	29.156	29.766	*	65	0	10	0.0	10.00	2.4161	3.3860
21	2	34	0.0250	5.72	19.082	20.526	*	66	2	30	0.0132	13.00	85.8978	88.9931
22	1	14	0.0527	2.25	7.800	7.800	*	67	1	16	0.0052	15.00	29.4710	32.0513
23	2	49	0.0151	6.77	41.814	41.880	*	68	1	13	0.0124	1.50	33.6429	33.6551
24	1	20	0.0033	7.75	76.958	79.286	*	69	0	30	0.0	15.00	9.3490	9.4164
25	0	7	0.0	15.00	8.248	9.893	*	70	0	10	0.0	15.00	3.5100	3.5115
26	0	13	0.0	15.00	38.066	38.218	*	71	1	20	0.0100	14.46	11.6042	13.1179
27	0	20	0.0	15.00	1.792	2.170	*	72	1	6	0.0100	15.00	3.6875	4.0795
28	1	20	0.0110	13.50	-4.024	6.847	*	73	1	16	0.0090	15.00	13.3858	13.4933
29	0	7	0.0	15.00	17.516	17.699	*	74	1	3	0.0052	15.00	6.1100	6.6228
30	1	15	0.0124	6.75	45.161	45.161	*	75	1	11	0.0124	8.25	37.4382	37.4382
31	3	21	0.0071	13.75	59.963	63.227	*	76	0	15	0.0	15.00	18.5407	18.8140
32	1	5	0.0052	15.00	12.303	13.686	*	77	0	16	0.0	12.00	25.8839	33.2042
33	1	14	0.0124	3.75	38.186	38.186	*	78	0	4	0.0	15.00	9.6611	10.0510
34	1	18	0.0049	15.00	24.430	25.947	*	79	0	13	0.0	15.00	36.5689	36.5689
35	0	18	0.0	10.00	-2.933	8.873	*	80	1	18	0.0038	15.00	29.1174	30.0514
36	0	5	0.0	15.00	21.909	22.327	*	81	0	8	0.0	15.00	9.1332	10.0027
37	0	13	0.0	15.00	36.885	36.889	*	82	0	10	0.0	15.00	30.1237	30.1280
38	0	20	0.0	15.00	19.146	20.783	*	83	1	18	0.0070	15.00	15.1265	15.8949
39	0	11	0.0	15.00	17.050	17.755	*	84	0	9	0.0	15.00	10.8320	11.8167
40	0	11	0.0	15.00	27.474	27.474	*	85	1	9	0.0124	1.88	32.2357	32.2357
41	1	21	0.0098	15.00	14.242	14.299	*	86	0	19	0.0	15.00	31.3608	31.6047
42	0	14	0.0	15.00	4.138	4.146	*	87	1	18	0.0100	11.84	6.4351	10.4890
43	4	35	0.0104	14.62	87.521	93.071	*	88	1	5	0.0024	15.00	18.4560	18.4580
44	0	24	0.0	15.00	29.487	32.530	*	89	0	14	0.0	15.00	7.8486	9.3819
45	1	13	0.0124	4.13	36.987	37.005	*	90	0	10	0.0	15.00	7.3719	7.5969

TOT NETWORK ENERGY = 2054.2278 KWH
TOTAL DIRECT NETWORK ENERGY = 2202.9321KWH

ENERGY CONSUMPTION THIS REPORT INTERVAL= 178.61 KWH
ENERGY PER VMT THIS REPORT INTERVAL= 1.27 KWH

PRINTOUT: TIME = 3600.00

*** ROUTE DATA ***

ROUTE ADHERENCE - INITIATION (BIN SIZE = 30 S)

