

UC Berkeley

Working Papers

Title

Users' Guide and Design Description SmartPath Simulator - Version MOU62

Permalink

<https://escholarship.org/uc/item/6qw9t7vf>

Authors

Hongola, Bruce

Tsao, Jacob

Hall, Randolph

Publication Date

1993-08-01

This paper has been mechanically scanned. Some errors may have been inadvertently introduced.

**CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY**

Users' Guide and Design Description
SmartPath Simulator — Version MOU62

**Bruce Hongola
Jacob Tsao
Randolph Hall**

UCB-ITS-PWP-93-8

This work was performed **as** part of the California PATH Program **of** the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department of Transportation, Federal Highway Administration.

The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views **or** policies of the State of California. This report does not constitute a standard, specification, **or** regulation.

AUGUST 1993

ISSN 1055-1417

Users' Guide and Design Description

for the SmartPath Simulator Version MOU62

Contents

1	Introduction	1
2	Users' Guide	7
	2.1 Program Compilation	7
	2.2 Data Base Initialization	9
	2.3 Program Execution	9
	2.4 Data File Description	10
	2.4.1 SmartPath Input File 1	10
	2.4.1.1 File 1 Items Not Used	11
	2.4.2 SmartPath Input File 2	12
	2.4.3 SmartPath Output File "ahs.sum"	14
	2.4.4 Simrep Output File	15
3	Data File Listing	16
	3.1 Example of Input File 1	18
	3.2 Example of Input File 2	23
	3.3 Example of Output File "ahs.sum"	25
	3.4 Example of Simrep Output File	33
	References	36
A.	Appendix	37
	A.1 Design Description	37
	A.2 Design Notes	46
	A.3 Input Parameter Notes	52
	A.4 Source Code and Data Files	53
	A.5 Glossary	55

Section 1. Introduction

This document contains the user instructions and software design description for Version MOU62 of the SmartPath Simulator. The baseline version of the simulator including user instructions is described in [1].

Objective: AHS Performance Study

The primary motivation for the concept of Automated Highway Systems (AHS) is the desire to significantly increase the performance of future highways without requiring a significant amount of extra right-of-way. Besides the automation technology supporting AHS, crucial performance categories of AHS operation include safety, throughput and human factors. Assuming technological feasibility and driver/public acceptability, MOU62 concentrates on the performance category of throughput. It studies the throughput performance of different AHS, where safety is represented as parametric input to the throughput model.

Throughput Performance Measures

The sustainable throughput performance of an AHS hinges upon how it is configured and operated. An operating strategy consists of a collection of operating rules. Major categories of operating rules include: access, lane flow, lane selection, lane change and egress. These rules, together with the AHS configuration, would determine the throughput of the AHS, which is measured by:

- traffic flow in each lane, throughout the **AHS** segment
- speed distribution, in particular the average speed and the standard deviation
- success rate of vehicles reaching the desired destinations, lanes or exits.

Scope

A major configuration option is the erection of lane barriers and the concomitant openings for lane changes. A major **AHS** operating rule that has extensive effect on all aspects of **AHS** operation is the vehicle-following rule governing the spacing between any two adjacent vehicles in a common lane. We limit the scope of our study to the effect of the vehicle-following rule, platooning or free-agent, as well as the lane barriers on three measures.

Approach

We study the effect by simulating a segregated **AHS** segment with existing highway traffic flowing in at the beginning of the segment and traffic in-flow and out-flow at the entrances and exits respectively. To achieve this and to gauge the throughput performance in **terms** of the three major measures, we make several major modifications to SmartPath.

Four Scenarios

We study four scenarios with different combinations of lane-keeping rule and barrier option: (i) platooning without barriers, (ii) platooning with barriers, (iii) free-agent lane-keeping without barriers, and (iv) free-agent lane-keeping with barriers.

AHS Configuration and Operation

For a detailed description of SmartPath capabilities, **see** [1]. We simulate the free-agent lane-keeping rule by setting the maximum platoon size to 1. For a successful lane change with lane barriers, the lane-changing vehicle and the receiving gap in the neighboring lane have to align with the gate. The new SmartPath does not contain any new logic to ensure such alignments. (Such alignment can be achieved with greater efficiency through coordination between the involved vehicles and platoons.)

Focus

This document focuses on the differences between this version and the baseline (original) version of SmartPath. This is done to avoid redundancy in describing the functional capabilities and design.

Retained Major SmartPath Capabilities

The following is a summary of general simulator capabilities and modeling assumptions:

- Movement of individual vehicles is modeled.
- Vehicle-to-vehicle communication within platoons is modeled.
- Platoons of vehicles can be formed in the automated lanes.
- Communication between platoon leaders is modeled.
- Vehicle-to-roadside sensor communication is assumed to exist.
- The simulation environment **CSIM** is used (Reference 3).
- Only free agent vehicles are allowed in transition lanes.
- Platoon maneuvers merge, split and change lane are modeled as described in Reference 1.
- The system control layers within the IVHS systems architecture (Ref. 5) are modeled as described in Reference 1.

Enhancements and Motivation

We summarize the enhancements of the new SmartPath as follows. Motivation is also provided.

- Platoons of vehicles are generated in automated lanes at the beginning of the highway segment. Furthermore, the volume and speed of the platoons depend on the traffic condition at the beginning of the segment. This way, the simulator simulates any arbitrary segment of the AHS, instead of the very beginning of an AHS.
- Vehicles entering from the beginning of the AHS segment that need to exit are determined randomly based on a specified percentage of the traffic entering from the beginning. An exit is randomly assigned to each of the exiting vehicles. Similarly, a

destination (automated) lane is randomly assigned to each vehicle that does not need to exit, based upon a matrix of specified percentages for vehicle routing in the segment. Note that vehicles that enter through the entrance on-ramps are assumed to have a destination beyond the end of the segment. This feature is needed to study the effect of the amount of lane changes on the performance of the AHS.

- Lane changes can be restricted to specified start and end distances (i.e., gates) along the automated lanes.
- The following statistics are generated to measure system performance and to help the user better understand the actual traffic flow:
 - Number and percentage of vehicles reaching each assigned automated lane and exit
 - Average speed and standard deviation of speed
 - Traffic flow and density
 - Statistics on lane change times and distances
 - Distribution of platoon sizes
 - Difference between the average and the target speed.

Software Environment

The program runs on a Sun Sparc Workstation.

Organization of Document

Section 2 contains the users' guide, Section 3 contains an example run, and the Appendix describes the software design. Source code, notes on parameters, and a brief glossary are included in the Appendix.

Section 2. Users' Guide

This section provides an overview of how to initialize the data base and compile and execute the programs. Figure 2-1 shows the data flow involving the user-set input files ("carout" and "simrep") and the output files ("carout" is the SmartPath executable program and "simrep" is a report program that produces lane change statistics).

If the user wishes to make software changes, the programs must be re-compiled as described in Section 2.1. If multiple simulation runs are desired, each requiring different input, then each run requires its own version of the input files identified in Section 2.2. Also, the corresponding output files identified in Section 2.3 must have unique names for each simulation run so that data generated by a previous run is not overwritten.

Section 2.1 Program Compilation

If the executables "carout" and "simrep" already exist, then program compilation is not necessary. Appendix A.4 lists the names of files needed to create the executables.

