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

Studies of Vehicle Collisions - A Documentation of the Simulation Codes: SMAC (Simulation Model of Automobile Collisions) Update 1

Ching-Yao Chan

California PATH Working Paper
UCB-ITS-PWP-99-4

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 Transportation, Federal Highway Administration.

Report for MOU 324

March 1999

ISSN 1055-1417

Studies of Vehicle Collisions – A Documentation of the Simulation Codes: SMAC (Simulation Model of Automobile Collisions) Update 1

Ching-Yao Chan
California PATH
Institute of Transportation Studies
University of California, Berkeley

ABSTRACT

This document describes part of the work conducted under MOU252 and MOU324, related to the studies of vehicle collisions in vehicle-following operations.

This working paper is a detailed documentation of a computer program that is the core element of the simulation tools for vehicle collision dynamics. The program, SMAC (Simulation Model of Automobile Collisions), and its PC-platform version EDSMAC have been used extensively in recent work at PATH to investigate the consequences of vehicle collisions and the effects of vehicle-following parameters on collisions.

A copy of the source codes of SMAC was obtained from the University of Michigan Transportation Research Institute (UMTRI). Revisions to the program were made to insert the option of exercising feedback control in collision situations. After these revisions, control algorithms can now be tested in crash scenarios to examine the feasibility and effectiveness of vehicle control in emergency conditions.

The structure of the program was outlined in this report with descriptions of the major subroutines. The added options for implementing user-specified steering inputs and feedback controllers were explained in details. An example input file for running the computer program was also provided to illustrate the formats and contents of input parameters. This working report provides a concise and essential documentation for the computer program.

KEY WORDS

Vehicle Collisions
Simulation of Vehicle Crashes
Vehicle Control in Collisions
Advanced Vehicle Control Systems

EXECUTIVE SUMMARY

This working paper is a detailed documentation of a computer program that is the core element of the simulation tools for vehicle collision dynamics. The program, SMAC (Simulation Model of Automobile Collisions), and its PC-platform version EDSMAC have been used extensively in previous studies at PATH to investigate the consequences of vehicle collisions and the effects of vehicle-following parameters on collisions.

SMAC analyzes the longitudinal and lateral movements of vehicles as well as the rotational motion about the vertical axis of vehicles on a horizontal plane. If a contact between vehicles is detected, the collision phase is analyzed. The external forces can be applied either at the tire/road interface or between the vehicles.

A copy of the source codes of SMAC was obtained from the University of Michigan Transportation Research Institute (UMTRI). The source code was written in FORTRAN. Revisions to the program, described in this paper, were made by the author to insert the option of exercising feedback control in collision situations. After these revisions, control algorithms can now be tested in crash scenarios to examine the feasibility and effectiveness of vehicle control in emergency conditions.

The structure of the program was outlined in this report with descriptions of the major subroutines. The added options for implementing user-specified steering inputs and feedback controllers were explained in details. An example input file for running the computer program was also provided to illustrate the formats and contents of input parameters.

With the completion of the revisions to the program, the simulation tool is now available for the studies of advanced vehicle control in collisions. This working report provides a concise and essential documentation for the computer program. It will facilitate the use of this program as a validation model or combine it with other simulation tools.

1.0 INTRODUCTION

This progress report is generated from the work conducted under MOU-324. In this project, which is an extension of MOU-252, a simulation program was used to study the phenomena of vehicle dynamics in vehicle-following collisions. As part of the proposed work, the source codes of this simulation program was obtained and modified so that additional features can be incorporated into the codes to serve the purposes of research activities.

In this report, the source codes are reviewed and documented for future references. A floppy disk containing the source codes is attached with this report.

2.0 BACKGROUND INFORMATION ON SMAC

The analysis of vehicle collisions in the work related to MOU-252 is conducted with a simulation program developed by Engineering Dynamics Corporation (EDC) for PC-DOS platforms. The software package, EDSMAC (Engineering Dynamics Corporation Simulation Model of Automobile Collisions), is used for the analysis of two-vehicle collisions. It is based on a program called SMAC [1-3] initially developed by Calspan Corporation for National Highway Traffic Safety Administration (NHTSA) and subsequently improved by EDC [4-7]. EDSMAC uses a set of assumed or estimated parameters, including vehicle and roadway properties to predict the outcome of a collision. Engineers have been using this simulation program to analyze vehicle dynamics and the damage resulting from crashes. Researchers have found that the program yields reasonable results with sound input data [8-13].