RT#	1	2	3	4	5	6	7	8	9	10	##	RT#	1	2	3	4	5	6	7	8	9	10
1	0	0	0	2	6	1	2	1	0	1	##	41	0	0	0	1	6	3	2	1	1	0
2	0	0	0	3	3	2	4	0	0	1	##	42	0	0	1	3	4	2	1	3	0	0
3	0	0	0	1	9	1	1	0	1	1	##	43	0	0	0	3	7	4	0	0	0	1
4	0	0	0	3	5	3	1	0	1	1	##	44	0	0	3	3	1	6	1	0	0	1
5	1	0	0	1	8	2	1	0	0	1	##	45	0	0	1	3	7	2	1	1	0	0
6	0	0	0	1	7	4	1	0	0	1	##	46	0	0	0	5	3	7	0	0	0	0
7	0	0	1	2	7	3	1	1	0	0	##	47	0	1	0	4	5	3	2	0	0	0
8	0	0	1	2	4	4	0	0	0	1	##	48	1	0	0	1	8	4	1	0	0	0
9	0	0	0	3	9	2	0	0	0	1	##	49	1	0	0	1	7	3	0	1	0	1
10	0	0	1	2	6	4	1	0	1	0	##	50	1	0	0	1	7	4	0	0	0	1
11	0	0	2	1	5	5	2	0	0	0	##	51	0	1	0	2	7	2	1	0	0	1
12	0	0	2	1	5	6	1	0	0	0	##	52	0	1	0	1	6	6	0	0	0	1
13	0	0	1	5	3	4	2	0	0	0	##	53	0	0	0	1	9	4	0	0	0	1
14	1	0	1	2	7	2	2	1	0	0	##	54	0	0	1	2	7	2	1	2	0	0
15	0	0	1	3	3	4	1	1	0	1	##	55	0	0	3	0	5	5	2	0	0	0
16	0	0	0	3	5	3	1	1	0	1	##	56	0	1	0	1	5	1	3	0	0	1
17	0	0	0	1	7	5	0	0	0	1	##	57	0	1	1	2	3	4	1	0	0	1
18	0	0	1	2	5	4	1	0	0	1	##	58	0	0	1	2	5	3	0	0	0	2
19	0	0	0	0	9	2	1	0	0	1	##	59	0	0	0	4	10	9	1	0	1	1
20	0	0	0	2	4	3	3	0	0	1	##	60	0	1	0	1	7	5	0	0	0	1
21	0	0	0	1	5	3	2	0	1	1	##	61	0	1	0	0	9	3	2	0	0	0
22	0	0	1	2	2	2	4	0	1	1	##	62	0	0	0	0	0	0	0	0	0	0
23	0	0	3	3	6	0	2	0	0	1	##	63	0	0	0	0	0	0	0	0	0	0
24	0	0	1	4	4	4	1	0	1	0	##	64	0	0	0	0	0	0	0	0	0	0
25	0	0	3	0	7	3	1	1	0	0	##	65	0	0	0	0	0	0	0	0	0	0
26	0	0	3	1	6	4	2	0	0	0	##	66	0	0	0	0	0	0	0	0	0	0
27	0	0	0	2	7	3	1	1	0	0	##	67	0	0	0	0	0	0	0	0	0	0
28	0	0	1	1	6	5	1	1	0	0	##	68	0	0	0	0	0	0	0	0	0	0
29	0	0	0	4	6	3	1	1	0	0	##	69	0	0	0	0	0	0	0	0	0	0
30	0	0	1	3	7	3	0	0	1	0	##	70	0	0	0	0	0	0	0	0	0	0
31	1	0	0	2	7	3	2	0	0	0	##	71	0	0	0	0	0	0	0	0	0	0
32	0	0	1	0	8	3	1	0	0	1	##	72	0	0	0	0	0	0	0	0	0	0
33	0	0	1	0	7	4	1	0	0	1	##	73	0	0	0	0	0	0	0	0	0	0
34	0	0	1	1	6	4	1	0	0	1	##	74	0	0	0	0	0	0	0	0	0	0
35	0	0	0	3	5	2	3	0	1	0	##	75	0	0	0	0	0	0	0	0	0	0
36	0	0	0	2	5	4	2	0	1	0	##	76	0	0	0	0	0	0	0	0	0	0
37	0	0	1	2	3	5	2	0	1	0	##	77	0	0	0	0	0	0	0	0	0	0
38	0	0	0	2	7	5	0	0	1	0	##	78	0	0	0	0	0	0	0	0	0	0
39	0	0	0	1	7	4	1	0	0	1	##	79	0	0	0	0	0	0	0	0	0	0
40	0	0	0	3	5	1	4	0	0	1	##	80	0	0	0	0	0	0	0	0	0	0
TOT	6	7	39	118	361	211	77	17	13	35												

PRINTOUT: TIME = 3600.00

*** ROUTE DATA ***

ROUTF ADHERENCE - TERMINATION (BIN SIZE = 30 S)

