U.S. Department
of Transportation

**National Highway
Traffic Safety
Administration**

People Saving People

DOT HS 808 504

September 1995

Final Report

Run-Off-Road Collision Avoidance Countermeasures Using IVHS Countermeasures

TASK 4

Volume 2: RORSIM Manual

This publication is distributed by the U.S. Department of Transportation, National Highway Traffic Safety Administration, in the interest of information exchange. The opinions, fmdings and conclusions expressed in this publication are those of the author(s) and not necessarily those of the Department of Transportation or the National Highway Traffic Safety Administration. The United States Government assumes no liability for its contents or use thereof. If trade or manufacturers' name or products are mentioned, it is because they are considered essential to the object of the publication and should not be construed as an endorsement. The United States Government does not endorse products or manufacturers.

Contracting Officer's Technical Representative's Precis

This report provides a basis for disseminating the preliminary contract results on a timely basis resulting in the information being available before the contract final reports are produced. Research performed during the remainder of the contract may support and/or modify the results, therefore, the material contained in this report should not be consider to be final. The current schedule calls for the completion of this research project by the third quarter of 1999.

This program was developed by Battelle for the U.S. Department of Transportation, National Highway Safety Administration as part of Contract DTNH22-93-C-07023.

Technical Report Documentation Page

1 Report No DOT HS 808 504	2 Government Accesson No	3 Recipient's Catalog No

4 Title and Subtitle Run-Off-Road Collision Avoidance Countermeasures Using IVHS Countermeasures Task 4 Report - Volume 2: RORSIM Manual	5 Report Date September 5, 1995 6 Performmg Organization Code
7 Author(s) M. Koenig, D. Pape, D. Pomerleau	8 Performing Organization Report No.
9 Performing Organization Name and Address Robotics Institute Carnegie Mellon University 5000 Forbes Avenue Pittsburgh, PA 15213	10 Work Umt No (TRAIS)n code 11 Contract or Grant No DTNH22-93-C-07023
12 Sponsoring Agency Name and Address National Highway Traffic Safety Administration 400 Seventh Street, S.W. Washington, DC 20590	13 Type of Reporr and Penod Covered Final Report 10-93 to 9-95 14 Sponsoring Agency Code

15 Supplementary Notes

16 Abstract

The Run-Off-Road Collision Avoidance Using IVHS Countermeasures program is to address the single vehicle crash problem through application of technology to prevent and/or reduce the severity of these crashes.

This report documents the RORSIM computer simulation developed in Task 4. RORSIM is a PC program which simulates the combined effects of the dynamical properties of the vehicle, the response of the driver, sensor measurements, environmental conditions and in-vehicle countermeasure systems. RORSIM is an extension of the commercial program VDANL (Vehicle Dynamic Analysis NonLinear) which was developed by Science Technology, Inc. (STI).

This report is an operating manual that contains detailed instructions on the operation of RORSIM. Results obtained using RORSIM are contained in a companion volume, Task 4 Volume I.

17 Key Words Run-Off-Road Collision Avoidance Single Vehicle Roadway Departure RORSIM Operating Manual	18 Dlstnbutlon Statement Document is available to the public through the National Technical Information Service, Springfield, VA 22161

19 Security Classif (of this report) Unclassified	20 Security Classif (of tbls page) Unclassified	21 No of Pages 40 + appendices	22 Price $621,184

Reproductionof completed page authorized

TABLE OF CONTENTS

TABLE OF CONTENTS
(Cont.)

Page

LIST OF TABLES

Page

LIST OF FIGURES

Executive Summary

In support of the mission of the U.S. Department of Transportation's National Highway Transportation Administration (DOT/NHTSA) to ensure the safety of U.S. highway systems, the team of Carnegie Mellon University (CMU), Battelle and Calspan was awarded contract (DTNH22-93-R-07023), "IVHS Countermeasures for Run-Off-Road Crashes". The overall objective of this three-phase contract is to develop practical performance specifications for single vehicle roadway departure (SVRD), or Run-Off-Road (ROR) crash avoidance systems.

Phase I of the program was conducted over the period September 1993 to September 1995, with a scope of "Laying the Foundation". Phase I consisted of the following four tasks:

- Task 1: Establish ROR crash subtypes and causal factors by thoroughly analyzing the crash problem
- Task 2: Establish functional goals of candidate countermeasures based on intervention opportunities and mechanisms
- Task 3: Obtain basic operational, performance and functional data by performing hardware testing of existing technologies
- Task 4: Develop preliminary performance specifications based on critical factors and models of crash scenarios

As part of Task 4, Battelle developed a sophisticated computer simulation package called RORSIM (Run-Off-Road SIMulation program) to predict the dynamic response of the driver/vehicle/countermeasure system under a wide range of operating, roadway and environmental conditions. RORSIM is an extension of the commercial code VDANL (Vehicle Dynamic Analysis NonLinear), which was developed and is owned by Science Technology, Inc. (STI) in Hawthorne, California.

A key feature of RORSIM is a menu system that enables the user to "point and click" to establish a simulation scenario. Results of the simulation runs can be viewed with VDANL utilities, exported to a spreadsheet format, or exported to be displayed with a postprocessor. The postprocessor supplied with RORSIM displays a plan view of the vehicle over a user-specified time period. A strobe-like depiction is created which shows the vehicles's position and attitude at fixed time intervals within the display period.

RORSIM was used to support all of the Phase I tasks, and particularly to support the development of preliminary performance specifications for a Run-Off-Road countermeasures system. RORSIM was designed to be a convenient and user-friendly analysis tool for use by NHTSA engineers and others involved in collision avoidance research.

This document is an operating manual that contains detailed instructions on the operation of RORSIM. Results obtained using RORSIM are contained in a companion document, the Task 4 Final Report Volume **I. Quick start instructions for running RORSIM are in Appendix A.**

1.0 Introduction

The RORSIM simulation package was developed as a tool for establishing preliminary performance specifications for Single Vehicle Roadway Departure or Run-Off-Road (ROR) countermeasure systems. RORSIM can be used to estimate how effectively a countermeasure system can reduce ROR accidents while maintaining an acceptably low false-alarm rate.

1.1 RORSIM Software Overview

The RORSIM simulation suite comprises of three distinct programs: the menu interface program used to define a simulation scenario, the dynamic simulation program, and the post-processor program used to display results. All programs in the RORSIM simulation suite are DOS applications.

1.1.l Menu Interface Program

The simplest way to begin defining a situation to simulate is through the menu program. Here the variables that define what is going to be simulated are set. The speed of the vehicle, the design of the roadway, the weather, the sensitivities of the countermeasure systems, the reaction times of the driver, and many more values are selected in the menu program. These values are saved in a series of disk files when you FILE then SAVE FlLE on the main menu. When you select RUN! on the main menu, the menu program ends and calls the simulation program, RORSIM. RORSIM, in turn, reads all of the values that have been most recently selected in the menu program and runs the simulation. As the simulation is running, the program plots four output variables on the computer screen. When the simulation is finished, other variables can be plotted, or groups of variables can be written to disk files for further analysis.

While the menu interface program is normally used to call the simulation program, users who are more familiar with the program may also define a scenario by editing the parameter files directly and then immediately running the simulation program. All of the simulation input and output files are ASCII text files, and can be modified with a text editor. Some of the parameter files cannot be modified through the menu system; the only method to modify them is with a text editor. The most notable files in this category are the vehicle parameter files, which can be selected, but not modified, in the menu program.

1.1.2 Dynamic Simulation Program

The dynamic simulation program is an enhancement to the VDANL program, which is a commercial, general-purpose, rubber-tired vehicle simulation program owned by Systems Technology, Inc. in Hawthorne, California. The VDANL simulation allows modifications to its operation through the use of a USERMODULE program. Battelle has developed this program to customize VDANL for simulation of Run-Off-Road scenarios.

Many levels of detail are available for reviewing the results of a simulation. The simplest analysis of a single simulation is to look at the file MAXYPOS.DAT, which records the leftmost and rightmost positions of the vehicle's front tires during the simulation. It is written at the end of every simulation. This summary of data is useful when comparing the results of many simulations, but it provides no clues how the vehicle reached those positions. The best way to examine the events as they unfold is to use VDANL's own plotting capabilities, as described in the VDANL Manuals and highlighted in Section 4.1, Graphical and Numeric Output Options. For more detailed analysis, the time histories of one or more variables can be written to a text file and then imported to a spreadsheet, analyzed with a special-purpose program, or plotted with the post-processor. It is assumed in this manual that the user has access to the VDANL manuals. Therefore, this RORSIM users manual will focus primarily on understanding and using the features added by Battelle to VDANL.

1.1.3 Post-processor Program

A post processor is included with the RORSIM suite of programs. The trajectory of the vehicle over the roadway can be plotted. The position of the vehicle is shown in plan view in several positions with a fixed time interval between each position. The procedure for producing the plots consists of saving the required variables from the simulation program, running a program to convert this output into a plotting format, and presenting the data using AutoCAD.

1.2 Purpose of the Model

The primary motivation for developing the RORSIM simulation package was to provide a tool for establishing preliminary performance specifications for Run-Off-Road countermeasure systems. The approach used was to evaluate the performance of several proposed countermeasure systems and from those results derive performance specifications. Proposed countermeasure systems can be analyzed by several approaches (e.g., by implementation in actual hardware, by human subjects operating a driving simulator, and by desktop simulation of specific incidents). Each of these approaches provides a different type of information.

The main advantage of the RORSIM desktop simulation is that the circumstances of the scenario can be closely controlled and repeated with specific variations. Also, a scenario can be simulated until a crash, which is not feasible for the actual hardware. During Phase 1 of the Run-Off-Road project, RORSIM has been used for studies in which a key parameter is varied over a range to determine its effect on the outcome of a particular scenario. RORSIM has also proven useful for performance testing a vehicle in various near-emergency maneuvers. These tests have provided valuable data on yaw rates, lane closure rates, steering angles, and other parameters when roadway departure is imminent.

1.3 Organization of this Manual

This manual contains instructions for using RORSIM, including the menu interface, dynamic simulation, and the post-processor. Technical details are provided on how the model is configured, so users of the program can appreciate its limitations, capabilities, and can make enhancements. Derivations of the underlying formulas for the countermeasure models are provided in Volume I of this report [Pape, et al, 1995]. The VDANL software is discussed in an earlier technical paper prepared for NHTSA by Systems Technology, Inc. [Allen, et al, 1992].

Those who may use RORSIM to study the built-in proposed countermeasure systems will find Sections 3 and 4 of this volume most useful. These sections contain specific instructions for the scenario input and data output of the program. The installation and Quick-Start instructions in Appendix A will be helpful to all. Occasional reference to Section 2 may be necessary to understand the model's capabilities. If you want to modify RORSIM to model newly conceived countermeasure systems or to model scenarios not presently available, refer to Section 2 on program architecture, Section 5 on customizing, and the separate 3-ring binder.