EDSMAC analyzes the longitudinal and lateral movements of vehicles as well as the rotational motion about the vertical axis of vehicles on a horizontal plane. If a contact between vehicles is detected, the collision phase is analyzed. The external forces can be applied either at the tire/road interface or between the vehicles.

In the collision phase of the simulation model, a force proportional to the amount of crush is exerted as the body of a vehicle is crushed. This is accomplished by dividing the vehicle's perimeter into equally spaced intervals. Each of these intervals forms a pie-shaped wedge having its focus at the center of gravity of the vehicle. The vehicle exterior is assumed to have homogeneous stiffness. By knowing where the vehicles are with respect to each other,

EDSMAC locates the wedges that are in contact and equalizes the force between them. The resulting summation of forces dictates the motion of each vehicle due to the collision. This process continues for each collision time increment until the vehicles are no longer in contact.

EDSMAC allows the direct entry of vehicle data by users or the selection of default values. The vehicles are categorized by their wheelbase lengths into several classes. Class I and II are small passenger cars while Classes III to V are medium to large cars. In this paper, default values of vehicle parameters provided by EDSMAC are used in the simulation during this study. [7]

Attempts were made at the beginning of MOU-324 to acquire the source codes of SMAC directly from NHTSA. Since the program was developed more than 20 years ago, efforts to track down and locate the source codes were not successful. Through a series of contacts, a copy of the source code was finally obtained from the University of Michigan Transportation Research Institute with the help of Charlie Compton and Joel MacWilliams of University of Michigan, Traffic Research Institute (UMTRI). The source code was written in FORTRAN with limited revisions by UMTRI researchers added to the original version.

3.0 STRUCTURE AND FLOW CHART OF SMAC

The SMAC program is separated into several main parts:

1. Input phase: This portion receives the information regarding the integration and output time steps, the vehicle parameters, state variables such as position and speed, steering and tire torque inputs from an input file.
2. Trajectory phase: A subroutine and associated functions calculate the trajectories of the subject vehicles while the vehicles are not in a collision.
3. Collision phase: The collision routines determine if the vehicles are in contact and calculate the contact forces and its direction, which is then used to determine the subsequent motions of the vehicles.
4. Output phase: The states of the vehicles are printed at specified intervals, and also plotted on the terminal display if desired.

The plotting routines in the output phase is removed from the current version since it was originally designed to work with a plotting library on a mainframe computer. The plotting features can be added back if needed for a particular type of operating systems and their graphic libraries.

The main sequence and flowchart of the program is depicted in Figure 1. The major subroutines and their related functions are grouped in the associated blocks.

3.1 Descriptions of Program Subroutines and Sequences

The main portion of the program is separated into several sections: Input, Trajectory, Collision, and Output, as indicated in Figure 1. This section describes the structure and the sequence that these subroutines are related.

3.1.1 Initiation - SMACPATH

The main portion of the original SMAC program is converted into a subroutine SMAC. A short initiation program, SMACPATH, is constructed as the main program. In SMACPATH, the files are open and ready for the program before SMAC is called.

3.1.2 Main Subroutine – SMAC

SMAC called several subroutines in the following sequence:

A. SMAC calls INPUT and CNSTNT to initialize constants and variables.
B. SMAC checks if there is contact between two vehicles.
C. SMAC calls RNGKT1 to integrate the motion of the vehicles for each time step.
D. SMAC calls ACCEL and SAVMAX to update the acceleration and the direction of impact.
E. SMAC calls OUTPUT if it is time to print.
F. SMAC checks the variables and simulation conditions for error handling.
G. SMAC ends the simulation if the final time has been reached or if other conditions have been met.