RT#	1	2	3	4	5	6	7	8	9	10	##	RT#	1	2	3	4	5	6	7	8	9	10
1	0	0	0	4	2	1	3	0	0	1	##	41	0	0	0	3	1	3	2	1	0	1
2	0	0	1	2	3	2	2	1	0	1	##	42	0	0	1	3	2	3	0	3	0	1
3	0	0	0	1	6	3	1	0	1	1	##	43	0	0	0	3	6	3	0	0	0	1
4	0	0	0	2	5	3	0	0	0	2	##	44	0	1	1	2	3	4	1	0	0	1
5	0	0	1	0	6	2	0	1	0	2	##	45	0	0	1	3	4	3	1	0	0	1
6	0	0	1	1	3	4	2	0	0	1	##	46	0	0	2	1	6	2	1	0	0	1
7	0	0	0	2	4	6	0	0	0	1	##	47	0	1	0	2	5	2	2	0	0	1
8	0	0	0	4	1	3	1	0	0	1	##	48	0	1	0	1	7	3	1	0	0	1
9	0	0	1	2	6	3	0	0	0	1	##	49	0	1	0	1	6	3	1	0	0	1
10	0	1	0	4	1	2	3	0	0	1	##	50	0	1	0	0	8	3	0	0	0	1
11	0	0	1	2	4	4	1	0	0	1	##	51	0	1	0	2	4	3	1	0	0	1
12	0	0	1	1	7	0	3	0	0	1	##	52	0	0	1	1	5	4	1	0	0	1
13	0	0	0	5	2	3	2	0	0	1	##	53	0	0	0	1	8	4	0	0	0	1
14	0	0	1	3	5	1	0	3	0	1	##	54	0	0	2	2	4	2	1	1	0	1
15	0	0	1	1	5	2	1	0	0	1	##	55	0	0	1	4	3	1	4	0	0	1
16	0	0	0	1	6	2	2	0	0	1	##	56	0	0	1	2	3	2	2	0	0	1
17	0	0	0	1	6	3	0	0	0	2	##	57	0	1	1	2	2	4	1	0	0	1
18	0	0	0	3	4	3	0	1	0	2	##	58	0	0	1	2	5	3	0	0	0	2
19	0	0	0	0	7	2	1	0	0	1	##	59	0	0	0	5	8	8	1	0	2	1
20	0	0	0	0	4	5	1	1	0	1	##	60	0	0	0	1	7	4	0	0	1	1
21	0	0	0	2	3	0	2	2	1	1	##	61	0	1	0	0	8	2	2	0	1	0
22	0	0	1	1	4	1	2	2	0	1	##	62	0	0	0	0	0	0	0	0	0	0
23	0	0	3	3	3	1	2	0	0	1	##	63	0	0	0	0	0	0	0	0	0	0
24	0	0	1	3	5	3	0	0	0	1	##	64	0	0	0	0	0	0	0	0	0	0
25	0	0	2	4	3	3	1	0	0	1	##	65	0	0	0	0	0	0	0	0	0	0
26	0	1	0	2	5	3	1	1	0	1	##	66	0	0	0	0	0	0	0	0	0	0
27	0	0	1	2	3	2	1	2	0	1	##	67	0	0	0	0	0	0	0	0	0	0
28	0	0	2	0	3	5	1	0	1	1	##	68	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	8	3	1	0	0	1	##	69	0	0	0	0	0	0	0	0	0	0
30	0	0	0	1	8	2	1	0	0	1	##	70	0	0	0	0	0	0	0	0	0	0
31	0	0	0	1	6	3	2	0	0	1	##	71	0	0	0	0	0	0	0	0	0	0
32	0	0	2	0	5	2	2	0	0	1	##	72	0	0	0	0	0	0	0	0	0	0
33	0	0	1	0	5	3	2	0	0	1	##	73	0	0	0	0	0	0	0	0	0	0
34	0	0	0	2	6	3	1	0	0	1	##	74	0	0	0	0	0	0	0	0	0	0
35	0	0	0	1	6	2	2	0	0	1	##	75	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	6	3	1	1	0	1	##	76	0	0	0	0	0	0	0	0	0	0
37	0	0	0	3	2	5	1	1	0	1	##	77	0	0	0	0	0	0	0	0	0	0
38	0	0	0	1	7	3	0	0	1	1	##	78	0	0	0	0	0	0	0	0	0	0
39	0	0	0	2	6	3	1	0	0	1	##	79	0	0	0	0	0	0	0	0	0	0
40	0	0	0	3	3	2	2	0	1	1	##	80	0	0	0	0	0	0	0	0	0	0
TOT	0	10	33	111	289	172	71	21	9	65	741											

PRINTOUT: TIME = 3600.00

*** HEADWAY DATA ***

HEADWAY DISTRIBUTION BY LINK (BIN SIZE = 5 S)