- The makefile is used to create "carout". There are two versions of makefiles, one with debug on and one without. The command "make" is used to execute the makefile.
- csim.h, csimdef.h and libcsim.a are to be placed in a separate library directory unless the makefile is changed.
- The following compilation command is used to create the report program "simrep" from the source code:

```
cc -o simrep simrep.c db_init.c -lm
```

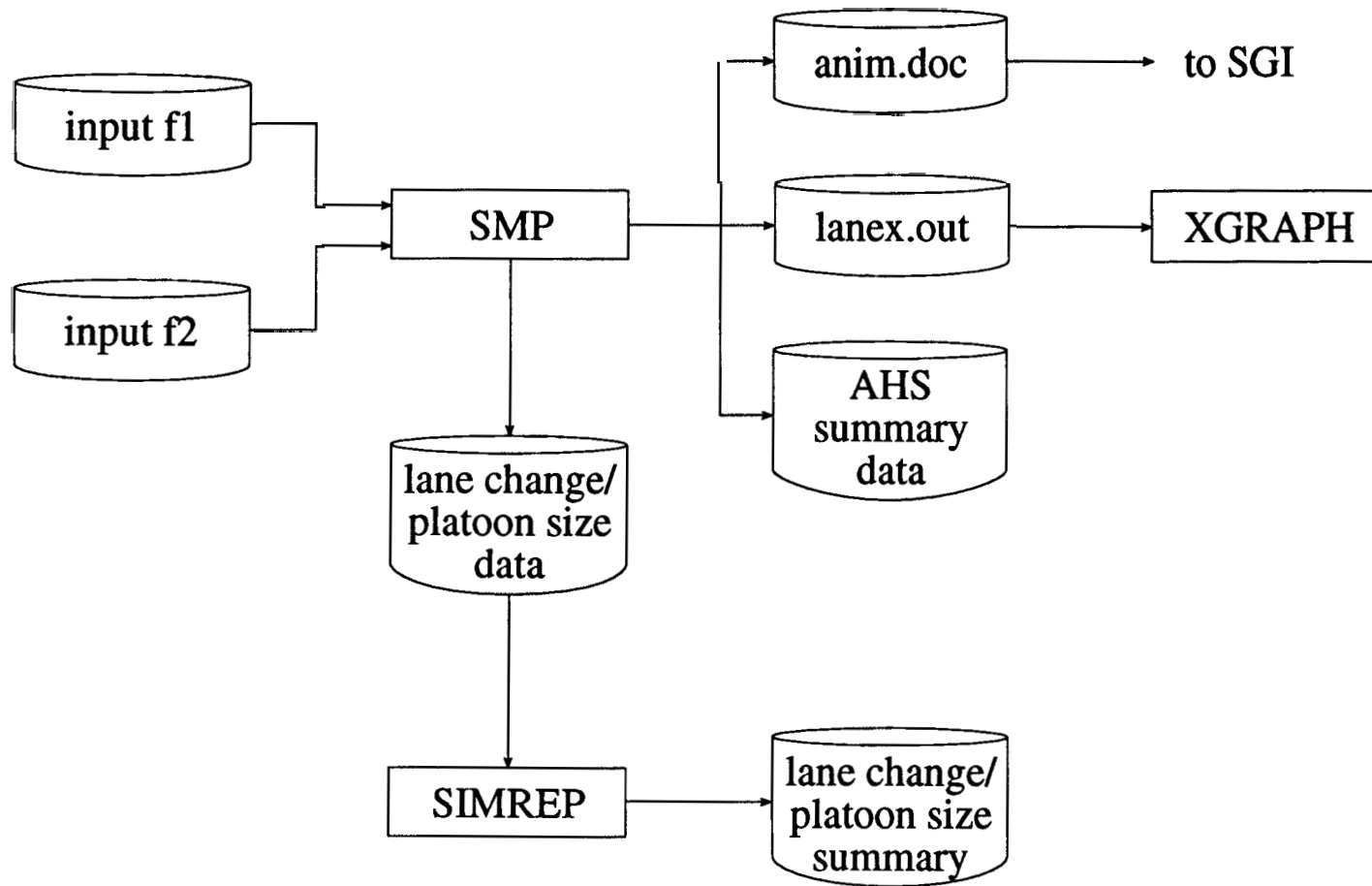


Figure 2-1 Smart PATH Version MOU 62 Data Flow

Section 2.2 Data Base Initialization

The user must set the following input files. They must be in the same directory as the executable programs.

- File "inputf1", the standard SmartPath input file (Section 2.4.1).
- File "inputf2", a new input file (Section 2.4.2).

These files can have any name that the user desires. A standard text editor on the workstation, such as vi, is used to set or modify these files.

Section 2.3 Program Execution

To run the MOU62 version of SmartPath, the user types:

```
carout inputf1 inputf2 > debug.
```

carout produces a file called "ahs.sum", described in Section 2.4.3. "debug" contains data used by the reporting function "simrep". simrep is run by typing:

```
simrep inputf2 < debug > outputf1,
```

where "outputf1" is a file containing statistics on lane changes, requests to change lanes, and platoon size distributions. It is described in Section 2.4.4.

If the user does not need to **look** at detailed lane change data, then the "debug" file does not have to be created, and the above two commands can be combined into:

```
carout inputfl inputf2 | simrep inputf2 > outputfl
```

The output of simrep (outputfl) can have any name the user desires. Section 2.4 contains definitions of the input and output file data items. Section 3 contains examples of input files used and output files generated during a particular simulation.

Section 2.4 Data File Description

Section 2.4.1 SmartPath Input File 1

The original SmartPath input file smp_inp was not changed by this development. The data items used by the simulator are defined below.

T__totalSimulation	—	Total simulation time (in seconds).
T__entcreate	—	Time between creation of two successive cars at entrances (in seconds).
T__beginEntrance	—	Time to start the creation of cars at entrances (sec).
T__endEntrance	—	Time to end the creation of cars from entrances (sec); = 0 if no cars desired from entrances.
T__entbreak	—	Time to go from creating cars in one set of entrances to another set (seconds).
entrance-sets	—	The cars will be created in the first set during the 0-T__entbreak interval, and in the second set during the T__entbreak-2*T__entbreak interval, etc. A set is defined by a first and last value; i.e., it includes all

	entrances in between (inclusive).
number-lanes –	Number of lanes.
highway-length –	Length of the highway (meters).
optimum-size –	Optimum (maximum) size of the platoon.
ramplength –	Length of entrance and exit ramps (meters).
rampvelocity –	The starting velocity of the cars in the entrance ramps (meters/sec).
Autolanes –	Number of automated lanes.
entrance-dist –	The location of entrances with respect to the beginning of the highway (meters).
exit_dist –	The location of exits with respect to the beginning of the highway (meters).
intraplat –	Distance between two cars in a platoon (meters).
detect-range –	Detection range of sensors for locating other cars in the highway.
Safety-distances –	Safety distance for each platoon according to its length (meters).
exit-number –	The exit number each car must take.

Section 2.4.1.1 Input File 1 Items Not Used

Some data items from the original SmartPath are not used in the MOU62 version of SmartPath. These items are listed below, along with an explanation for why they are no longer part of the simulation model.

T_endAuto – Time to end the creation of cars in automated lanes.

T_autocreate – Time between creation of two successive cars in automated lanes.

A user-specified number of platoons of varying sizes is set in the new input File 2 which determines the amount of automated traffic to generate. Also, the desired inter-platoon spacing is randomly determined using a “shifted exponential distribution” (see Appendix A.2) defined by parameters in File 2.

optimum_velocity — Optimum velocity in the highway segment.

Initial and desired velocities for each lane in the highway segment are defined in File 2 instead.

id-of-cars — Id of the cars to exit from the freeway.

A user-specified percentage of **all** automated vehicles to exit from the highway segment is set in File 2. It is no longer necessary to specify the ids **of** individual vehicles to exit from the highway segment.

Time-to-exit — Time to **start** the exit maneuver for each car.

The SmartPath function “exitcar” was changed so that vehicles that were assigned an exit will start the exit procedure at a specified distance into the highway segment.