Some of the major subroutines are explained as follow:

(1) INPUT
INPUT reads input information from an input file, INPUT.DAT. The details of the input file is given in the next section.
(2) ACCEL
ACCEL uses the change in linear and angular velocities to calculate the longitudinal and lateral components of vehicle accelerations.
(3) SAVMAX
In the process of calculation, SAVMAX saves those maxima that are followed by resultant acceleration less than 1.0G and then followed by resultant acceleration greater than 1.0G. At the end of run, SAVMAX arranges in order of decreasing maxima and computes the principal direction of impact. SAVMAX also calls subroutine DELTAV, which computes the velocity change of the vehicle center of gravity (CG) during a collision.
(4) RNGKT1
RNGKT1 is a fourth-order Runge-Kutta integration subroutine. A subroutine, DAUX, is supplied by users to describe the differential equations.
(5) DAUX
DAUX calls TRAJ to compute vehicle trajectories and COLL if vehicles are in a collision.
(6) TRAJ
TRAJ uses wheel inputs to calculate forces and torques applied to the vehicles.
(7) COLL
COLL checks around the profiles of the vehicles to see if contact is made. If so, calculates the relative speeds and forces. COLL calls PSBRAN, TORDER, SETIND. For details of the calculation in (6) and (7), please refer to McHenry, 1971 [1].
(8) OUPUT
OUTPUT produces a formatted output to OUTPUT.DAT. OUPUT calls other subroutines OUT2, RNGDAM, DAMAGE, NCOLDV. RNGDAM in turns calls ADJEND. DAMAGE calls FULSRC, CORSRC, DEFMJ. Most of these subroutines post-process the outcome of the collisions and computes the damage range, sum of velocity change, etc. The range of damage indicates the width, depth, and location of damage occurred to each vehicle.

3.2 Input File

The input file is assigned a name of 'input.dat' in the main program, SMACPATH. A sample file is shown in Figure 2. Except lines 1 and 2, each line in the input file has an integer entry on columns 73-80. The integer indicates the entry row number to the input subroutine so that the inputs are read into the proper parameters and variables. As a result, the lines in the input files can be entered in altered orders as long as the entry row numbers are associated with the correct contents. The descriptions of the contents are as follow:

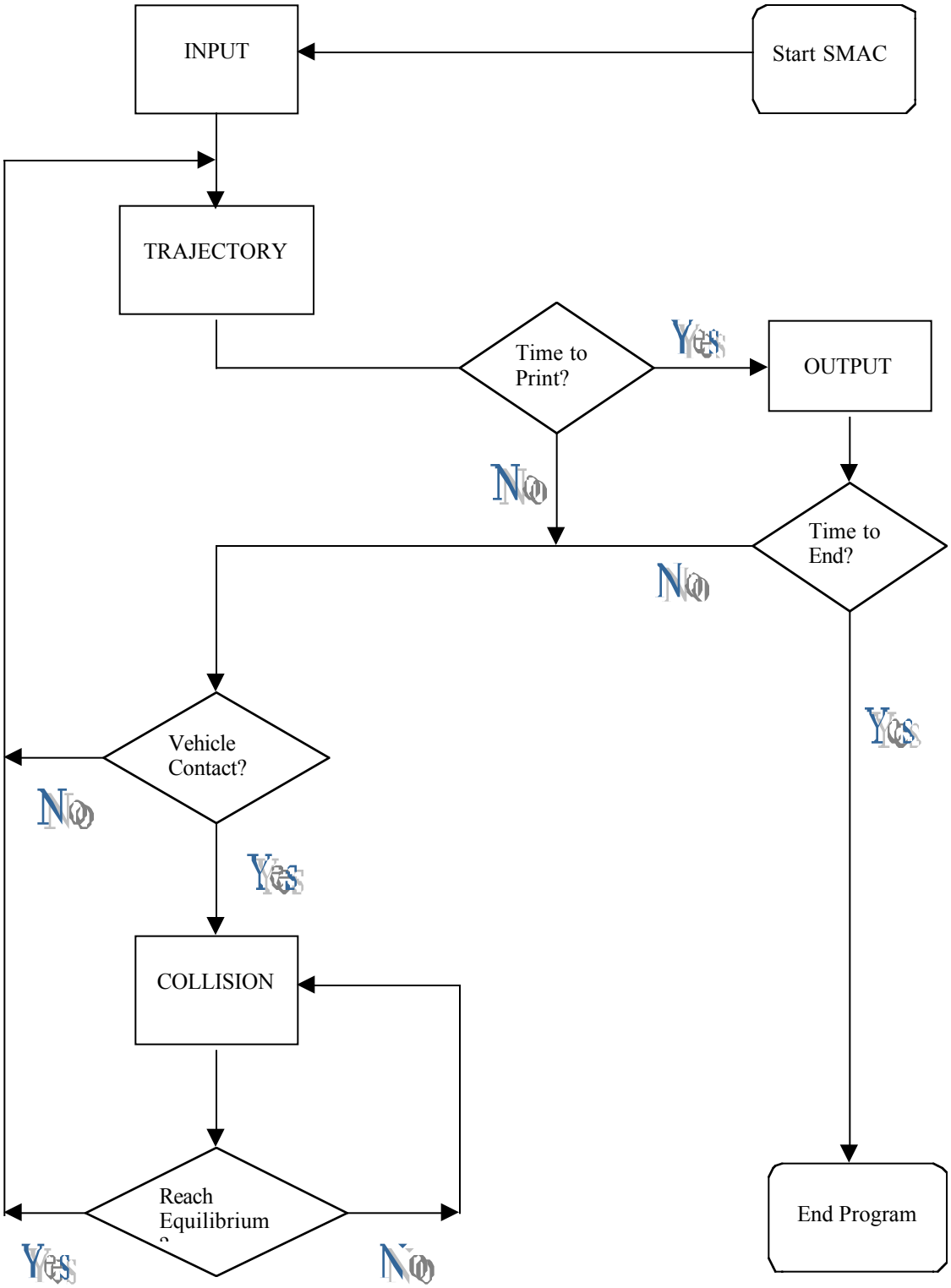
Line Sequence	Row Entry No.	Contents
1-2	N/A	Text Description of simulation Format: characters up to 80 per row
3	1	T0: begin time; all times in seconds TF: final time DTTRAJ: integration time step for trajectory phase DTCOLL: integration time for collision phase DTCOLT: integration time step following collision DTPRNT: time step for print out UVMIN: minimum velocity to terminate simulation, inch/sec PSIDOT: min. angular velocity to terminate simulation, deg/sec No. of vehicles: 1 or 2 Format: 9 floating numbers with each entry occupying 8 columns
4	2	Vehicle 1: XC10: initial x position, inch YC10: initial y position, inch PSI10: initial yaw angle, deg PSI1D0: initial yaw rate, deg/sec U10: initial longitudinal velocity, inch/sec

		V10: initial lateral velocity, inch/sec Format: 6 floating numbers with each entry occupying 8 columns
5	3	Vehicle 2: similar to the row above (replace 1 by 2 in variable names)
6	4	Vehicle 1: A1: distance CG to front axle, inch B1: distance CG to rear axle, inch TR1: track width, inch I1: yaw inertia, lb-sec-sec-inch M1: mass, lb-sec-sec/inch PSIR10: rear axle angle if damaged, deg XF1: distance CG to front end, inch XR1: distance CG to rear end, inch YS1: distance CG to side, inch Format: 9 floating numbers with each entry occupying 8 columns
7	5	Vehicle 2: similar to the row above (replace 1 by 2 in variable names)
8	6	Vehicle 1: C(1): cornering stiffness for right front tire, lb/radian C(2): cornering stiffness for left front tire C(3): cornering stiffness for right rear tire C(4): cornering stiffness for left rear tire Format: 4 floating numbers with each entry occupying 8 columns
9	7	Vehicle 2: C(5): cornering stiffness for right front tire, lb/radian C(6): cornering stiffness for left front tire C(7): cornering stiffness for right rear tire C(8): cornering stiffness for left rear tire Format: 4 floating numbers with each entry occupying 8 columns
10	8	Vehicle 1: TBTQ1 (initial time for torque input, sec) TETQ1 (final time for torque input, sec) TINCQ1 (time increment for torque input, sec) NTBLQ1 (if not 0.0, do not read table) Format: 4 floating numbers with each entry occupying 8 columns Note: The first 3 variables decide the number of entries in the table.
11	N/A	Following the row above, 7 entries per row for right front wheel with minimum of 3 and maximum up to 201 entries Format: 7 floating numbers with each entry occupying 10 columns
12	N/A	Similar to the row above with entries for left front wheel
13	N/A	Similar to the row above with entries for right rear wheel
14	N/A	Similar to the row above with entries for left rear wheel
15	9	Vehicle 2: similar to row 8 above
16-19	N/A	Similar to row 12-15 above
20	10	Vehicle 1: TBPSF1 (initial time for steer input, sec)