LNK	1	2	3	4	5	6	7	8	9	10	##	LNK	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0	##	46	1	1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	##	47	2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	##	48	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	##	49	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0	0	##	50	1	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	##	51	3	0	0	0	0	0	0	0	0	0
7	2	0	0	0	0	0	0	0	0	0	##	52	1	1	0	0	0	0	0	0	0	0
8	0	0	1	0	0	0	0	0	0	0	##	53	2	0	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0	##	54	0	0	0	0	0	0	0	0	0	0
10	0	1	0	0	0	0	0	0	0	0	##	55	1	0	0	0	0	0	0	0	0	0
11	0	0	1	0	0	0	0	0	0	0	##	56	0	0	0	0	0	0	0	0	0	1
12	2	0	0	0	0	0	0	0	0	0	##	57	0	0	0	0	0	0	0	0	0	1
13	0	0	0	0	0	0	0	0	0	0	##	58	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	1	##	59	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	##	60	0	0	0	0	0	0	1	0	0	0
16	0	0	0	0	0	0	0	0	0	2	##	61	0	0	0	0	0	0	0	0	0	1
17	0	1	0	0	0	0	0	0	0	0	##	62	0	0	0	0	0	0	0	0	0	0
18	1	0	0	0	0	0	0	0	0	0	##	63	0	0	0	0	0	0	0	0	0	0
19	1	0	1	0	0	0	0	0	0	0	##	64	0	0	0	0	0	0	0	0	0	0
20	3	0	0	0	0	0	0	0	0	0	##	65	0	0	0	0	0	0	0	0	0	0
21	2	0	0	0	0	0	0	0	0	0	##	66	2	0	0	0	0	0	0	0	0	0
22	0	0	0	0	1	0	0	0	0	0	##	67	0	0	1	0	0	0	0	0	0	0
23	1	0	1	0	0	0	0	0	0	0	##	68	1	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	1	##	69	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	##	70	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	##	71	1	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	##	72	0	1	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	1	##	73	0	0	0	0	0	0	0	0	0	1
29	0	0	0	0	0	0	0	0	0	0	##	74	0	0	0	0	0	0	0	0	0	1
30	0	0	0	1	0	0	0	0	0	0	##	75	1	0	0	0	0	0	0	0	0	0
31	1	1	1	0	0	0	0	0	0	0	##	76	0	0	0	0	0	0	0	0	0	0
32	0	0	0	1	0	0	0	0	0	0	##	77	0	0	0	0	0	0	0	0	0	0
33	1	0	0	0	0	0	0	0	0	0	##	78	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	1	##	79	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	##	80	0	0	1	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	##	81	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	##	82	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	##	83	0	0	0	0	0	0	0	0	0	1
39	0	0	0	0	0	0	0	0	0	0	##	84	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	##	85	0	0	0	0	0	0	0	0	0	1
41	0	1	0	0	0	0	0	0	0	0	##	86	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	##	87	1	0	0	0	0	0	0	0	0	0
43	1	2	0	0	0	0	0	0	0	1	##	88	0	0	0	0	0	0	0	0	0	1
44	0	0	0	0	0	0	0	0	0	0	##	89	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	1	0	0	0	##	90	0	0	0	0	0	0	0	0	0	0
TOT	33	9	7	3	1	0	2	0	0	15												

PRINTOUT: TIME = 3600.00

*** STATION TIME DATA ***

TIME DISTRIBUTION BY STATION (BIN SIZE = 30.000 S)

ST#	1	2	3	4	5	6	7	8	9	10
1	0	68	0	0	0	0	0	0	0	0
2	0	96	75	21	0	0	0	0	0	0
3	0	173	2	0	0	0	0	0	0	0
4	27	172	0	0	0	0	0	0	0	0
5	0	150	2	0	0	0	0	0	0	0
6	0	137	14	0	0	0	0	0	0	0
7	24	148	1	0	0	0	0	0	0	0
8	0	126	20	0	0	0	0	0	0	0
9	0	142	0	0	0	0	0	0	0	0
10	0	104	1	0	0	0	0	0	0	0
11	0	137	12	0	0	0	0	0	0	0
12	26	145	1	0	0	0	0	0	0	0
13	0	139	13	0	0	0	0	0	0	0
14	0	114	0	0	0	0	0	0	0	0
15	0	95	0	0	0	0	0	0	0	0
16	0	120	14	0	0	0	0	0	0	0
17	0	141	1	0	0	0	0	0	0	0
18	0	139	0	0	0	0	0	0	0	0
19	0	114	0	0	0	0	0	0	0	0
20	26	98	0	0	0	0	0	0	0	0
TOT	103	2558	156	21	0	0	0	0	0	0

PRINTOUT: TIME = 3600.00

*** O/D TRIP TIME DATA ***
TIME DISTRIBUTION BY STATION (BIN SIZE = 60.000 S)

# 1	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	13	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	12	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	12	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	13	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	13	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0

# 2	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	156	20	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	10	4	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0

# 3	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	173	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0

# 4	1	2	3	4	5	6	7	8	9	10
1	3	0	0	0	0	0	0	0	0	0
2	3	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	86	0	0	0	0	0	0	0	0	0
6	17	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	14	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	3	0	0	0	0	0	0	0	0	0
11	13	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	17	0	0	0	0	0	0	0	0	0
14	1	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	1	0	0	0	0	0	0	0	0	0
17	16	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	17	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0