Sufficient information on VDANL itself is presented to develop skills in plotting and saving the results. For complete details on VDANL, see its online help file, VDANL.HLP, and the VDANL user's manual [System Technology, Inc., 1989].

2.0 Dynamic Simulation Software Description

The dynamic simulation program, RORSIM, is an enhancement to VDANL. VDANL provides the vehicle dynamics model for the simulation as well as the closed-loop driver model. Battelle has developed enhancements to VDANL for use in evaluating Run-Off-Road countermeasure systems. These new capabilities include:

- crowned roadways with a banked curve,
- constant grade roadways,
- different roadway traction conditions,
- different sensor visibility conditions,
- drivers with different driving capabilities,
- momentary driver inattention,
- ROR countermeasure system characteristics, and
- various driver reactions, to countermeasure system outputs.

This section of the manual will discuss several of these features in further detail, allowing a deeper understanding of the model's operation. Because this is mostly details of what goes on inside RORSIM, you can skip straight to Section 3.0 if you're eager to get going.

2.1 Interfacing Enhancements to VDANL

Most of the software code for Battelle's RORSIM program is proprietary to STI, but the VDANL code was customized through the use of an "Open Module." When the program reaches certain locations in its execution, called breakpoints, it invokes a function called USERMODULE, which has been compiled and linked with VDANL.

Each breakpoint provides an opportunity to modify an aspect of VDANL's operation. For example, after VDANL has calculated the tire elevations and road surface properties according to its own internal procedure, it reaches the Terrain breakpoint. If that breakpoint is activated, it calls the USERMODULE function with the indication that it is currently at the Terrain breakpoint. The function can then adjust the tire elevations to simulate a different terrain profile. Execution of VDANL proper resumes with the modified elevations.

For a breakpoint to be active, it must be "turned on" at the beginning of the simulation through menu selections in VDANL. Because RORSIM requires several breakpoints and will not work properly unless all of them are turned on, VDANL should be run through the menu system or a macro whenever RORSIM functionality is desired.

All of Battelle's enhancements to VDANL, were implemented using the breakpoints. RORSIM uses several breakpoints, including terrain, lane and curvature errors, initialization, and end of run. Most of the actions are performed by code in the source file UMROR.BAS, which has separate sections of code to handle each breakpoint, The calculations of roadway elevation and surface properties are performed by a function in a separate source file, ROAD.BAS, which is called through USERMODULE. In addition, each countermeasure system comprises two source files, one for the sensor actions, and one for the decision making actions. The countermeasure implementation is discussed in greater detail in Section 2.5.

The communication of variable values between VDANL proper and the user module is primarily through two large arrays, which are in a common block defined in USERMOD.INC One array, VARYY ( ) , contains the entire state vector of the system.It holds all of the variables listed in VDANL.HLP, most of which are time-dependent. The other array, VDANLINP ( ) , contains the parameters read from FILES.DUM, which are essentially those that define the vehicle and driver. The RORSIM code references quantities in these arrays through subscripts defined in the include file VDANLVAR.INC. For example, VARYY (yyPSI) is the 45th element in the variable list, the heading angle of the vehicle. Several new variables, specific to the needs of RORSIM, such as the current TTD value, are also included in the VARYY ( ) array and defined in VDANLVAR.INC. The functions describing the road, sensors, decision making algorithms, and other Battelle-added features are declared in the include file named RORSM.INC.

2.2 Input Parameters

All of the simulation-specific parameters defining a simulation are read by RORSlM from disk files, which are in ASCII text format.The parameters required by VDANL itself are read from files named in FILES.DUM. This includes the entire definition of the vehicle. These files are read every time VDANL begins a new simulation.

After VDANL has read the parameter files named in FII.ES.DUM, it calls USERMODULE with the Initialize breakpoint indicated. The USERMODULE directs reading the RORSIM-specific parameters. All of these source files, each corresponding to a page in the menu system, are opened by USERMODULE. The parameters that USERMODULE needs are read first. Then a special initialize function in each source file is called to read the parameters needed in that source file.

In the menu system, the parameters are organized according to operator preferences regarding scenario input data. For example, dropout times for the sensors are specified on the same menu page as the inattention times for the driver. This arrangement makes it easy for the operator to schedule a sensor dropout at about the same time the driver is not paying attention. The menu system creates a file containing the parameters from each menu page. Each initialization function finds its own parameters within each file by searching for the line where the description of the function is enclosed in square brackets. For example, the roadway parameters in the road file and the environmental surface file follow the line, [Road]. The functions in READFILES.BAS perform the task of fetching and parsing the parameter lines.

The next to last line of each menu generated parameter file is an indicator of whether the file is “protected.” If it is protected, the file is one of the standard set of pre-defined conditions provided by Battelle with the RORSIM package. The menu system will not change a protected file. (Of course, these files, like the other parameter files, are text files and can be modified by any text editor.) The final line of each file is a verbal description of the conditions the parameters are intended to model. This description is displayed in the menu program on the main menu when the file is selected. Sample parameter files are listed in the accompanying 3-ring binder.

2.3 Simulation Features

Here are brief descriptions of how some special effects are modeled in RORSlM.

2.3.1 Vehicle Speed Control

Because the slip generated by the tires due to acceleration or braking affects the ability of the tires to produce lateral forces for controlling steering, it was essential that the speed of the simulated vehicle be controlled through the tires. Therefore, in the RORSIM program, the speed of the vehicle is adjusted through the throttle and brake pedals. The engine and drivetrain parameters used in the RORSIM model are typical for sedans, but they do not attempt to model the specific performance of a Ford Taurus. A simple speed controller was built in USERMODULE to actuate the throttle and brake pedals. The controller's primary function is to hold the vehicle's speed at the value commanded by the speed profile file. When the driver is inattentive to the vehicle's speed, the speed controller takes no action at all. There is a provision for a driver to brake in response to a surprise, even where that might not be an appropriate action. This situation is also handled by the speed controller, which does not alter acceleration or deceleration that might be applied by an active countermeasure system.

2.3.2 Roadway

Because complete coordination between the road elevation, surface properties, and curvature is essential for modeling the behavior of the countermeasure systems, the roadway properties are handled entirely through the VDANL, open module. A function in ROAD.BAS calculates the elevation and surface properties of each tire according to the vehicle's position and orientation at each integration step. The features that can be modeled are described in detail in Section 3.5, where the road menu page is presented. The instantaneous road curvature for the driver model must correspond to the vehicle's position in the roadway. It is read from a separate file than the roadway. To coordinate the two, the initialization function in ROAD.BAS creates a temporary file with the name TEMP#@@ ! . CRV, which has the curvature function in the proper format for the driver model.

2.3.3 Simulation Termination

When a vehicle departs its lane by more than a pre-determined distance, it is deemed to have crashed, and there is no point to continuing the simulation. The departure distance is defined as the error between the commanded and actual lateral lane position. The maximum tolerable error, Yemax, is set in the .AUX file. Because it does not appear on any menu, the only way to change it is to edit the file with a text editor. Because VDANL does not provide a method to terminate a simulation in the code, the simulation is stopped by applying an enormous force to the left suspension, which upsets the vehicle and ends the simulation. The last one or two data values recorded for a simulation that ended in a crash are usually invalid. This method of terminating execution occasionally causes a BASIC error, which terminates the entire program. If the program abruptly crashes when the vehicle is far off the road, Yemax can be adjusted by one foot or so in order to fix the problem in most cases. (A simulation can be stopped manually by pressing the Esc key.)

2.3.4 Driver Deadband

An attentive driver does not exactly follow the lane center. There is a certain amount of variation in lane position due to minor disturbances and a human's natural tolerance of small errors. Following the general approach of Carson and Wierwillie [1978], RORSIM has thresholds for heading error and curvature error. Errors below these thresholds are ignored. The threshold levels were selected so that the variance in lane position of a vehicle simulated in RORSIM traveling on a straight road is approximately the same as published values [Carson & Wierwille, 1978, Allen, et al, 1975]. However, because RORSIM does not have random disturbances, the bandwidth of the lane position function is less (i.e., the simulated lane position is a smoother curve).

2.4 Driver Inattention

A major cause of Run-Off-Road accidents [Calspan, 1994] is the driver's inattention to the driving task. Therefore, provision for modeling brief periods of driver inattention is essential to establishing a scenario for a potential Run-Off-Road accident.

There are two broad classes of countermeasure systems (e.g., longitudinal, which help the driver control the speed of the vehicle, and lateral, which help the driver steer the vehicle). Accordingly, RORSIM has the capability to model inattention to the speed-keeping task and inattention to the steering task. A simulation may have either one, both, or neither type of inattentive driver behavior.

The starting times for inattentive periods are specified before a simulation begins. An inattentive period ends at the time specified before the simulation or, if desired, when the driver reacts to a warning from a countermeasure system. The starting and ending times for inattentive periods for speed-keeping and steering can be set independently. When a warning is issued, both inattentive periods end after reaction times. (The driver is assumed to regain complete awareness of the driving situation, regardless of the source of warning.) Separate reaction times for steering and speed-keeping can be specified. Various alarm modalities (visual, aural, etc.) are not modeled explicitly. The modality of the alarm is assumed to affect only the driver's reaction time.

The driver's reaction time is a function of the driver's position in the assumed population (e.g., 20th percentile young, healthy male), the driver's current condition (e.g., alert), and the alarm modality (e.g., a driver may react faster to a shaking handwheel than to a low-intensity light). These separate effects are not included explicitly in the menu system; the operator merely selects a single steering reaction time and a single braking reaction time to account for whatever combined effects may be of influence.

When the driver is inattentive to steering, no adjustments are made to the handwheel position. In RORSIM, this specification is done by setting the curvature error to zero in the Curvature Error breakpoint. When a period of inattention to steering ends, the error signal is allowed to pass through the driver model. The steering correction is applied at a rate that depends on the driver parameters. When the driver is inattentive to speed, the throttle is held at its current position. In a simulation where the speed is to remain constant throughout, a period of inattention to speed will have little effect. Inattention to speed is critical when the simulation has a distance-dependent speed profile, which would be missed by an inattentive driver.

2.5 Countermeasure Systems

At present, three countermeasure systems are included in the RORSIM package. These systems are patterned after those developed in Task 3 of this project. Other systems with similar functions can be modeled by varying the describing parameters for the systems. Countermeasure systems which can not be accommodated with the existing code can be installed, according to Section 5, Customizing. The three systems are described briefly below.

The code for each countermeasure system is contained in two source files. As each prototype countermeasure system comprises a sensor and an algorithm for making a decision based on the output of that sensor, the systems are modeled as having two separate components. (A system, of course, could contain two or more separate sensors, but none are currently modeled that way.) Three countermeasure systems are modeled, forward-looking using Time to Trajectory Divergence (TTD), downward-looking using Time to Line Crossing (TLC), and Curve Warning. Thus there are a total of six source files for the three countermeasure systems.