		TEPSF1 (final time for steer input, sec) TINCP1 (time increment for steer input, sec) NTBLP1 (if not 0.0, do not read table) Format: 4 floating numbers with each entry occupying 8 columns Note: The first 3 variables decide the number of entries in the table.
21	N/A	Following the row above, 7 entries per row for right front wheel with minimum of 3 and maximum up to 201 entries Format: 7 floating numbers with each entry occupying 10 columns
22	N/A	Similar to the row above with entries for left front wheel
23	11	Vehicle 2: similar to row 10 above (replace 1 by 2 in variable names)
24-25	N/A	Similar to row 22-23 above
26	12	XBP(1): points defining terrain zones, inch YBP(1) XBP(2) YBP(2) XMU1: friction coefficient in zone 1 at zero speed XMU2: friction coefficient in zone 2 at zero speed CMU: coefficient of linear decrement of friction with tire speed FMOVIE: not used in this version Format: 7 floating numbers with each entry occupying 10 columns
27	13	DELPSI: interval between radial vectors, deg DELRHO: increment change in radius vector, inch LAMBDA: acceptance error in equilibrium, lb/inch ZETAV: minimum relative velocity for friction, inch/sec KV1: load-deflection characteristics, vehicle 1, lb/inch-inch KV2: load-deflection characteristics, vehicle 2, lb/inch-inch MU: inter-vehicle friction coefficient Format: 7 floating numbers with each entry occupying 8 columns
28	14	C0, C1, C2: coefficients of assumed parabolic variation of coefficient of restitu with deflection Format: 3 floating numbers with each entry occupying 8 columns
29	15	PSILIM1: PSIB Range Test, Collision Criteria, default 70 degrees PSILIM2: default 110 degrees PSILIM3: default 250 degrees PSILIM4: default 290 degrees PSILIM5: PSIB1 for RHOB1 Test, default 10 degrees PSILIM6: default 170 degrees PSILIM7: default 190 degrees PSILIM8: default 350 degrees Format: 7 floating numbers with each entry occupying 8 columns
30	16	ITRAJ: trajectory option, 0 for straight road, 1 for standard curved road, and 2 for specified trajectory given in a separate file; standard curved road consists of a straight section followed by a circular arc XC: x-coordinate of center of circular arc in inches YC: y-coordinate of center of circular arc in inches RWYRAD: radius of curvature of circular arc in inches

		Format: 4 integers, each occupying 8 columns
30	17	IOPTION: determine types of steering inputs (to be explained further in the next section) IOPT1: option for vehicle 1, if IOPT=0, no steering input for vehicle 1 IOPT2: option for vehicle 2, if IOPT=2, no steering input for vehicle 2
31	10001	End of input file