PRINTOUT: TIME = 3600.00

*** O/D TRIP TIME DATA ***
TIME DISTRIBUTION BY STATION (BIN SIZE = 60.000 S)

# 5	1	2	3	4	5	6	7	8	9	10	*****	# 6	1	2	3	4	5	6	7	8	9	10
1	23	0	0	0	0	0	0	0	0	0	*****	1	0	0	0	0	0	0	0	0	0	0
2	12	0	0	0	0	0	0	0	0	0	*****	2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	*****	3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	*****	4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	*****	5	0	0	0	0	0	0	0	0	0	0
6	24	0	0	0	0	0	0	0	0	0	*****	6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	*****	7	150	0	0	0	0	0	0	0	0	0
8	13	0	0	0	0	0	0	0	0	0	*****	8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	*****	9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	*****	10	0	0	0	0	0	0	0	0	0	0
11	12	0	0	0	0	0	0	0	0	0	*****	11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	*****	12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	*****	13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	*****	14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	*****	15	0	0	0	0	0	0	0	0	0	0
16	36	0	0	0	0	0	0	0	0	0	*****	16	0	0	0	0	0	0	0	0	0	0
17	13	0	0	0	0	0	0	0	0	0	*****	17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	*****	18	0	0	0	0	0	0	0	0	0	0
19	13	0	0	0	0	0	0	0	0	0	*****	19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	*****	20	0	0	0	0	0	0	0	0	0	0

# 7	1	2	3	4	5	6	7	8	9	10	*****	# 8	1	2	3	4	5	6	7	8	9	10
1	3	0	0	0	0	0	0	0	0	0	*****	1	0	0	0	0	0	0	0	0	0	0
2	14	0	0	0	0	0	0	0	0	0	*****	2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	*****	3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	*****	4	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	*****	5	0	0	0	0	0	0	0	0	0	0
6	3	0	0	0	0	0	0	0	0	0	*****	6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	*****	7	0	0	0	0	0	0	0	0	0	0
8	104	0	0	0	0	0	0	0	0	0	*****	8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	*****	9	144	0	0	0	0	0	0	0	0	0
10	16	0	0	0	0	0	0	0	0	0	*****	10	0	0	0	0	0	0	0	0	0	0
11	4	0	0	0	0	0	0	0	0	0	*****	11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	*****	12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	*****	13	0	0	0	0	0	0	0	0	0	0
14	4	0	0	0	0	0	0	0	0	0	*****	14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	*****	15	0	0	0	0	0	0	0	0	0	0
16	2	0	0	0	0	0	0	0	0	0	*****	16	0	0	0	0	0	0	0	0	0	0
17	15	0	0	0	0	0	0	0	0	0	*****	17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	*****	18	0	0	0	0	0	0	0	0	0	0
19	3	0	0	0	0	0	0	0	0	0	*****	19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	*****	20	0	0	0	0	0	0	0	0	0	0

PRINTOUT: TIME = 3600.00

*** O/D TRIP TIME DATA ***
TIME DISTRIBUTION BY STATION (BIN SIZE = 60.000 S)

# 9	1	2	3	4	5	6	7	8	9	10	*****
1	0	0	0	0	0	0	0	0	0	0	*****
2	12	0	0	0	0	0	0	0	0	0	*****
3	0	0	0	0	0	0	0	0	0	0	*****
4	0	0	0	0	0	0	0	0	0	0	*****
5	0	0	0	0	0	0	0	0	0	0	*****
6	12	0	0	0	0	0	0	0	0	0	*****
7	0	0	0	0	0	0	0	0	0	0	*****
8	0	0	0	0	0	0	0	0	0	0	*****
9	0	0	0	0	0	0	0	0	0	0	*****
10	61	0	0	0	0	0	0	0	0	0	*****
11	13	0	0	0	0	0	0	0	0	0	*****
12	0	0	0	0	0	0	0	0	0	0	*****
13	0	0	0	0	0	0	0	0	0	0	*****
14	13	0	0	0	0	0	0	0	0	0	*****
15	0	0	0	0	0	0	0	0	0	0	*****
16	13	0	0	0	0	0	0	0	0	0	*****
17	13	0	0	0	0	0	0	0	0	0	*****
18	0	0	0	0	0	0	0	0	0	0	*****
19	0	0	0	0	0	0	0	0	0	0	*****
20	0	0	0	0	0	0	0	0	0	0	*****

#10	1	2	3	4	5	6	7	8	9	10
1	13	0	0	0	0	0	0	0	0	0
2	12	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	11	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	14	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	12	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0
17	38	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0