Each source file contains two functions that are visible to USERMODULE, the initialization and the simulation function. They may contain purely internal functions as well. The initialization function is called once before the simulation starts. It reads the parameters that control the sensor or decision algorithm, and opens files that will be read during the simulation, if necessary. The parameters it reads from the files are shared with the simulation function through a common block that is private to the source module.

Every sensor’s signal is potentially subject to six kinds of degradation: dropout, bias error, random (zero-mean gaussian) error, process delay, finite update period, and finite output resolution. During sensor dropout, the output of the sensor remains constant. A bias error adds a constant offset to the sensors exact output and is constant throughout the simulation. The random noise, like the bias error, is added to the sensor’s exact output. The amplitude of the random noise set on the menu system is the standard deviation of a Gaussian noise signal. When process delay is selected, the function calculates the sensor output immediately, but does not make it available to **USERMODULE** until the end of the delay time. A finite update period forces the minimum time between successive changes of the sensor output. The output resolution is a quantization of the output and is applied after the other types of degradation, including the noise. The six kinds of degradation can be adjusted or disabled independently.

The amplitude-related degradation (bias, noise, or resolution) is performed by a function in DEGRADE.BAS. The function includes a random number generator for the random noise and is the only randomized quantity in RORSIM. All other quantities, including driver inattentive times and vehicle starting position, are completely deterministic and are established by the parameter files before the run starts.

The decision making algorithms examine the sensor output, apply the logic of the prototype countermeasure system, and, if necessary, issue a warning. The warning is generated by incrementing a variable WakeTheDriver. The different countermeasure systems apply a different increment according to a binary code, so the variable can be examined after a simulation to determine which system or systems have declared a warning at any time. The systems may add a specially coded increment if they deem that active intervention is necessary. Each countermeasure system has two thresholds (i.e., one for warning and one for intervention.) The curve warning applies the brakes when it intervenes. As the program is currently built, intervention by one of the lateral systems merely causes the driver to resume steering without reaction delay. The dynamics of an actuator is not modeled. The codes for issuing a warning or calling for intervention are listed in Table 2- 1.

Table 2-1
Coded Increments to the Warn Variables When
a Countermeasure System Senses Danger

System	Increment when issuing a warning	Increment when calling for intervention.
System #l (CMU Look-Ahead or TTD)	1	**10**
System #2 (CMU Look-Down or TLC)	**2**	**20**
System #3 (CMU Curve Warning)	**4**	**40**

The condition of the countermeasure systems can be determined after a simulation by examining the value of the variable labeled Warn. For example, if Warn has a value of 13 (1 + 2 + l0), then Systems #l and #2 issued a warning to the driver, and System #1 was calling for active intervention. When the value is zero, no warning is currently being issued. For convenience, separate warning indicators for each countermeasure system are also provided. When USERMODULE notes that Warn is nonzero, it begins reaction time counters for steering and speed-keeping, and, if possible, it begins taking the intervention action.

3.0 Menu System

The menu program was designed to be a convenient way to input or select the simulation parameters, with a main menu screen containing a summary of the simulation setup. The details on the input parameters are organized in groups on subgenus.

The menu program takes your input and creates a series of files on the computer disk that define a particular simulation. Selections are made with the mouse or keyboard entry. When using a mouse, place the mouse cursor over the desired selection and press the left mouse button. The arrow keys can also be used to move around the screen, to allow keyboard entry of values.

Table 3-l will help you find the parameters you're looking for. Appendix B has a sample of all the menus.

3.1 Starting the RORSIM Menu

To run the menu, type and enter RORMENU at the DOS prompt.

3.2 Main Menu

The main menu page displays a verbal description of the selections for each submenu file. The six subgenus accessible from the main menu are:

Vehicle
Roadway
Environmental Conditions
Driver Type
Simulation Scenario
Countermeasure System

To go down to a submenu to change, say, the driver characteristics, click on the button with the submenu's name, or push the function key listed next to the button. Instructions for operating the submenus are in Sections 3.3 (for the vehicle) and 3.4 (for all the others).

Three choices are located in the bar at the top of the page. The first is for loading or saving a file. It saves a file with all of the subfile names. It defines the entire simulation case, and it's the file that the menu program passes to the simulation program to run. (See section 3.6.) Its called an auxiliary file, and its extension is .AUX. You can also recall an entire case that you created yesterday using BILE I LOAD BILE. When you load a new file, the descriptions on the main menu page all show what you've selected.

Table 3-1
Where to Find Variables on the Menu System

	Vehicle	Roadway	Environment	Driver Characteristics	Scenario	Countermeasure System
VDANL Proper	vehicle files			response gain crossover frequency		
Main ROR function				reaction times deadbands	speed lane position inattention times nature of response	active control parameters
Roadway function		segment lengths lane widths crown superelevation curvature grade	friction rolling resistance			
TTD functions			noise error bias error		dropout times	processing time update period sensor resolution look-ahead time distance apart for TTD warning threshold intervention threshold
TLC functions			noise error bias error		dropout times	processing time update period sensor resolution believed lane width edge position warning threshold intervention threshold
GPS functions					dropout times	noise error bias error processing time update period sensor resolution warning threshold map resolution intervention threshold believed road properties
File name extension	FILES.DUM	.RD	.NVS (surface) .NVV (visibility)	.DVC	.SCN	.CMS

The next choice in the bar is RUN!. If you click this, the menu program will call the simulation program and run a 30-second simulation with the parameters you selected. If you've made changes but haven't saved your auxiliary file, RORMENU will ask if you want to give your case a name. If for some reason, the simulation program can't run, you'll stay in the menu program. If the simulation does run, you will not come back to the menu; you're just returned to the DOS prompt when you quit RORSIM.

The final choice in the bar is QUIT!. Click here if you want to leave the menu program and return to DOS. As with the RUN! button, if you've made changes but haven't saved your auxiliary file, RORMENU will ask if you want to give your case a name.

3.3 Vehicle

The Vehicle submenu lets you specify the three files that define the vehicle. These three files are used by VDANL and contain vehicle parameters, suspension parameters, and tire parameters. The three tiles included with the RORSIM package define a Ford Taurus, If you want to make files for other vehicles, use a text editor and follow the instructions contained in the VDANL manuals. See also Appendix E of Allen, et al [1972] Once developed, they can be used in the RORSIM package by entering their filenames and paths in the Vehicle menu.

3.4 RORSIM Submenus

All the subgenus (except the one for the vehicle) have the same basic appearance and operation. In each case a window shows the previously defined files, their descriptor, and the date of creation. There is a scroll bar on the right side of the file window to scroll through the list of files. Just below the file window is a text window that indicates which file is currently selected for use in the simulation and an arrow points to that file in the file window listing. The specific values shown below the file window are the simulation parameter values for the selected file. Your choices are as follows:

- Select a different file: Scroll through the list of files. To select a file double click on the file desired, single click to highlight the file and press the SELECT button, or single click to highlight the file and press the ACCEPT button. When the double click or SELECT options are used, the text box will show the selected file, the arrow in the files window will move to the selected file, and the parameters for the newly selected file will be displayed. If the ACCEPT option is used, you go back to the main menu. The description of the newly selected file is shown in the description window.

for a particular countermeasure system, click on the appropriate button). To modify any value, first highlight the desired value box by clicking the mouse on the number. Edit the exiting value using the keyboard. This new parameter set must be saved. To save the file click on the ACCEPT button. A window with three options is presented. The first option is to overwrite the file with which the modification began. This option is available only if the original file is not a protected file. Protected files can not be overwritten. The second option is to save the file under a new name. If this is selected, you will be prompted for a filename (8 characters maximum) and a description, which is used on the main menu description. The third option is to cancel the operation.

- Generating a new file: The only method for generating a new file is to modify an old file and save under a new file name. See Modify an existing file.

- Delete a file: Any non-protected file can be deleted. To delete a file, highlight it with a single click (do not select the file with a double click). Click on the DELETE pushbutton.

- Canceling the changes: By clicking the CANCEL button, the program will ignore the changes to the file being modified and return the operator to the main menu. (Note: files which have been deleted are not restored by the cancel command.)

3.4.1 Roadway

The Roadway submenu contains the parameters which are directly involved with the terrain. There are four protected roadway files supplied with the simulation. These are listed in Table 3-2:

Table 3-2
Protected Roadway Files

Road File Name	Description	Date Created
STRAIGHT.RD	Straight, crowned road	8/10/95
CURV0800.RD	**800** ft curve to right	8/10/95
CURV1000.RD	1000 ft curve to right	8/10/95
CURV4000.RD	**4000** ft curve to right	8/l0/95

The roadway is described by 13 parameters. The roadway available for use with RORSIM is a two lane road with a shoulder on each side of the road. The roadway always consists of a straight section, a transition into the curve, a curve section, a transition out of the curve, and a final straight section. Figure 3-l shows the plan view of a roadway. The figure shows a 90-degree curve to the right, which is positive curvature. Negative curvature is to the left. The curve can be more or less