Section 2.4.2 SmartPath Input File 2

The data items in the new File 2 that must be set by the user are identified below. In addition, a description follows on how they are used in the simulation.

pmax — Maximum platoon size.

plke –	Option to keep other platoon at the same velocity until the beginning of a gate is reached (1=yes, 0=no).
pcte –	Percentage of vehicles that need to exit in this highway segment (0-100).
pspi –	Initial platoon speed.
pspd[1 –	Desired platoon speed in each lane.
nump –	Number of platoons to be generated in each automated lane.
mean-space –	Mean initial spacing between platoons (meters).
equil –	Equilibrium at beginning of highway segment (0=no, 1=yes). This guarantees that all initial inter-platoon distances are greater than communication distance.
mean-plat –	Mean platoon size (Poisson distribution).
delm[][1 –	Desired lane change proportion matrix (n×n) where n is the number of automated lanes. delm[x][y] = percentage (0-1) of vehicles in lane x that are assigned to reach lane y by the end of the highway segment.
hwln –	Distance in the highway segment at which platoon sizes and number of platoons of each size is to be determined.
hwg[] –	For each automated lane: Number of gates. Start and end distances along the highway at which lane changes are allowed (i.e. gates). For each gate, designator for allowing left lane changes only ("l"), right lane changes only ("r") or both ("b").

Section 2.4.3 SmartPath Output File "ahs.sum"

The AHS summary data file generated by carout contains the following data:

Origin/Destination matrix; for each automated lane:

- Number of vehicles assigned to each automated lane and exit
- Number of vehicles reaching each automated lane and exit
- Percentage (0-100) of vehicles reaching each automated lane/exit
- Sum of vehicles assigned to all automated lanes and exits
- Sum of vehicles that reached all automated lanes and exits
- Percentage (0-100) of vehicles reaching all automated lanes/exits

Number of vehicles generated

Number of automated vehicles generated

Average speed and standard deviation of speed for the highway segment

For each section and for each lane in each section within the segment:

- Section boundary (in meters) at which traffic flow statistics are determined
- Number of vehicles crossing the boundary
- Average speed and standard deviation of **speed** in the section
- Traffic **flow** in vehicles/hour
- Density in vehicles per section distance (1000 meters)
- First and last time a vehicle crossed the boundary for each lane
- Number of lane changes from the left for each lane
- Number of lane changes **from** the right for each lane

Section 2.4.4 Simrep Output File

The statistical summary data file generated by simrep contains the following data:

- Number of attempted lane changes

- Number of lane changes

- Number of lane changes with communication not necessary

- Percentage of lane changes not requiring communication

- Number of lane change requests

- Mean and Standard Deviation of:

 - Distance and time of lane change begin to end

 - Distance and time of lane change begin to acceptance by other vehicle

 - Distance and time of lane change acceptance to end

 - Number of requests per vehicle

- Histogram data for lane change distances:

 - from begin to end

 - from begin to accept

- At a specified distance along the highway segment (Sect. **2.4.2** — hwlh):

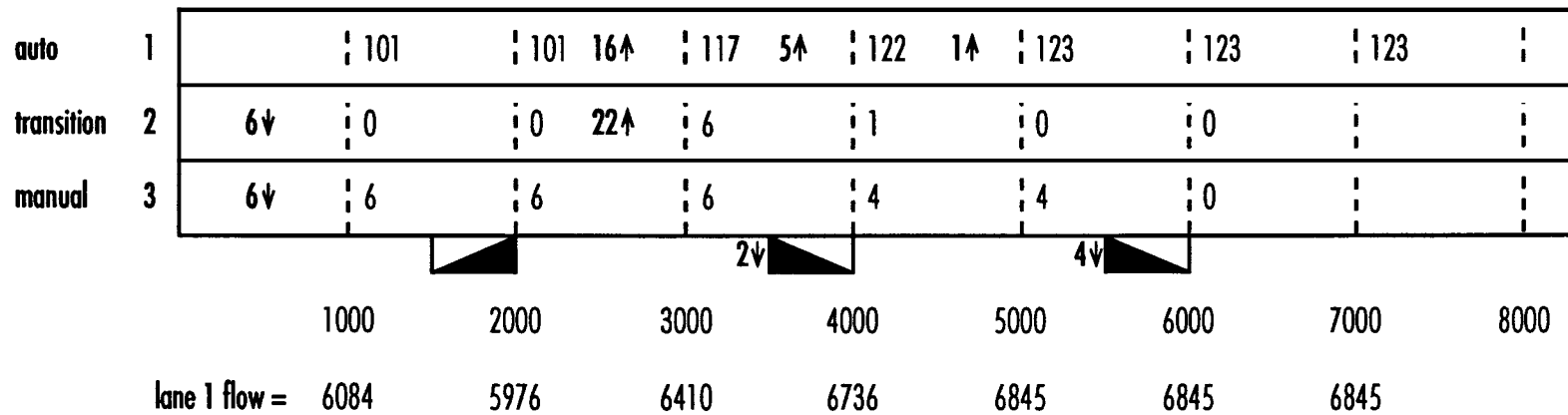
 - Average platoon size

 - Number of platoons crossing the boundary

 - Number of platoons of each size

Section 3. Data File Listing

This section contains examples of the files described in Sections 2.4.1—2.4.4. The input files in Sections 3.1—3.2 were used for an example simulation run. The output files in Sections 3.3—3.4 were produced by this run. Figure 3-1 shows a highway segment along with some of the parameters used in this simulation. The diagram shows the flow and lane change data produced in file ahs.sum (Section 3.3) along with some of the input parameters.



above data is from ahs.sun

- exit % = 6%
- desired speed = 27 m/sec
- platoons generated = 12
- mean platoon size = 8
- mean initial spacing = 80 m
- simulation time = 360 sec
- segment length = 10000 m
- detect range = 60 m
- safety distance = 40 m



-  — entrance
-  — exit
- ↓ — lane changes from left
- ↑ — lane changes from right
- ⋮ n — number of vehicles

Figure 3-1 AHS Highway Segment—Simulation Example

Section 3.1 Example of SmartPath Input File 1

The numbers above each parameter name are the number of entries followed by the lower end upper bounds. The value(s) are below the parameter name. Only the actual values are set or changed by the user. The pound sign (#) separates the information for each parameter.

8 9 3 4 {simulation, system, platoon, maneuver}

1 0. 9999999999.

T_total Simulation

360.

Total simulation time, in sec.

#

1 0. 9999999999.

T_endAuto

30.

Time (in **seconds**) to end the creation of *cars* in the platoon lane, **set** to zero if **no car** is desired in the platoon lane.

#

1.0 9999999999.

T_autocreate

1.

Time (**inseconds**) **between** creation of two successive platoons in the platoon lane.

#

1.0 9999999999.

T_entcreate

2.0

Time (**inseconds**) **between** creation of **two** successive *cars* from **entrances**.

#

1 0. 9999999999.

T_beginEntrance

15.

Time (in seconds) to **start** the creation of *cars* from entrances.

#

1 0. 9999999999.

T_endEntrance

120.

Time (**inseconds**) to end the creation of *cars* from entrances, **set** to **zero** if **no cars** is **desired**

from entrances.

#

1. 9999999999.

T_entbreak

200.

Time (in seconds) to go from creating cars in one set of entrances to another set.

#

14 1. 10.

entrance-sets

1

1

0

0

0

0

0

0

0

0

0

0

0

0

The cars will be created in the first set during the 0-T_entbreak interval, and in the second set of entrances during the T_entbreak- 2*T_entbreak interval and so on.

#

1 0 4

number-lanes

4

Number of lanes

#

1 0. 10000.

highway-length

10000.

Length (in meters) of the highway.