#11	1	2	3	4	5	6	7	8	9	10	*****
1	0	0	0	0	0	0	0	0	0	0	*****
2	0	0	0	0	0	0	0	0	0	0	*****
3	0	0	0	0	0	0	0	0	0	0	*****
4	0	0	0	0	0	0	0	0	0	0	*****
5	0	0	0	0	0	0	0	0	0	0	*****
6	0	0	0	0	0	0	0	0	0	0	*****
7	0	0	0	0	0	0	0	0	0	0	*****
8	0	0	0	0	0	0	0	0	0	0	*****
9	0	0	0	0	0	0	0	0	0	0	*****
10	0	0	0	0	0	0	0	0	0	0	*****
11	0	0	0	0	0	0	0	0	0	0	*****
12	147	0	0	0	0	0	0	0	0	0	*****
13	0	0	0	0	0	0	0	0	0	0	*****
14	0	0	0	0	0	0	0	0	0	0	*****
15	0	0	0	0	0	0	0	0	0	0	*****
16	0	0	0	0	0	0	0	0	0	0	*****
17	0	0	0	0	0	0	0	0	0	0	*****
18	0	0	0	0	0	0	0	0	0	0	*****
19	0	0	0	0	0	0	0	0	0	0	*****
20	0	0	0	0	0	0	0	0	0	0	*****

#12	1	2	3	4	5	6	7	8	9	10
1	13	0	0	0	0	0	0	0	0	0
2	16	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	17	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	2	0	0	0	0	0	0	0	0	0
11	4	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	76	0	0	0	0	0	0	0	0	0
14	4	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	16	0	0	0	0	0	0	0	0	0
17	4	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	15	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0

PRINTOUT: TIME = 3600.00

*** O/D TRIP TIME DATA ***
TIME DISTRIBUTION BY STATION (BIN SIZE = 60.000 S)

#13	1	2	3	4	5	6	7	8	9	10	*****	#14	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0	*****	1	0	0	0	0	0	0	0	0	0	0
2	71	0	0	0	0	0	0	0	0	0	*****	2	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	*****	3	0	0	0	0	0	0	0	0	0	0
4	2	0	0	0	0	0	0	0	0	0	*****	4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	*****	5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	*****	6	14	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	*****	7	0	0	0	0	0	0	0	0	0	0
8	12	0	0	0	0	0	0	0	0	0	*****	8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	*****	9	0	0	0	0	0	0	0	0	0	0
10	11	0	0	0	0	0	0	0	0	0	*****	10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	*****	11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	*****	12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	*****	13	0	0	0	0	0	0	0	0	0	0
14	37	0	0	0	0	0	0	0	0	0	*****	14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	*****	15	97	0	0	0	0	0	0	0	0	0
16	13	0	0	0	0	0	0	0	0	0	*****	16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	*****	17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	*****	18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	*****	19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	*****	20	0	0	0	0	0	0	0	0	0	0

#15	1	2	3	4	5	6	7	8	9	10	*****	#16	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0	*****	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	*****	2	13	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	*****	3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	*****	4	0	0	0	0	0	0	0	0	0	0
5	40	0	0	0	0	0	0	0	0	0	*****	5	13	0	0	0	0	0	0	0	0	0
6	13	0	0	0	0	0	0	0	0	0	*****	6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	*****	7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	*****	8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	*****	9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	*****	10	0	0	0	0	0	0	0	0	0	0
11	13	0	0	0	0	0	0	0	0	0	*****	11	51	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	*****	12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	*****	13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	*****	14	13	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	*****	15	0	0	0	0	0	0	0	0	0	0
16	27	0	0	0	0	0	0	0	0	0	*****	16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	*****	17	40	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	*****	18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	*****	19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	*****	20	0	0	0	0	0	0	0	0	0	0

PRINTOUT: TIME = 3600.00

*** O/D TRIP TIME DATA ***
TIME DISTRIBUTION BY STATION (BIN SIZE = 60.000 S)

#17	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	141	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0

#18	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	14	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	13	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	13	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	15	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	14	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	65	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0

#19	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	14	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20	99	0	0	0	0	0	0	0	0	0

#20	1	2	3	4	5	6	7	8	9	10
1	13	0	0	0	0	0	0	0	0	0
2	13	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	14	0	0	0	0	0	0	0	0	0
6	17	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	5	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	32	0	0	0	0	0	0	0	0	0
14	17	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	2	0	0	0	0	0	0	0	0	0
17	5	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	3	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0