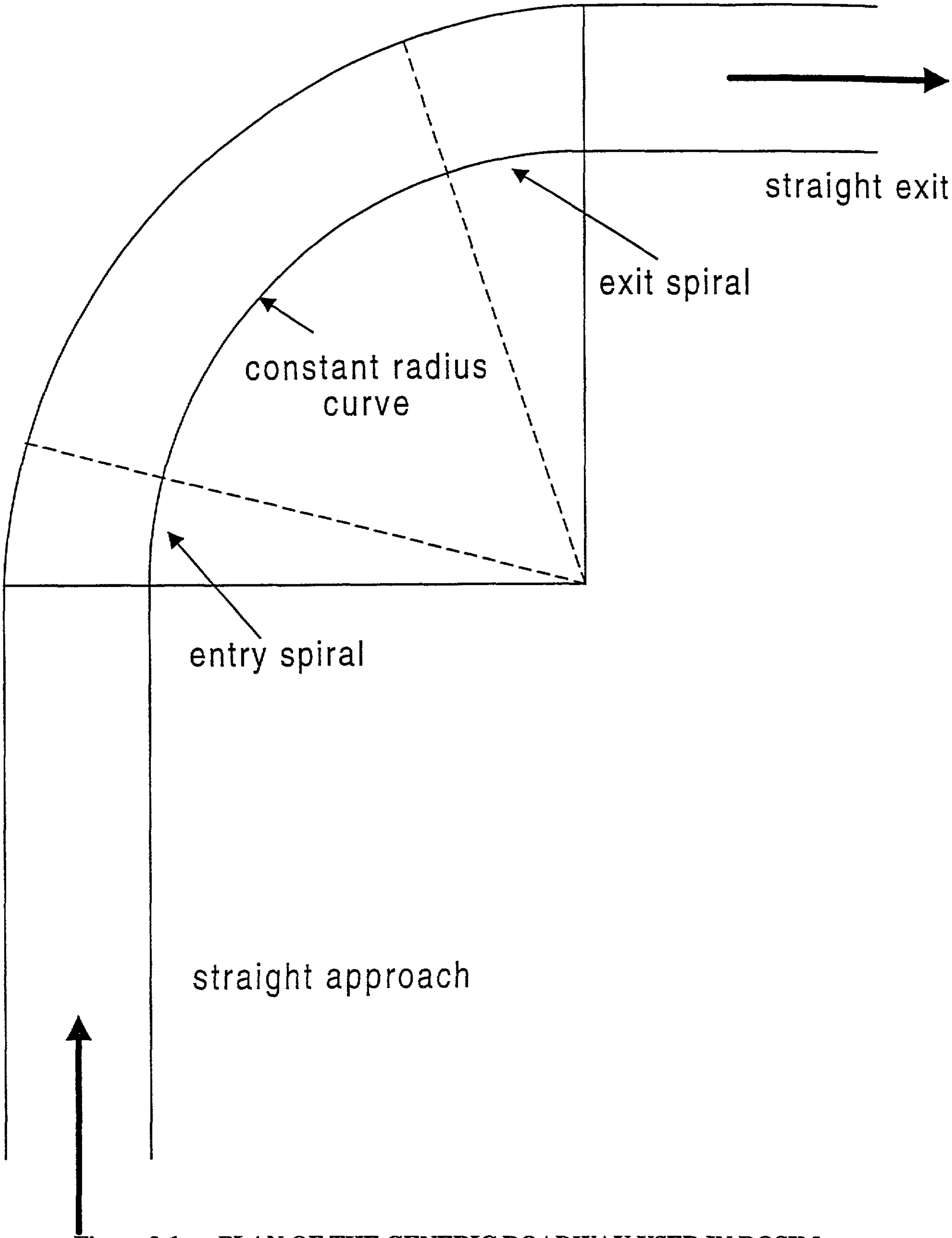


Figure 3-1 PLAN OF THE GENERIC ROADWAY USED IN ROSIM

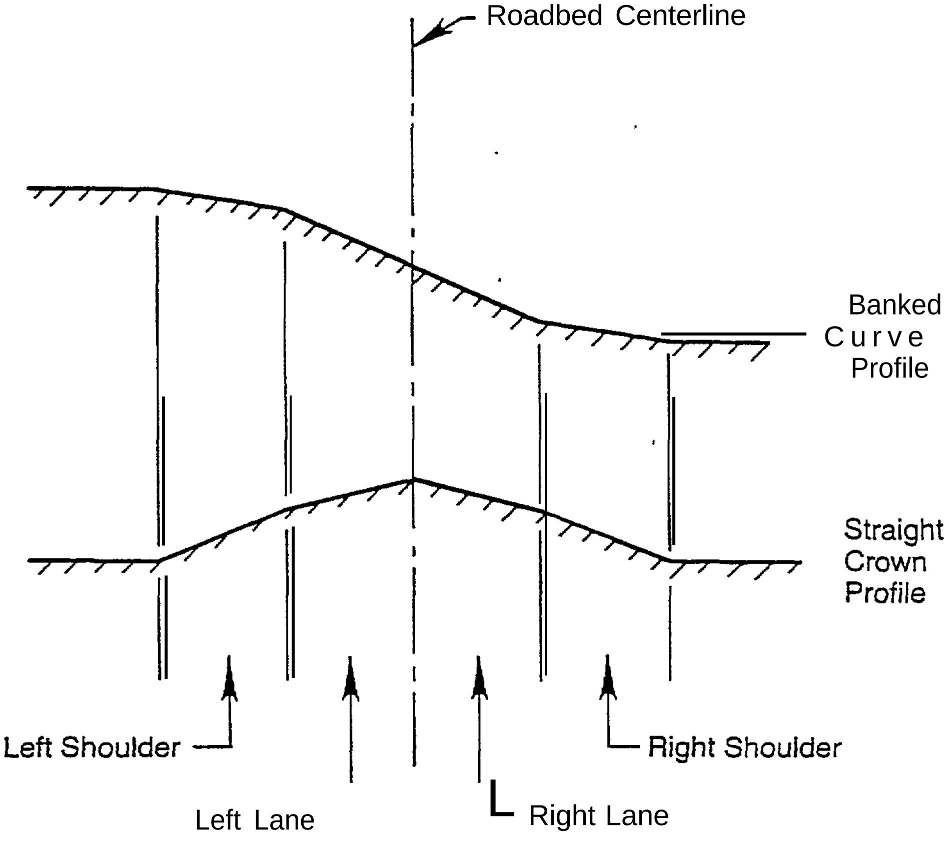
than a 90-degree curve. To simulate a straight road, specify the initial straight section to be very long. The width of the left and right lane widths are always equal. The shoulder widths are also equal. In the straight sections the slope of the lanes is equal and opposite (peak of crown at the centerline of the road). In the curve section, both lanes are of equal slope and both shoulders are of equal slope. The centerline of the roadway is a constant elevation unless a constant grade is specified, Figure 3-2 shows the slopes of the roadbed and shoulder.

The transition sections contain changes in both roadway slope and curvature. For the transition into the curve, the roadway curvature varies linearly along the length of the transition section from a curvature of zero in the straight section to the curvature specified for the curve. The elevation of the roadbed at any offset from the centerline is interpolated linearly between the elevation at that offset at the end of the straight section and at the beginning of the curve section.

Table 3-3
Parameters Input on Roadway Submenu

Parameter Name	Units	Description
Straight Length of Road before the Transition Section	ft	Length of straight roadway before the transition and curve sections.
Length of Transition Section of Roadway	ft	Length of transition section into curve.
Length of Curve	ft	Length of the constant curvature section of the roadway.
Length of Transition after the Curve	ft	Length of the transition section after the constant curvature curve.
Length of the Straight Road after the Curve	ft	Length of the straight section following the curve and exit transition.
Lane Width	ft	Width of the road lanes. The two lanes are of equal width.
Shoulder Width	ft	Width of the shoulder lanes. The two shoulders are of equal width.
Slope of the Road Surface in the Straight Sections	ft/ft	Slope of the roadbed in a direction perpendicular to the centerline of the road. In the straight sections, the slope of the two lanes is equal and opposite. A positive value for the slope creates a roadbed that is crowned. A negative value for the slope creates a roadbed that is dished.

Parameter Name	Units	Description
Slope of the Road Surface in the Curve Section	ft/ft	Slope of the roadbed in a direction perpendicular to the centerline of the road and is also known as superelevation. In the constant curvature section, the slope of the two lanes is equal. When viewed in the direction of travel, a positive slope gives a roadbed slope which is high on the left side and low on the right side. A negative slope gives a roadbed which is high on the right side and low on the left side. (Note: a positive slope is typical for a positive curvature).
Slope of the Shoulder Surface in the Straight Sections	ft/ft	Slope of the roadbed in a direction perpendicular to the centerline of the road. In the straight sections, the slope of the two shoulders is equal and opposite. A positive value for slope causes the shoulders to slope away from the edge of the road. A negative value for slope causes the shoulders to slope toward the edge of the road.
Slope of the Shoulder Surface in the Curve Section	ft/ft	Slope of the roadbed in a direction perpendicular to the centerline of the road. In the constant curvature section of the roadbed, the slope of the two shoulders is equal. When viewed in the direction of travel, a positive value for slope causes the shoulders to be high on the left side and low on the right side. A negative value for slope causes the shoulders to be low on the left side and high on the right side.
Curvature	ft/ft	The curvature defines the sharpness of the curve. Its value is the reciprocal of the radius of the curve. A positive value for curvature generates a curve to the right, a negative value for curvature generates a left hand curve.
Percent Grade	%	A constant grade can be applied to the road surface. Positive grade is an uphill grade, negative grade is a downhill grade.



For Straight Crown:

Slope of right lane = -slope of the left lane
Slope of right shoulder = -slope of left shoulder

For Banked Curve:

Slope of right lane = +slope of left lane

Slope of right shoulder = +slope of left shoulder

Figure 3-2 SECTIONS OF THE GENERIC ROADWAY USED IN RORSIM (ELEVATIONS ARE GREATLY EXAGGERATED)

3.4.2 Environmental Conditions

The environmental conditions menu allows input of specific roadway and visibility conditions. Five protected files are provided with the RORSIM package and contain typical values for these parameters. Table 3-4 lists these protected files:

Table 3-4
Environmental Condition Protected Files

Road File Name	Description	Date Created
NORMAL.NVS	Normal, dry pavement	8/10/95
WET.NVS	Moderately wet pavement	8/10/95
SLIPPRY.NVS	Freshly rained-on pavement	8/10/95
CLEARDAY.NVV	Clear, sunny day	8/10/95
FOG.NVV	Fog - TTD noise error	8/l0/95

The parameters specified in the Environmental menu are included in Table 3-5:

Table 3-5
Parameters Input on Environmental Submenu

Parameter Name	Units	Description
Roadway Coefficient of Friction	Non-dim. (nd)	This is the coefficient of friction for the road to tire interface. Typical values for this are from 0.2 for ice to 0.9 for tires on dry pavement. Reference [Dijks, 1974]
Shoulder Coefficient of Friction	nd	This is the coefficient of friction for the shoulder to tire interface. Typical values are from 0.2 for ice to 0.9 for tires on dry pavement.
Runout Area Coefficient of Friction	nd	This is the coefficient of friction for the runout area to tire interface. The runout area covers all the area outside the shoulders. Typical values are from 0.2 for ice to 0.9 for tires on dry pavement.
Roadway Rolling Resistance	nd	This is the coefficient of rolling resistance for the tire to road interface. A typical value for this variable is -0.015.
Shoulder Rolling Resistance	nd	This is the coefficient of rolling resistance for the tire to shoulder interface. A typical value for this variable is -0.015.

Parameter Name	Units	Description
Runout Area Rolling Resistance	n d	This is the coefficient of rolling resistance for the tire to runout area interface. A typical value for this variable is -0.015.
CMU Look Ahead (TTD) Countermeasure System Output Noise Error	ft	This is the location for inputting the random noise expected from the position sensor. It is included on the environmental submenu since it is largely due to degradation of performance as a result of weather/visibility conditions.
CMU Look Ahead (TTD) Countermeasure System Output Bias Error	**ft**	This is the location for inputting the constant offset, which is expected from the position sensor. It is included on the environmental submenu, since it is largely due to degradation of performance as a result of weather/visibility conditions
CMU Look Down (TLC) Countermeasure System Output Noise Error	ft	This is the location for inputting the random noise expected from the position sensor. It is included on the environmental submenu, since it is largely due to degradation of performance as a result of weather/visibility conditions.
CMU Look Down (TLC) Countermeasure System Output Bias Noise	ft	This is the location for inputting the constant offset which is expected from the position sensor. It is included on the environmental submenu since it is largely due to degradation of performance as a result of weather/visibility conditions

3.4.3 Driver Type

Several default driver types are included with the simulation. These drivers were chosen to cover a normal range of typical driver reactions.

Table 3-6
Driver Type Protected Files

Road File Name	Description	Date Created
NORMAL.DVC	Normal, alert driver	8/10/95
LONGREAC.DVC	Slow reaction, 90th percentile	8/10/95
BADDRIVR.DVC	Severely degraded driver	8/l0/95
NOREACT.DVC	Driver never responds	8/10/95

The following table lists the input parameters for creating a custom driver.