#

1 10. 30.

optimum-velocity

27.

Optimum velocity (meters per second) in the highway.

#

1.0 20.

optimum-size

20

Optimum size of the platoon.

#

1 0. highway-length

ramplength

300.

Length (in meters) of exit and entrance ramps.

#

1 10. optimum-velocity

rampvelocity

27.

The starting velocity of the cars in the entrance ramps.

#

1 0 4

Autolanes

1

Number of automated lanes.

#

10 0. highway-length

entrance-dist

2000

0

0

0

0

0

0

0

0

0

The location of entrances with respect to the beginning of the highway (has to end with a 0).

#

10 ramplength highway-length

exit-dist

4000

6000

0

40

Safety distance (meters) for each platoon according to its length.

#

10 1 60

id-of-cars

0

0

0

0

0

0

0

0

Id of the cars to exit from the freeway.

#

10 .1 T_totalSimulation

Time-to-exit

0

0

0

0

0

0

0

time to start the exit maneuver for each car, in seconds .

#

10 1 10

exit-number

1

2

0

0

0

0

The exit number each car must take.

#

Section 3.2 Example of SmartPath Input File 2

```
# This file contains the simulation input/control data and the
# platoon input data
#
# The keyword for each data item (or items) is followed by:
#   parameter values for a succession of items if they are 1x1 arrays
#   dimension size(s) if the item is an array (values follow)
#
# The following is the simulation input/control data
#
# itel parameter data. The format is
# itel plke pcte
#
# pmax - Maximum platoon size
# plke - Option to keep other plat at same veloc until gate begin (1=yes 0=no)
# pcte - Percentage of vehicles that need to exit
#
itel 20 0 6
#
# Platoon initial and desired speed by lane (meters/sec)
#
pspi 27
pspd 27 27 27 27
# The following is the vehicle/platoon input data
#
# ite2 parameter data. The format is
# ite2
#
# ite2 - keyword
# nump      - Number of platoons to be generated in each lane
# mean-space- Mean initial spacing between platoons (meters)
# equil     - Equilibrium at beginning of highway segment (0=no, 1=yes).
#           Ensure that initial inter-platoon distance > comm distance
# mean_plat - Mean platoon size (Poisson distribution)
#
ite2 12 80. 1 8
```

```

#
# Desired lane change proportion matrix (n x n) where n = number of lanes.
# g (x,y) = % of vehicles in lane x that are desired to change to lane y
# by the end of the highway segment.
#
delm
1.0 . 0. 0.
0. 1.0 . 0.
0. 0. 1.0 .
0. 0. 0. 1.0
#
# Report parameter data
#
# Highway segment length at which platoon sizes are measured
#
hwln 5000
#
# To the right of each lane, the start and end distances along the highway
# segment at which lane changes can occur (i.e., positions/lengths of gates).
# Designator for type of lane change allowed is also included
# l = only left lane change allowed (logical "on ramp")
# r = only right lane change allowed (logical "off ramp")
# b = both left and right lane changes are allowed
# Data is for one lane at a time, starting with lane 1. (Distance in meters)
#
hwg1 1 /* lane 1 */
0 10000 b
hwg2 1 /* lane 2 */
0 10000 b
hwg3 1 /* lane 3 */
0 10000 b
hwg4 1 /* lane 4 */
0 10000 b
hwg5 1
0 10000 b

```

Section 3.3 Example of Output File "ahs.sum"

Some of the data in this file are shown in Figure 3-1. The number of automated vehicles generated is 109. The number of vehicles generated **from** the manual entrance is $131-109=22$. All six of the vehicles that needed to exit in this segment did **so**. Therefore, the exit success percentage is 100%. The last lane in the matrix (Lane 4) is the exit in this example. The flow statistics are for the entire segment, for all lanes in a 1000 meter (sub)section, and for each lane in each section. The lane changes are given for each lane in each 1000 meter section. This file is described in Section 2.4.3.

AUTOMATED HIGHWAY SUMMARY DATA

Origin/Destination Matrix

Vehicles assigned/reached/percent reached each lane					
Lane	Lane 1	Lane 2	Lane 3	Lane 4	Total
1	103 100.0	0 .0	0 .0	6 100.0	109 100.0

VEHICLES= 131 AUTO VEH= 109
 AVE SPEED(m/sec)= 26.50 AVE SPEED(mph)= 59.25 STD DEV = 3.02

SECTION= 1 BOUNDARY= 1000 meters
 VEHICLES = 107
 AVE SPEED= 24.03 meters/sec
 AVE SPEED= 53.72 miles/hour
 STD DEV = 6.72 meters/sec
 TOT FLOW = 6519.10 vehicleshour
 DENSITY = 74.98 vehicles/section

SECTION= 1 LANE= 1
VEHICLES = 101
AVE SPEED= 23.97 **meters/sec**
AVE SPEED= 53.59 **miles/hour**
STD DEV = 6.77 **meters/sec**
FLOW = 6084.49 **vehicles/hour**
DENSITY = 70.51 **vehicles/section**
1ST TIME = 42.1 **seconds**
LAST TIME= 75.2 **seconds**
CHANGES FROM LEFT LANE = 0
CHANGES FROM RIGHT LANE = 0

SECTION= 1 LANE= 2
VEHICLES = 0
AVE SPEED= 0.00 **meters/sec**
AVE SPEED= 0.00 **miles/hour**
STD DEV = 0.00 **meters/sec**
FLOW = 0.00 **vehicles/hour**
DENSITY = 0.00 **vehicles/section**
1ST TIME = 0.0 **seconds**
LAST TIME= 0.0 **seconds**
CHANGES FROM LEFT LANE = 6
CHANGES FROM RIGHT LANE = 0

SECTION= 1 LANE= 3
VEHICLES = 6
AVE SPEED= 27.00 **meters/sec**
AVE SPEED= 60.36 **miles/hour**
STD DEV = 0.00 **meters/sec**
FLOW = 434.61 **vehicles/hour**
DENSITY = 4.47 **vehicles/section**
1ST TIME = 51.3 **seconds**
LAST TIME= 84.4 **seconds**
CHANGES FROM LEFT LANE = 6
CHANGES FROM RIGHT LANE = 0

SECTION= 1 LANE= 4
VEHICLES = 0
AVE SPEED= 0.00 meters/sec
AVE SPEED= 0.00 miles/hour
STD DEV = 0.00 meters/sec
FLOW = 0.00 vehicleshour
DENSITY = 0.00 vehicles/section
1ST TIME = 0.0 seconds
LAST TIME= 0.0 seconds
CHANGES FROM LEFT LANE = 0
CHANGES FROM RIGHT LANE = 0

SECTION= 2 BOUNDARY= 2000 meters
VEHICLES = 107
AVE SPEED= 27.05 meters/sec
AVE SPEED= 60.47 mileshour
STD DEV = 0.44 meters/sec
TOT FLOW = 6410.45 vehicleshour
DENSITY = 65.84 vehicles/section

SECTION= 2 LANE= 1
VEHICLES = 101
AVE SPEED= 27.05 meters/sec
AVE SPEED= 60.48 mileshour
STD DEV = 0.45 meters/sec
FLOW = 5975.84 vehicleshour
DENSITY = 61.36 vehicles/section
1ST TIME = 79.1 seconds
LAST TIME= 112.2 seconds
CHANGES FROM LEFT LANE = 0
CHANGES FROM RIGHT LANE = 0

SECTION= 2 LANE= 2
VEHICLES = 0
AVE SPEED= 0.00 meters/sec
AVE SPEED= 0.00 miles/hour
STD DEV = 0.00 meters/sec
FLOW = 0.00 vehicleshour
DENSITY = 0.00 vehicles/section
1ST TIME = 0.0 seconds
LAST TIME= 0.0 seconds
CHANGES FROM LEFT LANE = 0
CHANGES FROM RIGHT LANE = 0