Figure 1. Flow Chart of Simulation Program



```
SMAC INPUT EXAMPLE FILE

MODIFIED FOR PATH ON 11-12-1997

0.0     5.0     0.025   0.001   0.005   0.100   36.0    10.0    2.0     1

0.0     0.0     5.0     0.0     600.0   0.0                             2

500.0   0.0     60.0    0.0     0.0     0.0                             3
52.7    54.8    57.7    22424.  8.51    0.0     85.7    -100.0  35.7    4
57.3    59.7    60.0    29560.  9.86    0.0     94.8    -110.8  38.4    5
-10250  -10250  -10195  -10195                                          6
-10250  -10250  -10195  -10195                                          7
0.0     5.0     1.0     -0.1                                            8
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0     5.0     1.0     0.0                                             9
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0     5.0     1.0     0.0                                             10
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0     5.0     1.0     0.0                                             11
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
  -100.0  -100.0  -100.0   100.0     0.7     0.7     0.0               12
     2.0     0.2    15.0     5.0    50.0    50.0   0.550     0.0        13
 4606E-517547E-716711E-9                                                14
1       600     15748   15748                                           16
4       1       0                                                       17
                                                                        10001
```

Figure 2. A Sample Input File

3.3 Added Options

These added features are implemented for the purpose of ongoing development work at PATH. The addition of these options allows the studies of vehicle dynamics and control algorithms in collisions.

3.3.1 Row Entry 16
In row entry 16, an integer, ITRAJ, is used to indicate the type of trajectories. The options are:

A. ITRAJ = 0
Straight Road Option.
B. ITRAJ = 1
Run a simulation with a curved road option. The standard curved road is configured with a straight section thena circular arc. In this case, ITRAJ is followed 3 more integers that indicate the x and y coordinates of center and radius of the circular arc.
C. ITRAJ = 2
The designated trajectory is given in a file, DESTRAJ.DAT, which contains two columns of data for x and y coordinates of the trajectory.

In curved-road scenarios, a compensation scheme for control gains is deployed to minimize steady-state errors in curved sections. The following lines of codes indicate the entries and the default values for the compensator. [22]

```
C         USE DEFAULT FREQ FOR COMPENSATOR
C         WRITE(6,*) ' RUNNING CURVED ROAD OPTION'
C         WRITE(6,*) ' SELECT COMPENSATOR FOR POSITION FEEDBACK'
C         WRITE(6,*) ' (S + 2*PI*F1)/(S + 2*PI*F2) '
C         WRITE(6,*) ' DEFAULT: F1 = 0.1, F2 = 0.02 HZ'
C         WRITE(6,*) ' ENTER NEW VALUES FOR F1 AND F2: '
C         READ(5,*) FREQ1, FREQ2
C         FREQA = FREQ2*TWOPI
C         FREQB = FREQ1*TWOPI
```

3.3.2 Row Entry 17

In row entry 17, an integer, IOPTION, is used to indicate a few added options of executing the simulation. The options are:

D. ICNTRL = 1
Run a simulation with vehicle 1 and a step input of steering at the front wheels. The program will ask interactively for a steering angle. The step input starts at 0.25 second into the simulation.
E. ICNTRL = 2
Run a simulation with vehicle 1 and a sinusoidal input of steering angle at the front wheels. The program will ask for the frequency and amplitude of the input. The sinusoidal input starts at 0.25 second into the simulation.
F. ICNTRL = 3
Run a simulation with a feedback controller implemented in subroutine CNTRL.
G. ICNTRL = 4
Run a simulation with feedback controllers with time delay, also implemented in subroutine CNTRL.