Table 3-7
Parameters Input on Driver Type Submenu

Parameter Name	Units	Description
Steering Reaction Time	s	The steering reaction time is a function of the capability of the driver. The 50th percentile reaction time for steering is approximately 0.84 seconds. See Figure 3-3a for other values.
Braking Reaction Time	s	The braking reaction time is a function of the capability of the driver. The 50th percentile reaction time for braking is approximately 1.2 s. See Figure 3-3b for other values.
Driver Response Gain		The driver response gain is a function of the capability of the driver. A value of 400 is used with the Taurus.
Driver Response Crossover Frequency	rad/s	The driver response crossover frequency is a function of the capability of the driver. A typical value for this is 20 radians/second.
Lane Position Deadband	ft	The lane position deadband is a function of the driver style. It indicates the amount of lateral alignment the driver is willing to accept without initiating a corrective action. A typical value for this is 0.5 feet.
Curvature Deadband	1/ft	The curvature positron deadband is a function of the driver style. It indicates the amount of lateral alignment the driver is willing to accept without initiating a corrective action. A typical values for this is 0.00001 1/ft (1CE-5 1/ft).

The typical driver settings shown in the table produce an amplitude in lane position oscillation of about 0.6 ft, which is approximately that reported by Carson Wierwille (Reference [19781).

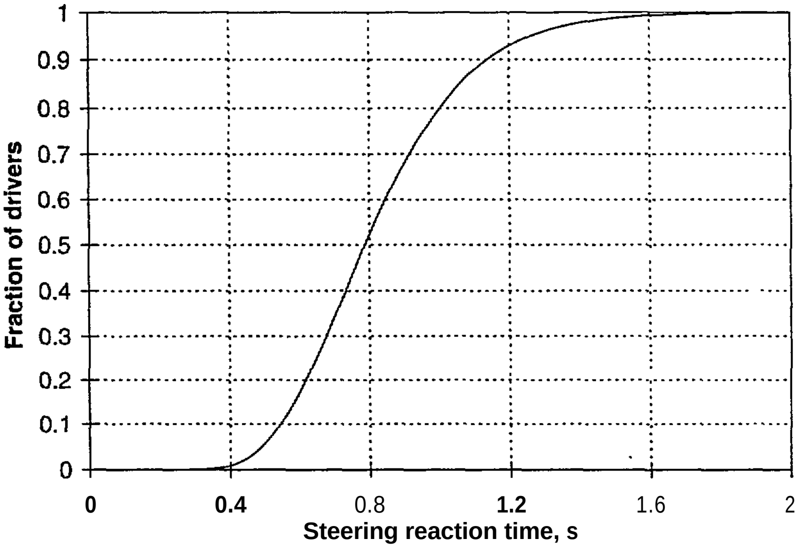


Figure 3-3a CUMULATIVE DISTRIBUTION OF STEERING REACTION TIMES. (AFTER MALATERRE AND LECHNER [1990])

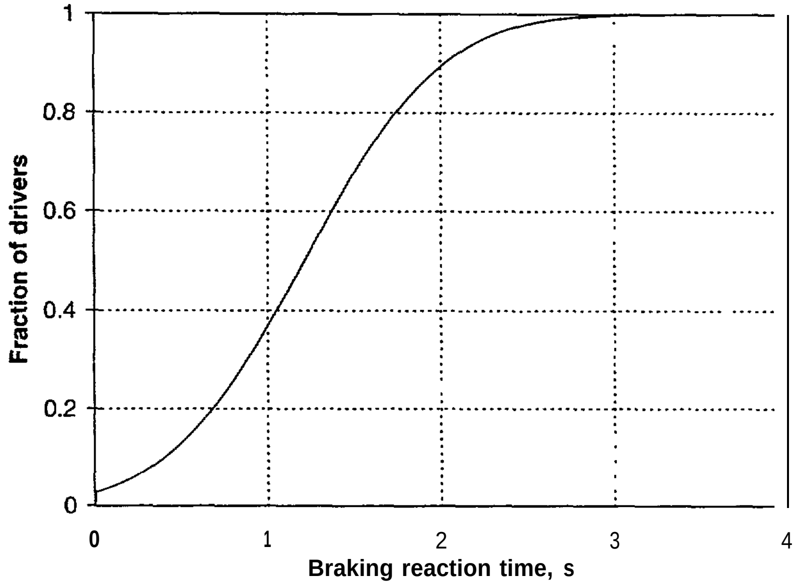


Figure 3-3b CUMULATIVE DISTRIBUTION OF BRAKING REACTION TIMES. (AFTER SIVAK [1982])

3.4.4 Simulation Scenario

Two simulation scenarios are include to demonstrate reasonable values for the parameters.

Table 3-8
Simulation Scenario Protected Files

Road File Name	Description	Date Created
NORMAL.SCN	Normal driving, 60 mph (88 fps)	8/10/95
CUTAT88.SCN	Curue cutting at 88 fps	8/l0/95
SNAP.CMS	Simulate Penna Turnpike SNAP	8/l0/95
TESTSNAP.SCN	SNAP system test scenario	8/l0/95

The following table lists and describes the simulation scenario input parameters.

Table 3-9
Parameters Input on Simulation Scenario Submenu

Parameter Name	Units	Description
Vehicle Speed	ft/s	The vehicle speed is the commanded constant speed that the driver is asked to maintain.
Vehicle Speed File Name	file name	If desired, a vehicle speed file can be generated offline and used by the simulation. This is the location in the menu system that the vehicle speed profile file is entered.
Lane Offset	ft	The lane offset is the constant offset from the centerline of the lane that the driver will try to follow. If this value is zero, the path is the centerline of the right hand lane. A positive value offsets the path to the right, a negative value offsets the path to the left.
User Defined Vehicle Path	file name	If desired, a vehicle path file can be generated offline and used by the simulation. This is the location in the menu system for inputting the name of this path file. This feature is used to make a driver cut a corner or perform lane changes

Parameter Name	Units	Description
Driver nattentive to steering	yes-no	This switch determines whether the driver will be inattentive to steering duties during the simulation. If the driver is inattentive, the steering wheel angle is fixed at the time inattention begins and is released at the end of the inattention interval.
Time On	s	The time the driver becomes inattentive.
Time Off	s	The time the driver resumes steering control.
Driver Inattentive to Speed	yes-no	This switch determines whether the driver will be inattentive to braking duties during the simulation. If the driver is inattentive, the brake is fixed at the time inattention begins and is released at the end of the inattention interval.
Time On	s	The time the driver becomes inattentive.
Time Off	s	The time the driver resumes braking control.
Driver Follows Speed Command on Warning	yes-no	This switch determines whether the driver will become attentive to the speed control tasks when a warning is issued.
Driver Brakes in Panic on Warning	yes-no	This switch determines whether the driver will brake according to the commanded speed profile or brake in panic. If braking in panic, the driver will brake at the entered g level (1 g = 32.2 ft/s^2) for the designated number of seconds
CMU Look Ahead (TTD) System Sensor Dropout	s - (from start of run)	This is the location to enable any loss of sensor data (dropouts). At the beginning of the dropout, the output of the sensor for offset to the centerline of the lane at the lookahead distance is fixed and held constant until the end of the dropout. If the dropout option is active, the beginning and ending times for the loss of sensor data are entered here.
CMU Look Down (TLC) System Sensor Dropout	s - (from start of run)	This is the location to enable any loss of sensor data (dropouts). At the beginning of the dropout, the sensor value of the offset from the centerline of the lane at the vehicle is fixed and held constant until the end of the dropout. If the dropout option is active, the beginning and ending times for the loss of sensor data are entered here.

Parameter Name	Units	Description
CMU Curve Warning System Sensor Dropout	s - (from start of run)	This is the location to enable any loss of sensor data (dropouts). At the beginning of the dropout, the sensed distance to the curve threshold is fixed and held constant until the end of the dropout. If the dropout option is active, the beginning and ending times for the loss of sensor data are entered here.

3.4.5 Countermeasure System Type

The RORSIM package contains simulations of three countermeasure systems. These are the TTD Look-ahead system, the TLC Look-down system and the Curve warning system. The variables for each of these systems is contained in the Countermeasure System Menu. Probably the most basic item to understand about the simulation of the countermeasure systems is that they are calculated in every simulation. By turning the systems ON or OFF, you are deciding whether to issue the warning to the driver. The countermeasure systems were modeled in this fashion because it is often desirable to track the values of TTD and TLC for simulations of driving without one of these systems active.

There are three countermeasure systems; however, there is only one configuration file per simulation. This file contains the parameter values for all three systems. You can enable warnings from more than one system if you want to.

Table 3-10
Countermeasure System Protected Files

Road File Name	Description	Date Created
TTD88.CMS	TTD for 88 fps parameter study	8/l0/95
TTD66.CMS	TTD for 66 fps parameter study	8/10/95
TTD44.CMS	TTD for 44 fps parameter study	8/10/95
TLC44.CMS	TLC for 44 fps parameter study	8/10/95
SNAP.CMS	Simulate Pennsylvania Turnpike - SNAP	8/10/95
ALLOFF.CMS	All systems turned OFF	8/10/95
CURVWARN.CMS	Warn of a l000-ft-radius curve	8/10/95

The following three tables contain the input variables for the three countermeasure systems.

Table 3-11
Parameters Input on CMU Look Ahead (TTD)
Countermeasure System Submenu

Parameter Name	Units	Description
Countermeasure System Processing Time	s	The amount of time which elapses between capture of the digital video image and the output of the offset from the'centerline at tbe look ahead distance.
Update Period	s	The time step at which the offset value is updated.
Sensor Output Resolution	ft	The output of the offset is rounded to the nearest resolution step before being used in the simulation.
Look Ahead Distance	**ft**	The distance in front of the vehicle that the look ahead system uses to determine the offset from centerline.
Warning Threshold	s	This is the value of the Time to Trajectory Divergence (TTD) at which a warning will be issued. If the OFF button is selected, no warning will be issued.
Intervention Threshold	s	This is the value of the Time to Trajectory Divergence (TTD) at which steering control augmentation will become active. If the OFF button is selected, no active intervention will occur.
Tolerance Distance	ft	The Time to Trajectory Divergence (TTD) value is the time until the vehicle diverges from the ideal path by a certain distance. This certain distance is input to the simulation as the Tolerance distance.
Allow Cutting	Yes-No	When in the YES position, the warning algorithm is modified to allow the driver to "cut the comer" without generating nuisance alarms.

Table 3-12
Parameters Input on CMU Look Down (TLC)
Countermeasure System Submenu

Parameter Name	Units	Description
Processing Time	s	The amount of time which elapses between capture of the digital video image and the output of the offset from the centerline.