SECTION= 2 LANE= 3 VEHICLES = 6
AVE SPEED= 26.98 meters/sec
AVE SPEED= 60.32 miles/hour
STD DEV = 0.12 meters/sec
FLOW = 434.61 vehicleshour
DENSITY = 4.47 vehicles/section
1ST TIME = 88.3 seconds
LAST TIME= 121.4 seconds
CHANGES FROM LEFT LANE = 0
CHANGES FROM RIGHT LANE = 0

SECTION= 2 LANE= 4
VEHICLES = 0
AVE SPEED= 0.00 meters/sec
AVE SPEED= 0.00 miles/hour
STD DEV = 0.00 meters/sec
FLOW = 0.00 vehicleshour
DENSITY = 0.00 vehicles/section
1ST TIME = 0.0 seconds
LAST TIME= 0.0 seconds
CHANGES FROM LEFT LANE = 0
CHANGES FROM RIGHT LANE = 0

SECTION= 3 BOUNDARY= 3000 meters
VEHICLES = 129
AVE SPEED= 25.57 meters/sec
AVE SPEED= 57.16 miles/hour
STD DEV = 4.86 meters/sec
TOT FLOW = 7171.01 vehicles/hour
DENSITY = 77.81 vehicles/section

SECTION= 3 LANE= 1
VEHICLES = 117
AVE SPEED= 25.54 meters/sec
AVE SPEED= 57.11 miles/hour
STD DEV = 4.98 meters/sec
FLOW = 6410.45 vehicles/hour
DENSITY = 69.71 vehicles/section
1ST TIME = 116.2 seconds
LAST TIME= 149.3 seconds
CHANGES FROM LEFT LANE = 0
CHANGES FROM RIGHT LANE = 16

SECTION= 3 LANE= 2
VEHICLES = 6
AVE SPEED= 24.97 meters/sec
AVE SPEED= 55.82 miles/hour
STD DEV = 4.59 meters/sec
FLOW = 325.95 vehicles/hour
DENSITY = 3.63 vehicles/section
1ST TIME = 122.0 seconds
LAST TIME= 155.1 seconds
CHANGES FROM LEFT LANE = 0
CHANGES FROM RIGHT LANE = 22

SECTION= 3 LANE= 3
VEHICLES = 6
AVE SPEED= 27.00 **meters/sec**
AVE SPEED= 60.36 **miles/hour**
STD DEV = 0.00 **meters/sec**
FLOW = 434.61 **vehicles/hour**
DENSITY = 4.47 **vehicles/section**
1ST TIME = 125.3 **seconds**
LAST TIME= 158.4 **seconds**
CHANGES FROM LEFT LANE = 0
CHANGES FROM RIGHT LANE = 0

SECTION= 3 LANE= 4
VEHICLES = 0
AVE SPEED= 0.00 **meters/sec**
AVE SPEED= 0.00 **miles/hour**
STD DEV = 0.00 **meters/sec**
FLOW = 0.00 **vehicles/hour**
DENSITY = 0.00 **vehicles/section**
1ST TIME = 0.0 **seconds**
LAST TIME= 0.0 **seconds**
CHANGES FROM LEFT LANE = 0
CHANGES FROM RIGHT LANE = 0

SECTION= 4 BOUNDARY= 4000 meters
VEHICLES = 127
AVE SPEED= 26.99 **meters/sec**
AVE SPEED= 60.34 **miles/hour**
STD DEV = 1.66 **meters/sec**
TOT FLOW = 7062.36 **vehicles/hour**
DENSITY = 72.69 **vehicles/section**

SECTION= 4 LANE= 1
VEHICLES = 122
AVE SPEED= 26.99 meters/sec
AVE SPEED= 60.33 miles/hour
STD DEV = 1.68 meters/sec
FLOW = 6736.40 vehicles/hour
DENSITY = 69.33 vehicles/section
1ST TIME = 153.2 seconds
LAST TIME= 186.3 seconds
CHANGES FROM LEFT LANE = 0
CHANGES FROM RIGHT LANE = 5

SECTION= 4 LANE= 2
VEHICLES = 1
AVE SPEED= 27.00 meters/sec
AVE SPEED= 60.36 miles/hour
STD DEV = 0.00 meters/sec
FLOW = 0.00 vehicles/hour
DENSITY = 0.00 vehicles/section
1ST TIME = 163.4 seconds
LAST TIME= 196.5 seconds
CHANGES FROM LEFT LANE = 0
CHANGES FROM RIGHT LANE = 0

SECTION= 4 LANE= 3
VEHICLES = 4
AVE SPEED= 27.00 meters/sec
AVE SPEED= 60.36 miles/hour
STD DEV = 0.00 meters/sec
FLOW = 325.95 vehicles/hour
DENSITY = 3.35 vehicles/section
1ST TIME = 180.0 seconds
LAST TIME= 213.1 seconds
CHANGES FROM LEFT LANE = 0
CHANGES FROM RIGHT LANE = 0

SECTION= 4 LANE= 4
VEHICLES = 0
AVE SPEED= 0.00 meters/sec
AVE SPEED= 0.00 miles/hour
STD DEV = 0.00 meters/sec
FLOW = 0.00 vehicles/hour
DENSITY = 0.00 vehicles/section
1ST TIME = 0.0 seconds
LAST TIME= 0.0 seconds
CHANGES FROM LEFT LANE = 2
CHANGES FROM RIGHT LANE = 0

.... data continues for up to 10000 meters of highway

Section 3.4 Example of Simrep Output File

This file is described in Section 2.4.4. The Lane Change Summary Data is applicable to the entire highway segment (10,000 meters in this case). "no comm" means that a lane change did not require vehicle-to-vehicle communication to accomplish a lane change. This means that no neighboring vehicle needed to coordinate with the vehicle needing to change lanes. The mean distances and times for lane change maneuvers include begin to end, begin to accept, and accept to end. The maneuver begins when the vehicle is a free agent and it needs to change lanes. The actual lane change does not *start* until after acceptance is gained from a neighboring platoon, if there is one. Acceptance means that the neighboring platoon changes velocity or splits to accommodate a lane change. "end" happens when the actual lane change is completed. The number of requests and mean number of requests per lane change are applicable up until acceptance is gained from a neighboring platoon. The histogram data show the distribution of lane change distances, i.e., the number of lane changes for each 20-meter interval. The platoon distribution data is for all platoons crossing a boundary for all automated lanes.