The codes in the following inserted section show the types of control variables that can be entered interactively. In options 3 and 4, the control gain and look-ahead distance are required. [22] The bias in each wheel caused by collision damage along with time delay and sensor noise can be entered here as well. [23]

```
IF (IOPTION .EQ. 1) THEN
        FVEH0 = 1.0
        DTINT = 0.25
        WRITE(6,*) ' ENTER STEERING STEP INPUT PARAMETERS'
        WRITE(6,*) ' AMPLITUDE (DEG) AND TIME TO GO UP TO AMP (SEC):'
        READ(5,*) STEPPSI, STEPDT
    ELSEIF ( IOPTION .EQ. 2 ) THEN
        FVEH0 = 1.0
        DTINT = 0.25
        WRITE(6,*) ' ENTER STEERING SINUSOIDAL INPUT PARAMETERS'
        WRITE(6,*) ' AMPLITDUE (DEG) AND FREQUENCY (HZ):'
        READ(5,*) SINPSI, SINFREQ
    ELSEIF ( IOPTION .EQ. 3 ) THEN
        WRITE(6,*) ' ENTER FEEDBACK CONTROL PARAMETERS'
        WRITE(6,*) ' GAIN (0.01-0.20) AND LOOK-AHEAD DISTANCE (M):'
        READ(5,*) GAIN, DISLKAHD
        WRITE(6,*) ' ENTER WHEEL MISALIGNMENT AFTER COLLISION'
        WRITE(6,*) ' DW1F, DW1R (CAR 1 FRONT AND REAR WHEEL)'
        READ(5,*) DW1F, DW1R
        WRITE(6,*) ' ENTER WHEEL MISALIGNMENT AFTER COLLISION'
        WRITE(6,*) ' DW2F, DW2R (CAR 2 FRONT AND REAR WHEEL)'
        READ(5,*) DW2F, DW2R
    ELSEIF ( IOPTION .EQ. 4 ) THEN
        WRITE(6,*) ' ENTER FEEDBACK CONTROL PARAMETERS'
        WRITE(6,*) ' GAIN (0.01-0.20) AND LOOK-AHEAD DISTANCE (M):'
```

```
            READ(5,*) GAIN, DISLKAHD
            WRITE(6,*) ' ENTER WHEEL MISALIGNMENT AFTER COLLISION'
            WRITE(6,*) ' DW1F, DW1R (CAR 1 FRONT AND REAR WHEEL)'
            READ(5,*) DW1F, DW1R
            WRITE(6,*) ' ENTER WHEEL MISALIGNMENT AFTER COLLISION'
            WRITE(6,*) ' DW2F, DW2R (CAR 2 FRONT AND REAR WHEEL)'
            READ(5,*) DW2F, DW2R
            WRITE(6,*) ' ENTER TIME DELAY (DIGITIZED CYCLE IN ACTUATION)'
            WRITE(6,*) ' DTDGT'
            READ(5,*) DTDGT
C       ASSIGN A SMALLER VALUE TO DTDGT FOR ARRAY STAORAGE
C       IN SUBROUTINE DLY
            DTDGT = DTDGT/10.
            IF (DTTRA0 .GT. DTDGT) DTTRA0 = DTDGT
            IF (DTCLT0 .GT. DTDGT) DTCLT0 = DTDGT
            TDGT = DTDGT
C       ADD SENSOR NOISE TO YAW ANGLE
            WRITE(6,*) ' ADD SINUSOIDAL NOISE TO YAW ANGLE SIGNAL'
            WRITE(6,*) ' ENTER MAGNITUDE OF NOISE (DEG)'
            READ(5,*) YMAGD
            YMAG = YMAGD*RAD
            WRITE(6,*) ' ENTER FREQUENCY OF NOISE (HZ)'
            READ(5,*) YFREQHZ
            YFREQ = TWOPI*YFREQHZ
         ENDIF
```

4.0 CONCLUDING REMARKS

This report summarizes the contents of SMAC source codes. A copy of the code is attached to this report. This documentation will allow future developers to utilize this program as a validation model or combine it with other simulation tools.

This work is part of the research project conducted under MOU-324 and an extension of MOU-252. Readers can refer to previous publications [16-21] for findings using this simulation tool. Future work of this project includes the testing of control algorithms in certain collision scenarios.

ACKNOWLEDGMENT

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 paper reflect the views of the author who is 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.

The author would like to express special thanks to Charlie Compton and Joel MacWilliams of University of Michigan, Traffic Research Institute (UMTRI) for their help in providing the source codes of SMAC. I would also like to thank Seymour Stern of National Highway Traffic Safety Administration (NHTSA), who directs me to Joel after a search for the source codes at NHTSA was unsuccessful.

A commercial PC-DOS version of EDSMAC was provided by Engineering Dynamics Corporation (EDC) of Beaverton, Oregon, to PATH at no cost. Thanks should go to Terry Day, who offered the program and provided helpful instructions.