Parameter Name	Units	Description
Update Period	s	The time step at which the offset value is updated.
Sensor Output Resolution	ft	The output of the offset is rounded to the nearest resolution step before being used in the simulation.
Edge Line Distance from Edge of Lane	ft	This allows the edge line to be offset from the actual edge of the lane. A positive offset positions the edge line away from the center of the lane and a negative offset positions the edge line closer to the center of the lane.
Believe lane Width	ft	The sensor system may not necessarily know the actual lane width. This is the lane width that the sensor system believes to be true.
Warning Threshold	s	This is the value of the Time to Line Crossing (TLC) at which a warning will be issued. If the OFF button is selected, no warning will be issued.
Intervention Threshold	s	This is the value of the Time to Line Crossing (TLC) at which steering control augmentation will become active. If the OFF button is selected, no active intervention will occur.

Table 3-13
Parameters Input on CMU Curve Warning
Countermeasure System Submenu

Parameter Name	Units	Description
Countermeasure System Processing Time	s	The amount of time which elapses between capture of the digital video image and the output of the offset from the centerline at the look ahead distance.
Update Period	s	The time step at which the offset value is updated.
GPS Sensor Output Resolution	ft	The output of the offset is rounded to the nearest resolution step before being used in the simulation.
Map System Sensor Output Resolution	ft	The output of the offset is rounded to the nearest resolution step before being used in the simulation.
Position Output Bias Error	ft	This is a constant position bias error added to the actual position.

Parameter Name	Units	Description
Position Output Noise Error	ft	This is a random positional error value which is added to the actual position.
Deceleration Warning Threshold	g's	When enabled, a warning is given to the driver a distance before the curve equal to the distance that the car will travel during the reaction deadtime and a constant deceleration level to the desired speed. This input specifies the constant deceleration level.
Deceleration Intervention Threshold	g's	When enabled, a control system will intervene to slow the car to the desired speed. The intervention occurs at a distance before the curve equal to the distance that the car will travel during the constant deceleration level to the desired speed. This input specifies the constant deceleration level.
Curve Safe Speed/ Road Parameters	Switch	This switch determines how the safe speed of the curve to be used by the curve warning system is determined. If Curve Safe Speed is chosen, the value entered is used for the safe speed. If Road Parameters is entered, the safe speed is calculated based on the believed values of friction, superelevation, and curvature.
Curve Safe Speed	ft/s	This is the speed that the countermeasure system uses as the goal speed for the curve.
Believed Coefficient of Friction	ND	This is the coefficient of friction that the curve warning system uses to estimate the safe speed in the approaching curve. It does not need to be the same value as the actual coefficient of friction.
Believed Superelevation	ft/ft	This is the superelevation (bank) that the curve warning system uses to estimate the safe speed in the approaching curve. It does not need to be the same value as the actual superelevation.
Believed Curvature	1/ft	This is the curvature that the curve warning system uses to estimate the safe speed in the approaching curve. It does not need to be the same value as the actual curvature.

3.5 Parameters Not Appearing on Menus

The menu system provides a convenient way to select the values that define a potential Run-Off-Road situation, but some of the parameters that control RORSlM's operation do not appear on the menus. The only way to adjust these parameters is to use a text editor to modify the file in which they are contained.

Most of the off-menu parameters are for VDANL itself. These parameters are contained within the files named in FILES.DUM, which VDANL reads whenever it begins a simulation. These parameters and FILES.DUM itself are discussed in the file VDANL.HLP. The vehicle parameters, including their definitions and how to measure them are, are discussed further in the rollover report accompanying VDANL [Allen, et al, 1992, particularly Appendix A].

The variables specific to RORSIM that are not in the menu system include two in the .AUX file, Yemax and StartSteer, and all of the speed controller parameters, which are usually in SPDCNTRL.SCF. The menu program reads these three lines from the .AUX file (the speed controller file name and the two variables) as whole units when it reads a new scenario definition. When the operator saves a scenario in the menu program and a .AUX file is written, the program simply writes back the most-recently read form of these lines without modification.

3.6 Running RORSIM

After selecting all of the values to set up the simulation desired, click the mouse on FILE, then on SAVE FILE. You will be prompted for a name under which to save the configuration, After saving, click on RUN! in the bar on the main menu. Because the menu program and the simulation program cannot fit in the computer's memory simultaneously, the menu program terminates and launches the simulation program. The simulation program does not return to the menu program. If you want to set up another simulation, you have to exit the simulation program and re-start RORMENU from the DOS command line. The most convenient way to proceed is to define a situation, simulate it immediately, and analyze the results before running the next simulation.

RORSIM can also be run directly from the DOS command line if the parameter files that describe the desired simulation already exist. For example, your screen might look like,

```
C:\RORSIM>VDANL,  ROR60.MAC,CLOSCALL.AUX,88
```

where **C : \RORSIM** is your current working directory, and the rest of the line (starting with VDANL) is what you type. The name of the program you are running is VDANL.EXE. There are three arguments following the program name. They are separated by commas, without spaces. The first argument is the name of the VDANL macro file. It is essentially the keystrokes you would have made, were you running VDANL manually. The second argument is the name of the User Parameters File. It was created by the menu system, and contains references to the parameter files associated with each of the pop-up menus. When you selected Save from the bar at the top of the main menu page, this is the name you gave (The .AUX extension was appended automatically). VDANL passes the name of this file to the function **USERMODULE,** which opens it and handles the input as described in Section 2.2. The third parameter is simply the starting speed of the vehicle, in feet per second. When you know ahead of time what simulations you want to run, it is probably more convenient to define them all in the menu system, save them with different names, and then run them later.

It is convenient to run several simulations together in a DOS batch file, but you need to know something about macro tiles. The planned duration of the simulation is coded into the name of the macro file. For example, ROR6O.MAC calls for a 60-second simulation. (Of course, the simulation could end before then if the vehicle runs off the road) The macro files with a five-character base name (ROR60, ROR30, etc) run the simulation and leave the display of the variables on the computer screen. This is the type of macro to use if you want to run one simulation and examine the results within VDANL. When you run a simulation from the menu system, the menu system selects the macro file with the five-character name corresponding to the duration you specify. When you are running several simulations in a row from a batch file, you need VDANL to exit so the next simulation can start. If you want, you can save time histories all of the variables of PARAM.INP in a text file. Use one of the macro files with the letters SAV in the name. A line in the batch file might look like,

```
VDANL ROR60SAV.MAC,CLOSCALL.AUX,88,CLOSCALL.VDA
```

There is a fourth parameter, which is the name of the output file. The format and possible uses of the file are described in Section 4.1. Sometimes you might want to run many simulations together (for example, in a Monte Carlo experiment) and be interested only in summary output (i.e.: what was the maximum lane exceedance of the vehicle?). The output file MAXYPOS.DAT was designed for this purpose. It is a one-line file written at the end of every simulation. It records the leftmost position of the left front tire during the entire simulation and the rightmost position of the right front tire. The name of this file is always the same, so it has to be saved after every run of VDANL. A batch file might look like,

```
VDANL ROR6OEND.MAC,CASEl.AUX,88
TYPE M?iXYPOS.DAT > RECORD.DAT
VDANL ROR60END.MAC,CASE2.AUX,88
TYPE MAXYPOS.DAT >> RECORD.DAT
VDANL ROR6OEND.MAC,CASE3.AUX,88
TYPE MAXYPOS.DAT >> RECORD.DAT
VDANL ROR60END.MAC,CASE4.AUX,88
TYPE MAXYPOS.DAT >> RECORD.DAT
```

When everything has finished, RECORD.DAT will have one line for each simulation. Note that the macro file names contain the letters END, which indicates that they terminate execution of VDANL but do not save the entire history of all variables. Of course, the macro files with SAV could be used in this manner, too, if you want the details to back up your analysis.

When you run several simulations in a row like this, remember that the auxiliary file defines the entire case, except the vehicle type. VDANL, always reads the file named FlLES.DUM to find the vehicle parameters. If you want to run simulations for two different vehicles in the same batch file, you will need to rename or copy files so that FILES.DUM always points to the parameters you want.

If you are running VDANL, and it seems to stop before it gets started, leaving an error message on the screen, look in the file OUTPUT.DAT. As RORSIM opens the parameter files, it records its progress in this file. When a file is missing or has the wrong format, the program will terminate, The last line of OUTPUT.DAT will give clues as to what went wrong.

When you have finished looking at time traces or saving data from within VDANL for one RORSIM simulation, you need to exit VDANL and start the next RORSlM simulation from RORMENU or the DOS prompt. It is possible to start another VDANL (without RORSIM) simulation without leaving VDANL, but you cannot use a macro that way. Running RORSIM without a macro file is possible, but it is not a good idea because you have to make sure that all of the right breakpoints are turned on.

While the macro files are technically optional, but realistically essential, the menu program is truly optional. For users unfamiliar with the form of the parameter files, RORMENU is a convenient means of putting all of the desired settings in the parameter files. However, if you know exactly which parameter you want to change and you are not intimidated by a text editor, you are not compelled to use the menu to change it. In fact, if you want to do five simulations, in which the only difference is the driver's steering reaction time, it is straightforward to create the parameter files with a text editor. Start with the first reaction time in the first file (e.g., DRlVER1.DVC), change the reaction time to the next value and save it with a slightly different name (e.g., DRIVER2.DVC), and so forth. Then make the respective auxiliary (.AUX) files, each one listing a different driver condition file. Then set up a batch file to run them all together.

4.0 Output Options

Several forms of output presentation are available. Each is suited to a particular need. Functions can be graphed, diagrams of the vehicle on the roadway can be made, and numerical data can be saved for different kinds of analysis. Much of RORSIM's output capabilities are simply the output features of VDANL. They are the primary subject of Section 4.1. A separate post-processor is available for generating the diagrams of the vehicle's position. It is described in Section 4.2.

4.1 Graphical and Numeric

When a simulation is in progress, the values of four variables are drawn across the computer screen, like a very slow oscilloscope. One way to view the results of a simulation is to look at these and similar plots of other key variables. Let us begin by talking about this plot. If you press the 3 key, the third graph will be expanded to occupy the entire screen. Press Return again, and the original graph will reappear. To put a title on the plot, press the T key. A little entry box will appear where you can type the title. That title will be at the top of all graphs you make until you change the title again. That is handy if you want to print several graphs of results from a single simulation because they all have the same title to tie them together. If you want to take a closer look at an event that happened between, say, 10 and 20 seconds into the simulation, press the S key to bring up an entry box for changing the scaling. Follow the instructions in the box to change the horizontal or vertical scaling, then press P and Return to plot the graphs with the new scaling. If the right kind printer is connected to your computer, (Epson dot matrix, IBM dot matrix, and HP laser printer) you can print the graph on paper. Press the P key to bring up the print menu. The first time you do this, it may be necessary to change the configuration of the printer (Type C). After that, just type L for Landscape printing. (If you type P for Portrait, VDANL will want to put two graphs on the same page. If you use landscape, you will not have to worry about whether another plot is needed to flush the printer.) This is just enough to get started on dressing up plots; more details are inVDNL.HLP under the heading "DISPLAY GRAPHICS".