TOT	2744	24	0	0	0	0	0	0	0	0

PRINTOUT: TIME = 3600.00

VEH ID	VEH ENERGY	VEH POWERD		VEH ID	VEH ENERGY	VEH POWERD		VEH ID	VEH ENERGY	VEH POWERD		VEH ID	VEH ENERGY	VEH POWERD	
1	0.2270E+02	0.9430E+05	*	2	0.2284E+02	0.9512E+05	*	3	0.2451E+02	0.9931E+05	*	4	0.2300E+02	0.9660E+05	*
5	0.2414E+02	0.9967E+05	*	6	0.2417E+02	0.9987E+05	*	7	0.2179E+02	0.9104E+05	*	8	0.2167E+02	0.9090E+05	*
9	0.2184E+02	0.9218E+05	*	10	0.2363E+02	0.9751E+05	*	11	0.2202E+02	0.9202E+05	*	12	0.2453E+02	0.9954E+05	*
13	0.2307E+02	0.9679E+05	*	14	0.2402E+02	0.9858E+05	*	15	0.2286E+02	0.9442E+05	*	16	0.2230E+02	0.9286E+05	*
17	0.2187E+02	0.9035E+05	*	18	0.2264E+02	0.9270E+05	*	19	0.2364E+02	0.9669E+05	*	20	0.2230E+02	0.9290E+05	*
21	0.2215E+02	0.9301E+05	*	22	0.2193E+02	0.9084E+05	*	23	0.2155E+02	0.8970E+05	*	24	0.2288E+02	0.9516E+05	*
25	0.2136E+02	0.8944E+05	*	26	0.2163E+02	0.9025E+05	*	27	0.2116E+02	0.8755E+05	*	28	0.2151E+02	0.8912E+05	*
29	0.2187E+02	0.9051E+05	*	30	0.2242E+02	0.9170E+05	*	31	0.2197E+02	0.9120E+05	*	32	0.2377E+02	0.9684E+05	*
33	0.2256E+02	0.9249E+05	*	34	0.2115E+02	0.8776E+05	*	35	0.2132E+02	0.8870E+05	*	36	0.2270E+02	0.9347E+05	*
37	0.2180E+02	0.9041E+05	*	38	0.2220E+02	0.9190E+05	*	39	0.2088E+02	0.8759E+05	*	40	0.2313E+02	0.9434E+05	*
41	0.1991E+02	0.8453E+05	*	42	0.2288E+02	0.9365E+05	*	43	0.2323E+02	0.9436E+05	*	44	0.2262E+02	0.9325E+05	*
45	0.2125E+02	0.8858E+05	*	46	0.2215E+02	0.9033E+05	*	47	0.2202E+02	0.9118E+05	*	48	0.2242E+02	0.9198E+05	*
49	0.2119E+02	0.8977E+05	*	50	0.2193E+02	0.9011E+05	*	51	0.1958E+02	0.8273E+05	*	52	0.2188E+02	0.9047E+05	*
53	0.2180E+02	0.9102E+05	*	54	0.2140E+02	0.8913E+05	*	55	0.2188E+02	0.8972E+05	*	56	0.2085E+02	0.8622E+05	*
57	0.2097E+02	0.8718E+05	*	58	0.2037E+02	0.8504E+05	*	59	0.2178E+02	0.8934E+05	*	60	0.2136E+02	0.8831E+05	*
61	0.2089E+02	0.8701E+05	*	62	0.2012E+02	0.8426E+05	*	63	0.2371E+02	0.9742E+05	*	64	0.2130E+02	0.8845E+05	*
65	0.1922E+02	0.8175E+05	*	66	0.2129E+02	0.8933E+05	*	67	0.2113E+02	0.8757E+05	*	68	0.2148E+02	0.8796E+05	*
69	0.2220E+02	0.9192E+05	*	70	0.2067E+02	0.8635E+05	*	71	0.2112E+02	0.8832E+05	*	72	0.1967E+02	0.8194E+05	*
73	0.1913E+02	0.8028E+05	*	74	0.2013E+02	0.8369E+05	*	75	0.2106E+02	0.8730E+05	*	76	0.1944E+02	0.8203E+05	*
77	0.2035E+02	0.8478E+05	*	78	0.2139E+02	0.8750E+05	*	79	0.1960E+02	0.8345E+05	*	80	0.2054E+02	0.8591E+05	*
81	0.2065E+02	0.8567E+05	*	82	0.2109E+02	0.8645E+05	*	83	0.2031E+02	0.8474E+05	*	84	0.1869E+02	0.7869E+05	*
85	0.1882E+02	0.7569E+05	*	86	0.1968E+02	0.8239E+05	*	87	0.2004E+02	0.8641E+05	*	88	0.1948E+02	0.8119E+05	*
89	0.1927E+02	0.8028E+05	*	90	0.1981E+02	0.8249E+05	*	91	0.2000E+02	0.8323E+05	*	92	0.1958E+02	0.8213E+05	*
93	0.1949E+02	0.8159E+05	*	94	0.2104E+02	0.8719E+05	*	95	0.1966E+02	0.8181E+05	*	96	0.1922E+02	0.8001E+05	*
97	0.1992E+02	0.7964E+05	*	98	0.1965E+02	0.8095E+05	*	99	0.2117E+02	0.8660E+05	*	100	0.2063E+02	0.8523E+05	*
101	0.2037E+02	0.8419E+05	*	102	0.2079E+02	0.8581E+05	*	103	0.1867E+02	0.7750E+05	*	104	0.0	0.0	*