LANE CHANGE SUMMARY DATA

```

number of lane changes - attempted = 56
number of lane changes - total = 56
number of lane changes - nocomm = 22
pct of lane chgs(total/attempted) = 1.00
pct of lane chgs(nocomm/total) = 0.39

mean= 283.4 std dev= 391.5 distance - begin to end
mean= 10.9 std dev= 15.0 time - begin to end
mean= 156.3 std dev= 367.6 distance - begin to accept
mean= 5.9 std dev= 13.9 time - begin to accept
mean= 127.1 std dev= 79.3 distance - accept to end
mean= 5.0 std dev= 3.3 time -accept to end
mean= 0.0 std dev= 0.0 number of gates passed

```

number of requests= 912
mean= 16.3 std dev= 36.8 number of requests

histogram data - lane change distances from begin to end

i= 0 from 0 to 20 meters numlc= 0
i= 1 from 20 to 40 meters numlc= 0
i= 2 from 40 to 60 meters numlc= 3
i= 3 from 60 to 80 meters numlc= 4
i= 4 from 80 to 100 meters numlc= 27
i= 5 from 100 to 120 meters numlc= 1
i= 6 from 120 to 140 meters numlc= 0
i= 7 from 140 to 160 meters numlc= 1
i= 8 from 160 to 180 meters numlc= 0
i= 9 from 180 to 200 meters numlc= 0
i= 10 from 200 to 220 meters numlc= 2
i= 11 from 220 to 240 meters numlc= 1
i= 12 from 240 to 260 meters numlc= 1
i= 13 from 260 to 280 meters numlc= 3
i= 14 from 280 to 300 meters numlc= 1
i= 15 from 300 to 320 meters numlc= 0
i= 16 from 320 to 340 meters numlc= 0
i= 17 from 340 to 360 meters numlc= 0
i= 18 from 360 to 380 meters numlc= 0
i= 19 from 380 to 400 meters numlc= 0
i= 20 from 400 meters numlc= 12

histogram data - lane change distances from begin to accept

i= 0 from 0 to 20 meters numlc= 21
i= 1 from 20 to 40 meters numlc= 0
i= 2 from 40 to 60 meters numlc= 1
i= 3 from 60 to 80 meters numlc= 0
i= 4 from 80 to 100 meters numlc= 0
i= 5 from 100 to 120 meters numlc= 0
i= 6 from 120 to 140 meters numlc= 0
i= 7 from 140 to 160 meters numlc= 0
i= 8 from 160 to 180 meters numlc= 0
i= 9 from 180 to 200 meters numlc= 1
i= 10 from 200 to 220 meters numlc= 1
i= 11 from 220 to 240 meters numlc= 0

i= 12 from 240 to 260 meters numlc= 0
i= 13 from 260 to 280 meters numlc= 0

i= 14 from 280 to 300 meters numlc= 0
i= 15 from 300 to 320 meters numlc= 0
i= 16 from 320 to 340 meters numlc= 0
i= 17 from 340 to 360 meters numlc= 0
i= 18 from 360 to 380 meters numlc= 0
i= 19 from 380 to 400 meters numlc= 1
i= 20 from 400 meters numlc= 9

Platoon Distribution Data

ave platoon size= 12.5 number of platoons= 10 distance= 5000 meters
platoons of size 1 = 0
platoons of size 2 = 0
platoons of size 3 = 1
platoons of size 4 = 0
platoons of size 5 = 0
platoons of size 6 = 1
platoons of size 7 = 0
platoons of size 8 = 0
platoons of size 9 = 0
platoons of size 10 = 0
platoons of size 11 = 2
platoons of size 12 = 1
platoons of size 13 = 0
platoons of size 14 = 0
platoons of size 15 = 1
platoons of size 16 = 2
platoons of size 17 = 1
platoons of size 18 = 1
platoons of size 19 = 0
platoons of size 20 = 0

References

1. F. Eskafi and P. Varaiya. "SmartPath: an Automated Highway System Simulator," PATH Tech. Mem. 92-3, Institute of Transportation Studies, University of California, Berkeley, CA, October 1992.
2. B.S.Y. Rao and P. Varaiya. "Flow Benefits of Autonomous Intelligent Cruise Control in Mixed Manual and Automated Traffic," Institute of Transportation Studies, University of California, Berkeley, CA, July 1992.
3. CSIM Reference Manual. Revision 13, Microelectronics and Computer Technology Corp., 3500 West Balcones Center Dr., Austin, TX 78759.
4. A. Hsu, F. Eskafi, S. Sachs, and P. Varaiya. "The Design of Platoon Maneuver Protocols for IVHS," PRR-91-6, Institute of Transportation Studies, University of California, Berkeley, CA, April 1991.
5. P. Varaiya and S. Shladover. "Sketch of an IVHS Systems Architecture," PRR-91-3, Institute of Transportation Studies, University of California, Berkeley, CA, February 1991.
6. A. Hitchcock, "Physical Form of Dividers and Gates," Institute of Transportation Studies, University of California, Berkeley, CA, 1992.
7. A. Hitchcock, "Methods of Analysis of IVHS Safety", PATH Research Report, Institute of Transportation Studies, University of California, Berkeley, CA, 1992.

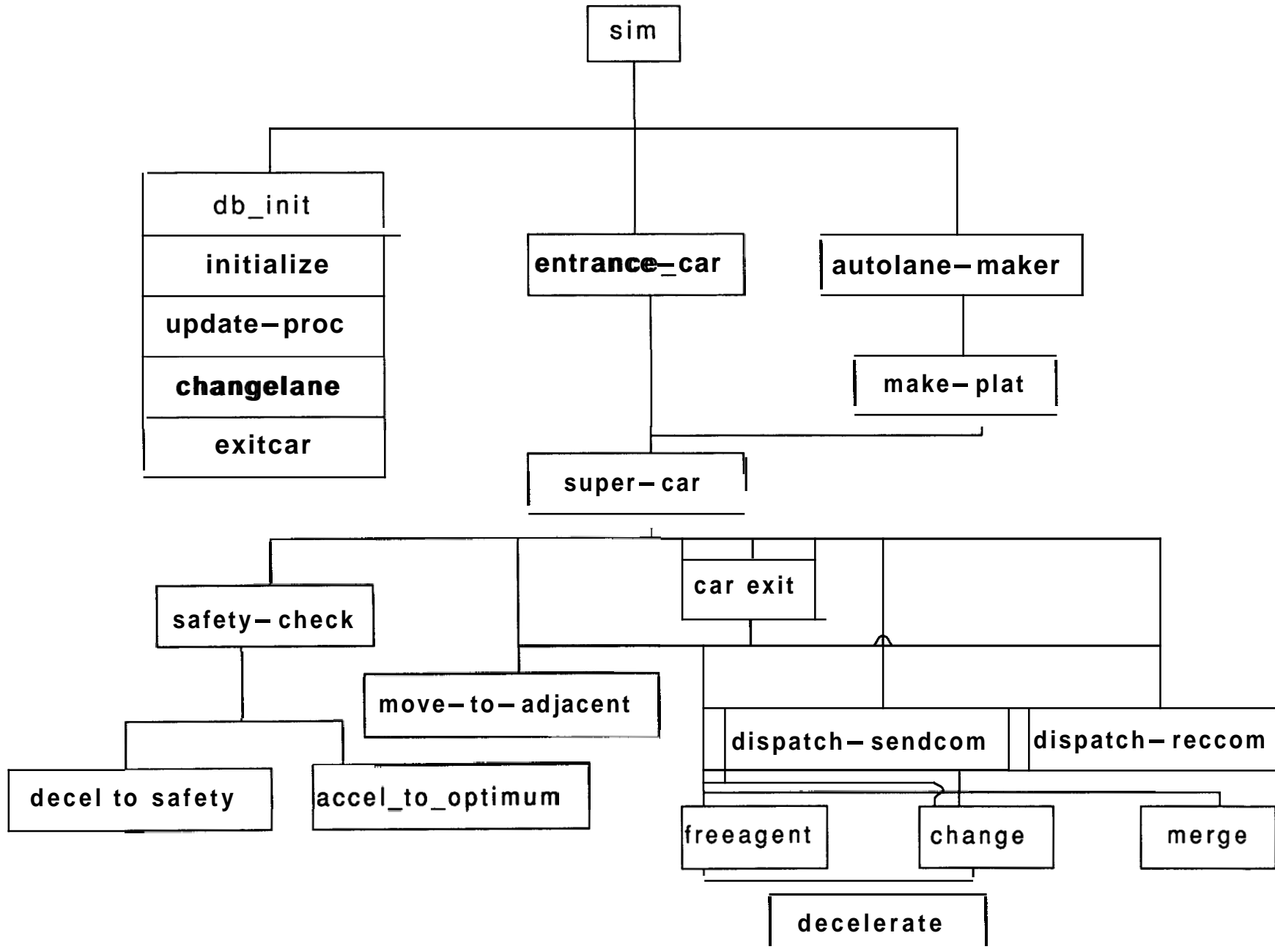
Appendix

A.1 Design Description

The system design of SmartPath is described in Reference [1]. This section contains a design overview with emphasis on describing modifications to simulator functions. More detailed design notes are contained in Appendix A.2.