To plot other variables from the simulation, press the Enter key to remove the existing plot. The VDANL main menu will appear. Type T for Transient response plots. You may now select one to four of the thirty variables listed. After you press the P key or select the fourth variable, the computer will draw the plot on the screen. It can now be scaled or printed as discussed in the preceeding paragraph. When you are finished with this plot, press the Enter key to return to VDANL's transient response plots menu. You may make another plot or press R to return to the VDANL main menu.

All of the plotting so far has been of variables as a function of time. One variable can be plotted against another by selecting XY plots from the VDANL main menu.

To export VDANL results to another program, including the RORSIM Post Processor (RORPLOT), select Save data from the VDANL main menu. One or more of the 30 variables can be selected here as on the transient response plot menu, After selecting all of the desired variables, press the S key, then provide a file name and starting and ending times as prompted. By convention, the file names have the extension .VDA. If the definition for a case was CLOSCALL.AUX, the output file name might be CLOSCALL.VDA. The save parameters menu, unlike the transient response menu, has no upper limit on the number of variables that can be saved. Also, if you select S for save before specifying any variables, all thirty will be saved. This is handy for saving the results of a simulation and recalling them to VDANL later. (To bring them back into VDANL, use L for Load from the VDANL main menu.)

The variables are saved to ASCII text files. They can be viewed with any text editor, and they may be imported to spreadsheets for detailed analyses. The data are in column format with the leftmost column representing time. The other variables follow in columns in the order they were selected. The variables in the file are listed at the top of the file, after a line with the date stamp.

VDANL sets aside a fixed amount of memory for saving results during the simulation. Therefore, the time interval of the output file depends on the number of variables being tracked (which is usually 30) and the duration of the simulation. For example, if 30 variables are saved and the simulation was expected to last 60 seconds, the VDANL sets the sampling interval to 0.015 s. If you are planning to plot the results of one simulation alongside the results of another simulation, we recommend saving the same number of variables for the same run duration so that the sampling interval will be the same.

The thirty variables available for plotting represents only about one tenth of the total internal variables in VDANL. Any combination of thirty or fewer variables (selected before the start of a simulation!) can be listed or plotted or saved. To change the list, edit PARAM.INP, following the instructions in VDANL.HLP. The complete list of internal VDANL variables is in VDANL.HLP; the variables that are explicitly used by the RORSIM enhancements to VDANL are listed in VDANLVAR.INC, along with a few variables defined specifically for RORSIM. If you want to follow the evolution of an event at a level of detail not possible through the available variables, you can modify the program as outlined in Section 5.2. Any quantity, including an internal calculation variable, can be written to OUTPUT.DAT at every time step. Again, details on handling the results within VDANL are provided in the VDANL documentation.

The final form of numerical output is the brief summary in the file MAXYPOS.DAT. This one-line file has three fields: the name of the simulation, the farthest left that the left front tire ever traveled during the simulation, and the farthest right that the right front tire traveled during the simulation. It is written every time RORSIM runs. You do not need to specifically ask for it, but you do need to rename it before you run RORSIM again if you want to keep it.

4.2 Post-Processing

To help in visualizing, the results of RORSIM, the trajectory of a vehicle can be plotted along with lane lines over a critical interval, as shown in Figure 4-l. The position of the vehicle is shown at several points in time with the plots resembling a strobe photograph form a bird's eye view. These plots are generated by RORPLOT, a program separate from VDANL. The procedure for producing the plots comprises three steps: (1) save the required variables from VDANL, (2) run RORPLOT, and (3) run AutoCAD to view or print the plots. Details follow.

Before the timeline plots can be produced, the proper variables must be exported from VDANL. First run the simulation. When the simulation finished, press Return, and the VDANL main menu will replace the four traces on the computer screen. Press S for Save data to file. Then, using the mouse, the arrow keys, or by entering numbers, select these variables in this order:

Trip
Yi
PsiOff
LWidth
AheadRPsi
AheadRX
AheadRY.

(Time is automatically saved.)

After selecting the variables, press the S key. Select a name for a temporary output file and type it when requested. Usually, merely pressing the Enter key when asked for the starting and stopping times is adequate. VDANL will tell you while it is writing the file. The save data menu will reappear. Press R to return to the VDANL main menu. Then press E to exit the program, and answer Yes when asked if you are sure.

From the DOS prompt, run the program RORPLOT. This will convert the data to a form meaningful to AutoCAD. RORPLOT will prompt you for several selections. If you have created a RORSIM output file named CRASH.VDA, a typical RORPLOT session would look like the sequence below. The computer's prompts, which are in all capitals, **LOOK LIKE THIS** below, and the words you would type **are in this style.** (Do not forget to type a carriage return after each command.) Commentary is in italics.

TYPE THE NAME OF THE VDANL OUTPUT FILE.
crash.vda — *You have to type the* .VDA.

TYPE THE AUTOLISP FILENAME.
crash.lsp — *iThe .LSP extension is not required, but it is a good idea.*

```
TYPE THE HISTORY DATA FILENAME.
crash.dat
```

Again, the .DAT extension is a good idea.

```
TOTAL SIMULATION TIME = 28.4000 s.
```

This reminder helps you with the next selections.

```
TYPE THE TIME OF THE FIRST CAR, s
10.0
```

The first car in the plot will be at the vehicle's position at this time into the simulation.

```
HOW MANY CARS TO PLOT? ( 1 TO 10 )
```

This is the number of cars that will be in the plot.

```
TYPE THE TIME INTERVAL BETWEEN CARS, s
1.0
```

This will be rounded to a multiple of the print interval in the .VDA file.

```
TYPE THE CAR ORIENTATION AMPLIFICATION FACTOR.
1.0
```

Sometimes rotations are hard to see.
A value of 1.0 means no amplification.
A value of 4.0 is appropriate for some cases.

A summary of your entries will be printed.

```
DO YOU WISH TO CONTINUE?   (Y=YES N=NO)
```

Then it will ask if it is ok.

If all is well, there will be no further messages from RORPLOT.

Now transfer the .LSP file to the AutoCAD directory. Type

```
ACAD
```

at the DOS prompt to `start` AutoCAD. Go past the introductory sections begin a new plot and add name to your file, At the Autocad command prompt, type

```
 (LOAD "CRASH.LSP")
```

You must type the parantheses and the double quotes.

The file is loaded and the line C:\INSCAR `appears` at the bottom of the screen. To create the plot, type

```
INSCAR
```

at the prompt. The commands being executed will scroll by.

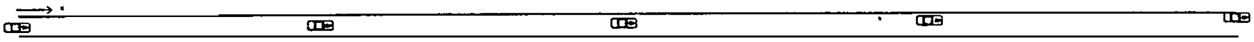
When the cars and road appear, they may be quite small. Use the mouse to move the cursor up to the top of the screen so the menu bar appears. Click the mouse on Display. A menu appears. Select Zoom Dynamic with the mouse. A large rectangle appears with an X at the center. The road and cars will still be displayed. By clicking the left button of the mouse, you can switch between adjusting the position and size of the rectangle. When the X disappears and an arrow appears at the right side of the rectangle, you can adjust the size of the rectangle. Move and resize the rectangle to frame the cars as you wish. The aim is to move the rectangle to frame the cars as closely and as much to the center of the rectangle as possible. Click the **right** mouse button to enlarge the drawing.

To print the drawing, click the mouse on File (in the menu bar at the top) and choose Print. A prompt will appear at the bottom of the screen; press Enter. The specifications for the plot will appear. They should be the following:

Plot will NOT be written to a selected file
Sizes are in inches
Plot origin is at (0.0, 0.0)
Plotting area is 7.99 wide by 10.29 high (max size)
2D plots are rotated 90 degrees clockwise
Hidden lines will NOT be removed
Plot will be scaled to fit available area.

When the specifications are set, enter file name for plot. The plot will be processed and printed.

After printing, you will be back in the drawing editor. Click on File in the menu at the top. To save the drawing, click on Save. Enter a file name at the command line. After saving the file, again select File from the menu at the top and click on End. This exits AutoCAD.



5.0 Customizing

RORSIM establishes a basic capability in simulating the performance of countermeasure systems in potential Run-Off-Road events. More features have been built in than have been used so far during Phase I in evaluating proposed countermeasure systems. Many more questions can be answered with the programs as they stand. To be a completely useful tool for evaluating any countermeasure system that may be proposed in the future, the program provides the flexibility of modeling a system that is completely different from the three currently installed.

Adding a new countermeasure system, on the first level, requires only that the code describing the system's operation be written in a new source file and that minor modifications be made at places where the new subroutines are declared and called. Because the input parameters are read from text files, evaluation of the new countermeasure can proceed without modifying the RORSIM menu.

If you skipped Section 2.0 when you were learning RORSIM, now would be a good time to go back and read it.

5.1 The Menu System

The menu system was written in the C programming language, using Borland C++ version 4.0. The buttons and menus use features available in Lab Windows version 2.3. Minor modifications can be effected simply by changing the C code. Adding new entry fields, and other significant changes, require that the interface be modified first within Lab Windows, then code changes to accommodate the new field must be made in the C program.

All of the menus follow a similar format. The procedure for adding a new entry field or button is first to place the field on the menu. This is done with the program UIEDIT, which is part of Lab Windows. [National Instruments, 1994]. Then, following the format used for the existing entry fields, write the code in the appropriate C function.

When adding new entry fields, be sure the format of the parameter files written by the menu is compatible with that expected by the input functions in RORSIM. The variable name on the left side of the = sign in the parameter files is only a mnemonic for humans; it is ignored by the computer program. The variables must be in the order given. Rearranging the variable order will cause errors in execution.

To make any changes in the menu program, you will need to license Borland C++ and Lab Windows, as they are not provided with RORSIM.

5.2 The Simulation Program

The behavior of the countermeasure systems can be modified by changing the RORSIM code provided with this package. VDANL itself and the RORSlM modifications are written in Microsoft

Professional Basic version 7.1. The RORSIM code is connected to VDANL proper using the VDANL Open Module described in Section 2.1; the code of VDANL itself is not available for modification.

The procedure for making a modification is first to edit one or more source files (and the include files if necessary). Secondly, compile the files with a command like,

```
      C:\RORSIM\SOURCE>BC  UMROR.BAS  /OT/FPI/G2
```

Then link the program with the command,

```
      C:\RORSIM\SOURCE\LINK  @ROR.LNK
```

The file ROR.LNK supplied on the diskette has the directories as they were on the computer where RORSIM was developed. You need to change ROR.LNK to reflect you own directory structure.

The file ROR . LNK supplied on the diskette has the directories as they were on the computer where RORSIM was developed. You need to change ROR.LNK to reflect you own directory structure.
Then copy the executable file RORSIM.EXE to the directory where programs are being run. Details on using the VDANL open module are provided in the VDANL, open module documentaiton [Rosenthal, 1994].