ENERGY USED WITH A SPEED DEPENDENT EFFCIENCY 0.2540E+04

APPENDIX D
DEFINITION OF NETWORK MANAGEMENT SIMULATION VARIABLES

LINK VARIABLES

I = link ID

ALLL(I)	link length
ALMV(I)	link maximum velocity
ALXX(I)	x-coordinate of link head
ALYY(I)	y-coordinate of link head
LNEL(I)	exit link ID
LNFV(I)	ID of first vehicle on link
LNLI(I)	(dependent on link type T)
	T = 2 parallel data link ID
	= 3 station ID
	= 4 station egress link ID
	= 5 station bypass link ID
	= 6 secondary exit link
LNVL(I)	ID of last vehicle on link
LNRN(I)	route number
LNTV(I)	total number of vehicles on link
LNTY(I)	type
	= 1 nominal
	= 2 parallel
	= 3 station entry
	= 4 station bypass
	= 5 station egress
	= 6 divert

STATION VARIABLES

I = station ID

ISQS(I,J,K)	queue status of *k*th berth		
	J	= 1	entrance
		= 2	dock
		= 3	exit
	K	= 0	unoccupied
		= 1	occupied
		= 2	reserved
ISRN(I)	number of rejections		
ISTA(I,J)	station queue characteristics		
	J	= 1	entrance queue berth number
		= 2	dock queue berth number
		= 3	exit queue berth number
		= 4	entrance queue vehicle number
		= 5	dock queue vehicle number
		= 6	exit queue vehicle number
		= 7	number of passengers waiting in station
		= 8	trip number waiting in station
ISTL(I,J)	*j*th trip ID in station		
ISTX(I)	exit link ID		

SYSTEM MANAGEMENT VARIABLES

I = route ID
J = station ID

IDSP	pointer to elements of IVDSP
IRTSCH(I,J)	
	(I,1) = assigned service interval for route I (SEC)
	(I,2) = clock time of last vehicle assignment to route I
	(I,3) = clock time of last vehicle arrival at 1st station of the last station set by route I
IRTSTC(I,J)	
	= 0 if route I does not serve station J
	= 1 if route I serves station J
	= 2 if station J originates route I
	= -2 if J is first station of last set served by I
IVDSP(K)	ID of *k*th station from which vehicles are initially dispatched
NDSP	number of stations used for dispatch
NER	number of empty redistribution routes
NFLT	total desired number of vehicles
NSEQ(I,K)	identifies sequence of station served by route I; K = 1 is first station served, K = 2 is second station served, etc.
NSR	number of service routes
NSRT(I)	number of stations served by route I
NVDS(I)	destination station for *i*th vehicle dispatch
RTFRQ(I)	assigned operating frequency for route I (veh/hr)

TRIP VARIABLES

I = trip ID

ITDS(I)	destination station
ITOS(I)	origin station
ITPS(I)	party size
ITQC(I)	= ID of next trip in some queue as this trip = 0 if trip is not currently a member of a list
ITRD(I)	destination ID if trip is rejected at desired destination; 0 otherwise
ITSB(I)	station boarding number assigned to trip = 0 unassigned = route number
ITTD(I)	intermediate destination if a transfer is required
ITUB(I)	number of unsuccessful boardings
ITVI(I)	ID of vehicle boarded or reserved; special code for multi-route classification
TRAT(I)	station arrival time
TRBT(I)	boarding time
TRCT(I)	cumulative out-of-vehicle time (to indicate transfer time)
TRPT(I)	platform arrival time

VEHICLE VARIABLES

I = vehicle ID

IVCM(I)	control mode
	= 1 regulation = 2 velocity-command = 3 discrete-event
IVCS(I,J)	
	j = index to sequence through station stops on current list = -1 for all j greater than the number of stops on current route = available capacity on vehicle on departure from stop j
IVLS(I)	current link/station ID
IVLV(I)	lead vehicle ID
IVLV2(I)	companion lead vehicle ID
IVND(I)	next destination station ID
IVPX(I)	number of passengers on board
IVQCH(I)	
	= ID of next vehicle in same queue as vehicle = 0 if vehicle I is not in a queue
IVRF(I)	reject flag
	= 0 no station rejection = 1 station rejected
IVSE(I)	statin queue berth number
IVSR(I)	service route ID
IVSS(I)	station status
	= 0 not in station = 1 entrance queue = 2 dock queue = 3 dock queue, dwell ended = 4 exit queue
IVTP(I)	tail pointer of trip list
IVTV(I)	trailing vehicle ID
VACC(I)	vehicle acceleration
VENR(I)	vehicle energy
VDIST(I)	cumulative distance traveled by vehicle
VPOS(I)	vehicle position relative to the link head
VVEL(I)	vehicle velocity