If the user is not interested in the software design or in details describing software changes, this section can be skipped. Both new and existing capabilities for each function along with new functions are listed below. Figure A-1 contains a functional hierarchy diagram. New or changed capabilities are identified by an asterisk(*). Modified files now have one or more of the new #include files ahss.h, deff.h, dec.h and simrpt.h (see Appendix A.4).

Figure A-1. SmartPath Version MOU 62 Functional Hierarchy



The functions shown in Figure A-1 are described below. The order is from higher to lower level functions.

sim — Main function

- Calls function initialize to read SmartPath Input File 1.
- Calls function db_init to read SmartPath Input File 2. (*)
- Calls the following functions:
 - initialize
 - update_proc
 - changelane
 - exitcar
 - entrance_car
 - autolane_maker
- Terminates when time reaches the end simulation time.

initialize — Initialization function

- Sets data from Input File 1.
- Calls function "auto_traffic" to determine number of automated vehicles to generate. (*)
 - auto_traffic calls: (*)
 - function "numplat" to determine the number of vehicles in each platoon to be generated.
 - function "plat_int" to determine the inter-arrival times between all platoons to be generated
- Allocates memory needed for the SmartPath arrays

update_proc — Keeps hash table and distance array up-to-date for all vehicles in the highway segment.

- Keeps positions in the hash table up-to-date for all vehicles.
- Keeps distance array up-to-date for **all** vehicles based on velocity at previous time increment.
- For each simulation time increment:
 - write vehicle state data to the animation file (anim.doc).
 - save platoon size data **so** that size distribution can be determined by function "simrep". (*)
 - call function "simrpt" to accumulate vehicle state data. (*)
 - write vehicle **state** data to lane files used by "xgraph".
- call function sim_report to generate **AHS** summary report. (*)

changelane — Assign automated lanes to incoming manual vehicles

- Set the number of automated vehicles to be generated **so** that the remaining vehicle ids are used for manual vehicles coming from entrances. (*)
- Assign automated lanes to incoming manual vehicles.

exitcar — Assign an exit for each vehicle that needs to exit

- Based on a specified percentage of automated vehicles that need to exit, randomly determine which vehicles (among all automated vehicles to be generated) need to exit and assign each one an exit. (*)
- Activate an exit for each vehicle that needs to exit when the vehicle reaches the distance to start the exit maneuver. (*)

entrance_car — Generate traffic from entrances

Do the following for each set of entrances:

- Initialize state data for each vehicle.
- Generate manual traffic between the user-specified **start** and end times
- Wait the following amount **of** time before generating the next vehicle in a given entrance: (*)

The time corresponding to the length and speed of the present vehicle plus the desired constant inter-arrival time.

autolane_maker — Generate traffic from automated lanes

Do the following for each automated lane and each platoon to be generated: (*)

- Call function "check_traffic" to do the following:
 - Wait until the downstream platoon is further away than the safety distance before starting creation **of** platoon.
 - Set initial velocity of platoon to the same value **as** the downstream platoon if it is within communication distance.
- Call function "make_plat" to generate the platoon. (*)
- Wait the following amount of time before generating the next platoon in the same automated lane: (*)
 - The time corresponding to the length and speed **of** the present platoon plus the desired inter-platoon spacing.

make_plat — Generate a platoon in an automated lane (*)

- Initialize state data for each vehicle
- Call function "assign_lane" to assign an automated lane for each vehicle.
 - "assign_lane" uses the desired lane change proportion matrix from input file 2.
- If the vehicle needs to exit, assign the last lane (exit).
- Call function "super_car" for each vehicle.
- Wait the following amount of time before generating the next vehicle in this platoon:
 - The time corresponding to the length of a vehicle, **speed** of the platoon and the desired intra-platoon spacing.

super_car — Main process for each vehicle (see Reference [1]).

- Perform the following initialization tasks:
 - Initialize number of lanes and automated lanes for function "sim_report". (*)
 - Enter initial distance in distance array.
 - Open mailbox for this vehicle.
 - Set vehicle ID in the hash table.
 - Wait until the entire platoon is created before continuing. (*)
 - Initiate the "safety_check" function.
- If vehicle is in an on-ramp (rightmost lane) call function "move-to-adjacent" to enter the highway.
- For each updated (by "update_proc") distance along the highway:

- Determine if the exit procedure should be initiated
 - See if exit number has been activated by "exitcar".
 - If necessary, call function 'carexit' to exit the vehicle.
- If the vehicle does not need to exit call "dispatch_sendcom".
 - Change lanes if assigned lane is not the same as own lane.
 - If in assigned lane, initiate merge procedure if possible.
- Check mailbox and if there is mail, call "dispatch_reccom".
 - Determine the action which should be taken.
 - Send acknowledge or busy message to sender of message.

safety_check — Change acceleration for each vehicle as necessary based on lane following rules.

- Set optimum velocity based on desired speed for this lane. (*)
- If vehicle is platoon leader:
 - Decelerate if within safety distance of downstream platoon.
 - Accelerate if not within safety distance and if velocity is less than optimum velocity.

move-to-adjacent — Change lanes to enter an off-ramp or enter highway from an on-ramp.

- Do the following within the time needed to change lanes:
 - Determine distance needed to change lanes.
 - Wait until space becomes available to change lanes.
 - Return not successful if destination is occupied.
- If lane change was successful:
 - Update vehicle state information.

- Set final lane for vehicle (for use by "sim_report") if final lane is an exit. (*)

carexit — Call necessary maneuvers to exit the vehicle.

- if the vehicle is not a free agent, call function "freeagent".
 - "freeagent" performs the split maneuver.
- Call function "move-to-adjacent" **if** the vehicle is in the lane next to the exit.
- Call function "change" if the vehicle **is** more than one lane away from the exit.

change — Perform necessary functions to accomplish a lane change.

- If there is a vehicle in the destination lane within sensor range:
 - Send message to request lane change.
 - If the other platoon is not busy, proceed with lane change.
 - If the other platoon is busy, return not successful.
 - If request to change lanes is accepted by the other platoon:
 - Send message to cause deceleration or split to create room for lane change.
 - Wait until gate is available to **start** lane change. (*)
 - Call function "check_gates".
 - During the time needed to change lanes:
 - Check to see if there is a vehicle in the next lane.
 - If **so**, start the lane change procedure over again.
 - If at the end time of lane change a divider is in the way, **start** the lane change procedure over again. (*)
 - Call function "check_gates".

- If divider is not in the way, change vehicle state information including new lane number. Return with successful result.
- If there is no vehicle in the destination lane within sensor range and there is no vehicle in the 2nd lane to the left or right:
 - Wait until gate is available to **start** lane change. (*)
 - Call function "check_gates".
 - During the time needed to change lanes:
 - Check to see if there is a vehicle in the next lane.
 - If so, start the lane change procedure over again.
 - If at the end time of lane change a divider is in the way, **start** the lane change procedure over again. (*)
 - Call function "check_gates".
 - If divider is not in the way, change vehicle state information including new lane number. Return with successful result.
- If there is a vehicle in the 2nd lane to the left or right:
 - Send message to ask vehicle for promise not to move.
 - If accepted, continue with lane change procedure.
 - If a vehicle is in the next lane **start** the lane change procedure over again.
 - Wait until gate is available to **start** lane change. (*)
 - Call function "check_gates".
 - During the time needed to change lanes:
 - Check to see if there is a vehicle in the next lane.
 - If **so**, **start** the lane change procedure over again.