REFERENCE

1. McHenry, R.R., "Development of a Computer Program to Aid the Investigation of Highway Accidents," Calspan Report No. VJ-2979-V-1, DOT HS 800 821, NTIS-PB 208-537, 1971.
2. Solomon, P.L., "The Simulation Model of Automobile Collisions (SMAC) Operator's Manual," US DOT, NHTSA, Accident Investigation Division, 1974.
3. Noga, T., Oppenheim, T., "CRASH3 User's Guide and Technical Manual," U.S. DOT, January, 1981.
4. T.D. Day, R.L. Hargens, "Differences between EDCRASH and CRASH3," SAE Paper No. 850253.
5. Day, T., Hargens, R., "An Overview of the Way EDSMAC Computes Delta-V," SAE Paper No. 880069, Society of Automotive Engineers, 1988.
6. Engineering Dynamics Corporation, EDCRASH, "Reconstruction of Accident Speeds on the Highway," Version 4.5, June 1989.
7. Engineering Dynamics Corporation, EDSMAC, "Simulation Model of Automobile Collisions," Version 2.4, May 1989.
8. Jones, I.S., "Results of Selected Applications to Actual Highway Accidents of the SMAC Reconstruction Program, " SAE Paper No. 741179, 1974.
9. Jones, I.S., "The Application of the SMAC Accident Reconstruction Program to Actual Highway Accidents," Proceedings of the Eighteenth Conference of the American Association of Automotive Medicine, 1974.
10. R.S. Smith, J.T. Noga, "Accuracy and Sensitivity of CRASH," National Center for Statistics and Analysis, NHTSA, January 1982.
11. A.K. Prasad, "CRASH3 Damage Algorithm Reformulation for Front and Rear Collisions," SAE Paper No. 900098.
12. "Collision Deformation Classification," SAE Technical Report J224 MAR80, March 1980.
13. Day, T., Hargens, R., "Application and Misapplication of Computer Programs for Accident Reconstructions," SAE Paper No. 890738, 1989.
14. R.R. McHenry, N.J. Deleys, "Vehicle Dynamics in Single Vehicle Accidents – Validation and Extensions of a Computer Simulation, " Calspan Report No. VJ-2251-V-3, NTIS-PB 182-663, 1968.
15. R.R. McHenry, et al. "Mathematical Reconstruction of Highway Accidents " Calspan Report No. ZM-5096-V-1, DOT HS 053-1-146, NTIS-PB 220-150, 1973.
16. C. Chan, "Studies of Collisions in Vehicle Following Operation by Two-Dimensional Impact Simulations," *Proceedings of the 1996 Annual Meeting of ITS America*, Sixth Annual Meeting, Houston, Texas, April 1996.
17. C. Chan, "Collision Analysis of Vehicle Following Operation in Automated Highway Systems," *Proceedings of the Third World Congress of Intelligent Transport Systems*, Orlando, Florida, October 1996.
18. C. Chan, "Collision Analysis of Vehicle Following Operations by Two-Dimensional Simulation Model: Part I - Effects of Operational Variables," *California PATH Research Report, UCB-ITS-PRR-97-4*, January 1997.
19. C. Chan, "Collision Analysis of Vehicle Following Operations by Two-Dimensional Simulation Model: Part II - Vehicle Trajectories with Follow-Up Maneuvers," *California PATH Research Report, UCB-ITS-PRR-97-5*, January 1997.
20. C. Chan, "Simulation of Vehicle Trajectories and Maneuvers in Vehicle-Following Collisions," *Proceedings of the 1997 Annual Meeting of ITS America*, Seventh Annual Meeting, Washington, D.C., June 1997.
21. C. Chan, "Studies of Vehicle Collisions by EDSMAC," *California PATH Research Report, UCB-ITS-PRR-98-11,* March 1998.
22. C. Chan, H-S Tan, "Automated Steering Control in Vehicle-Following Collisions, "*Proceedings of the 1999 Annual Meeting of ITS America*, Ninth Annual Meeting, Washington, D.C., April 1999.
23. C. Chan, H-S Tan, "Lane Tracking Control in Vehicle-Following Collisions, "*Proceedings of the 1999 American Control Conference*, San Diego, June 1999.