- If divider is not in the way, change vehicle state information including new lane number.
- If there is no vehicle in the 2nd lane to the left or right:
 - Wait until gate is available to **start** lane change. (*)
 - Call function "check_gates".
 - During the time needed to change lanes:
 - Check to see if there is a vehicle in the next lane.
 - If **so, start** the lane change procedure over again.
 - If divider is not in the way, change vehicle **state** information including new lane number. Return with successful result.

A.2 Design Notes

This section contains the following:

- Detailed design notes for new functions;
- Notes on setting input parameters;

Function "db_init"

db_init initializes the input data from SmartPath Input File 2. It also adjusts the input mean platoon size to account for the fact that the Poisson distribution is truncated for values greater than the maximum platoon size.

Function "auto_traffic"

- auto_traffic determines the number of vehicles in each platoon (call num_plat) and the inter-arrival times between each platoon (call plat_int). It returns the total number of automated vehicles to be generated.
- The number of vehicles in a platoon to be created is randomly determined from the truncated Poisson distribution mentioned above.
- Each inter-arrival time for platoons **is** randomly determined from a "shifted exponential distribution" **as** follows:

Determine a mean time value to be used for an exponential probability distribution:

$$\text{mean_time} = (\text{mean_space} - \text{safetydist})/\text{initvel}$$

where mean_space = mean inter-platoon initial spacing
safetydist = safety distance for this platoon size
initvel = initial velocity

- Using mean_time, call "plat_int" to obtain a value from the exponential distribution (headexp).
- **So** the initial inter-platoon spacing **is**:

$$\text{inter_plat} = \text{headexp} + \text{safetydist}/\text{initvel}$$

- If equilibrium (i.e., a steady **state flow**) is desired at the beginning of the highway then:
 - Increase or decrease inter_plat (if necessary) so that it is always in the following range:

From $\text{drange}/\text{initvel}$

To $(\text{drange} + 2 * (\text{mean_space} - \text{drange}))/\text{initvel}$

where $\text{drange} = \text{detection range}$

This is done so that the initial inter-platoon distance is always greater than the communication distance. This is necessary if no platoon merging is desired near the beginning of the highway.

Function "check_gates"

- `check_gates` determines whether or not a lane change can be started at a particular time given:
 - distance along the highway
 - present lane
 - destination lane
 - start/end gate distances for each lane

Function "assign_lane"

- `assign_lane` randomly assigns a lane for a vehicle to be in before the end of the highway segment using the initial lane and the desired lane change proportion matrix.
- **If** the vehicle needs to exit, the value provided by `"assign_lane"` is changed and an exit (last lane) is assigned.

Function "simrpt"

- simrpt accumulates statistics from vehicle state data including:
 - lane change to right or left
 - number of vehicles
 - velocities, sums, sums of squares of velocities
 - final lane each vehicle is in

Function "sim_report" uses this data.

Function "sim_report"

- sim_report generates the output data for average velocities, **flows** and densities for all highway sections and lanes in "ahs.sum".
- **flow** is in vehicles/hour
- density is in vehicles/section
 - density in vehicles/meter = flow/average_speed
 - vehicles/section = vehicles/meter * sectionlength

Using sums and sums of squares of velocities to compute average velocity and standard deviation of velocity would result in very large sums by the end of the Simulation. Instead, an input (global) velocity value is subtracted from the accumulated velocities so that the sums are small by the end of the simulation. The global value is also used to obtain average velocity by adding it to the sum of velocity differences at the end of the simulation. Computation of average velocity and standard deviation of velocity follows.

INPUT:

Input velocity value (global) v

Sums of velocity differences

$$\sum_{i=l}^{i=n} (v_i - v)$$

Sums of squares of velocity differences

$$\sum_{i=l}^{i=n} (v_i - v)^2$$

Number of velocity values accumulated n

OUTPUT:

Actual average velocity v_a

Variance var

Standard Deviation std

PROCESSING:

$$v_a = \frac{1}{n} \sum_{i=1}^{i=n} (v_i - v) + v$$

since

$$\frac{1}{n} \sum_{i=1}^{i=n} (v_i - v) = \frac{1}{n} \sum_{i=1}^{i=n} (v_i) - v = v_a - v$$

$$var = \frac{1}{n} \sum_{i=1}^{i=n} (v_i - v)^2 - (v_a - v)^2$$

$$std = \sqrt{var}$$

A.3 Input Parameter Notes

The following table can be used as a guide on how large the mean platoon size must be set to achieve a desired initial flow in the highway segment. The platoon sizes and inter-platoon distances are rounded off.

Flow (veh/hr)	Values of N (platoon size)						
7000	7	8	8	9	10	11	12
6000	5	6	6	7	8	8	9
5000	4	4	5	5	6	6	7
4000	3	3	4	4	4	5	5
3000	2	2	3	3	3	3	4
2000	1	1	2	2	2	2	2
inter-platoon= distance (meters)	53	60	67	74	80	87	94

$$Flow = \frac{3600 \cdot v \cdot N}{N(l + d) - d + int}$$

where: int = inter-platoon distance = $D_{safe} + expm \cdot v$
 N = mean platoon size
 v = velocity (27 meters/sec)
 l = length of vehicle (5 meters)
 d = intra-platoon spacing (1 meter)
 int = inter-platoon distance (meters)
 3600 — conversion of veh/sec to veh/hr
 D_{safe} — safety distance
 expm — mean of exponential distribution

A.4 Source Code and Data Files

The following is a listing of the file names and sizes for source code, executables, include files and data files. An asterisk (*) is used to indicate new files.

Kilobytes	File Name
4323	ahss_inp • (input file)
2591	smp_inp * (input file)
344064	carout * (executable)
2805	csim.h In lib directory
4543	csimdef.h In lib directory
272932	libcsim.a In lib directory
1983	ahss.h *
10551	break.c
565	car.h
27070	change.c
5617	db_init.c *
903	dec.h •
4767	decelerate.c
3733	def.h
988	deff.h *
4930	dispatch.c

Kilobytes	File Name
1868	exit.c
5003	free.c
10689	initialize.c
539	makefile * (with debug)
539	makefile.nodeb * (no debug)
519	makefile.smp original SMP
11652	merge.c
9306	sim.c
3233	sim.h
49152	simrep * (executable)
13887	simrep.c *
2837	simrpt.h *
1823	simuser.h
26252	supercar.c
1768	user.h
122880	xgraph (executable)

A.5 Glossary

AHS – Automated Highway System.

animation file – The file of vehicle state data for each time increment. It is used by the Silicon Graphics display function.

automated lane – An AHS lane containing automated vehicles.

carout – Name of executable for the SmartPath function.

CSIM – A computer language used for inter-process communication. It provides a quasi-parallel processing environment so that the processes (e.g., vehicles) are in sync over time.

hash table – A table whose index represents each position and lane along the highway and contains the vehicle ID. The table is updated at each time increment.

inter-platoon distance – Distance between successive platoons.

intra-platoon distance – Distance between members of platoons.

mailbox – Repository of messages for each vehicle received from other vehicles at a given time (part of **CSIM**).

manual lane – An AHS lane containing manually operated vehicles.

platoon – Automated vehicles traveling in a formation such that they are equally spaced apart by a short distance.

section – A partition of a highway segment of distance 1 kilometer.

segment — A length of highway with distance up to 10 kilometers.

simrep — Name of executable for the reporting function.

transition — A manual lane that is adjacent to the rightmost automated lane.

xgraph — A function that graphically displays time and distance data for all platoons.