The low-level details of each variable's function and the calculation methodologies are explained in the code comments. They are not repeated in this manual, so be sure to read the comments carefully before attempting any significant code change.

The simplest modifications that might be made to RORSIM are to write to OUTPUT.DAT so a variable can be followed as an event evolves. The file remains open throughout the program, so any PRINT or WRITE statement to #23 1 will put data in the file. Many of the statements that were used in developing the program remain in the code as comments. Following their example would be helpful.

The most important enhancement that might be made to RORSIM would be to add a new countermeasure system. As noted in Section 2.5, the code for a countermeasure system is contained in at least two files, one for the sensor and one for the decisionmaking algorithm. To add a new countermeasure system, one or more source files must be written to model its sensor or sensors, and another source file, to model its decisionmaking algorithm. Then the new system has to be "wired in" by putting declarations for its functions in RORCM.INC, constants for its new variables in VDANLVAR.INC, and the code for calling it in UMROR.BAS. Probably the best way to write a new sensor or decisionmaking source file is to start with one of the existing files and modify it for the new system. This will help you to follow conventions for reading parameters at initialization, degrading the sensor output, and so forth. Calls need to be added in UMROR.BAS in two places. The initialization function has to be called at the initialization breakpoint. The model function is called in the curvature error breakpoint. (The curvature error breakpoint is the last breakpoint in the closed-loop driver model, so all countermeasure systems go there regardless of whether they actually involve curvature error.) Any code to handle active intervention specified by the countermeasure system would also be in the curvature error breakpoint.

The settings (sensor noise level, warning threshold, etc.) for the new countermeasure system can be listed in the same parameter files as the settings for the existing systems. The settings for the new system should be listed in the files after a line with square brackets identifying the new system, [new system] . Actually, the name of the new system can be anything unique, though it should be descriptive, as long as the characters in the files exactly match the characters in the code where the files are read. The match is not case sensitive ([new sys tern] matches [New Sys tern] , but not [newsystem]). The presence of parameters for new systems in the parameters files will not disturb the existing countermeasure systems. The menu program, however, will malfunction if it tries to read a file with extra parameters. It can continue to read and change files that are not affected by the new parameters. For example, suppose you write a new decisionmaking algorithm to handle the output of the GPS sensor. You would put the parameters for your new system in a countermeasures file with the extension .CMS. Any time you make changes to the parameters in this file, you need to edit it manually with a text editor, whether you are changing the parameters for your new system or one of the original systems. You would also use a text editor to put the name of the new countermeasures file in your auxiliary (.AUX) file, on the line **COUNTERMEASURES = NEWSYSTM . CMS.** If you wanted to change, say, just the roadway for a simulation, you could use RORMENU, as long as you never went to the menu page for the countermeasure system. Of course, if you wanted to fully integrate your new system with RORSIM, you could write the menu page for your new system.

To make the modeling realistic, the following conventions should be followed for the sensor and decision-making functions. The sensor function can access any variable in the VARYY array needed to simulate the response of a sensor. The decision-making function, on the other hand, should use only measurements coming from the sensor, or those that could be programmed directly into a system, such as the wheelbase of the vehicle. For convenience, however, other variables may be used directly, as was done with the perfect yaw rate sensor in TTD. Be sure to provide a way to set the threshold levels for the new system so that it can be disabled when not needed.

References

Allen, R.W., et al, (1975), "Alcohol Effects on Driving Behavior and Performance in a Car Simulator," ***IEEE Transactions on Systems, Man, and Cybernetics,*** Vol. SMC5.

Allen, R.W., et al, (1992), "Vehicle Dynamic Stability and Rollover," ***Final Report on NHTSA,*** Contract DTNH22-88-C-07384.

AutoCAD Manuals, Autodesk, Inc.

Carnegie Mellon University and Calspan Corporation, (1994), October 28, "Run-Off-Road Collision Avoidance Using IVHS Countermeasures," ***Task I Final Interim Report,*** Volume I: Technical Findings, NHTSA Contract No. DTNH22-93-C-0723.

Carson, J. M., and Wierwille, W.W., (1978), "Development of a Strategy Model of the Driver in Lane Keeping," Vehicle System Dynamics, Volume 7, Number ***4, International Journal* of *Vehicle Mechanics ad Mobility.***

Dijks, A., (1974), May, 'Wet Skid Resistance of ***Car*** and Truck Tires", ***Tire Science and Technology,*** TSTCA, Vol. 2, Nl. 2, pp. 102-l 16.

Malaterre, G., & Lechner, D., (1990) "Emergency maneuver at junctions using a driving simulator." Transportation and Traffic Theory: Proceedings of the 1 lth International Symposium on Transportation & Traffic Theory, 213-232.

National Instruments Corporation, (1994), Lab Windows User Interface Library Reference Manual, Version 2.3.

Pape, D., Narendran, V., Hadden, J., Everson, J, Koenig, M., and Pomerleau, D., (1995), "Run-off-road collision avoidance using IVHS countermeasures: Task 3--Volume I." (Contract No. DTNH22-93-C-07023). Pittsburgh, PA: Carnegie-Mellon University Robotics Institute.

Rosenthal, Theodore J., (1994), ***VDANL User-Defined Modules,*** VDANL companion manual from Systems Technology, Inc.

Sivak, M., Olson, P.L., & Farmer, K.M., (1982) "Radar-measured reaction times of unalerted drivers to brake signals." Perceptual & Motor Skills, 55:594.

Systems Technology, Inc., (1989), ***VDANL User's Guide,*** VDANL companion manual from Systems Technology, Inc.

Wood, N.E., (1994), January "Shoulder rumble strips: a method to alert 'drifting' drivers." Transportation Research Board 73rd Annual Meeting. Paper No. 94-03 12.

APPENDICES

Appendix A. Quick Start Instructions

This appendix will help you get RORSIM installed, and it will show you the basic skills for using it.

A-l Installing RORSIM

To install RORSIM, create a directory on your hard drive. You can name the directory RORSIM if you want. You'll need about 4 Megabytes of free disk space for the installation.' Copy all files on the distribution diskette to your directory. Type INSTALL at the DOS prompt when you are in your directory. The sequence might look like this, where you type everything that's underlined:

```
C:\>  mkdir  rorsim

C:\>  cd rorsim

C:\RORSIM> copy a:*.*
A:FILELIST.DOC
A:INSTALL.BAT
A:CASES.EXE
A:PROGS.EXE
A:RORPLOTZ.EXE
A:SOURCE.EXE
A:TAURUS95.EXE
        7 file(s) copied

C:\RORSIM> install
```

If you're a little tight on disk space, you can delete the files in the SOURCE and RORPLOT subdirectories. The files in SOURCE are necessary only when you make changes to the simulation code. The file in RORPLOT is necessary only when you make plots with the post-processor.

A-2 A Simple Simulation To Make Sure It's Installed

If RORSIM is installed and your VDANL hardware key is connected to your parallel port, go to the RUN subdirectory. Type RORMENU at the DOS prompt. The menu program should start and display the main menu screen. Since you have just installed RORSIM, the case loaded in the menu is for a normal driver at 88 fps on 1000-ft-radius curve to the right. (Every time after this when you start RORMENU, it will have the case exactly as you had it the most recent time you left RORMENU). To confirm everything is working, use your mouse to click on RUN! at the top of the screen. The screen should go blank for a few seconds, then several characters will be displayed briefly. Next, four graphs will appear on the screen. You are now watching the evolution of TTD, TLC, Yi, and PsiOff as the simulation proceeds. (TTD and TLC are confidence measures of the countersystem. Yi and PsiOff are the position and yaw angle in local, road-based coordinates. See

Appendix E.) After the simulation has finished, press Enter, E, and then Y to leave the program and return to the DOS prompt. If the program did not seem to work or gave you an error message, see Section 3.6 for help.

A-3 How To Change Parameter: Setting Up A Crash Simulation

Now we can go through the process again, adding a few details. Start RORMENU again. This time, click on FILE in the upper left comer. Select LOAD. You will now see a list of the pre-defined cases. Battelle has included a few standard cases with the RORSIM delivery package. Any created by you in the future will be listed here, too. Select PLAIN.AUX. This case will simulate a vehicle at 88 fps on a straight, crowned road. After the file loads, the description under the blue Roadway button will tell you that the road is straight. Let us make a Run-Off-Road crash this time. Click on the blue button labeled, "Simulation Scenario." This will call up a menu page that describes the input scenario. To cause the crash, we will make the driver divert his attention from steering at a critical moment. In the right central part of the screen there is a group of entry fields under the heading "Driver Response." The top item in this group is, "Driver inattentive to steering?". Use your mouse to drag the blue switch up to Yes. The numerals in the two fields below this will now indicate that the driver will be inattentive from 0 s to 99 s, which is the entire simulation. Click in the box with the 0 and change the value to 10.4 s. The driver will steer normally for the first 10.4 s of the simulation. This gives the simulation the opportunity to initialize and settle out. At 10.4 s, the driver will stop making steering adjustments. (Maybe the driver wanted to select a different radio station.) The driver will remain inattentive to steering for the rest of the simulation, which will not continue for the full 30 s! This is the only change in the menu to make for now, so click on the pink Accept button in the upper right. The program noticed that you made a change, so it asks if you want to save this under another name or cancel the changes. The scenario where the driver is always attentive is one of the cases defined by Battelle, so it is protected, which is why you do not get the choice to overwrite the scenario you started with.. Click on Save. Enter a new file name. "OFFAT10" is a good choice. (The program will automatically append the .SCN extension to the file name, so do not type one yourself.) A good description for this tile would be "Driver inattentive at 10.4 s," but you are free to type anything (within 30 characters) that will help you remember what the file does.

A-4 Saving Your Changes And Running The Simulation

After you complete the change to the scenario, the Main Menu will reappear. Note that the scenario description you entered is now displayed under the Simulation Scenario button. Click on Run! to run your simulation. Before running the simulation, the program will ask if you want to save the configuration. If you save this configuration, you create a new file with the .AUX extension that completely describes the case (e.g., roadway, scenario, driver, etc.). Even if you do not save it now, the case will automatically be re-loaded the next time you start RORMENU. We recommend that if you ever might want this scenario again, now is the time to save it. Saving the configuration file in this way is equivalent to selecting File then Save in the far upper left of the main menu. Whether you save this file now is up to you. Either way, the simulation will start right after you leave the menu program. The same four data traces will be displayed as the simulation continues. You will know when the vehicle crashes because Yi, its distance from the lane center, will get larger once the driver becomes inattentive. The program automatically stops the simulation when the vehicle is 8 ft off the road,

A-5 Viewing The Results

Once the simulation has terminated, you have several options for viewing the data. Press Enter to return to the VDANL main menu. Type T for "Transient response plots" to plot some more variables. Displayed is a list of the 30 variables available for plotting. Select dsw, psid, TrueYRF, and F. To do so, type "14 Enter 16 Enter 19 Enter 24 Enter". After you type the fourth number and press Enter, the four plots will appear. The second line needs to be resealed, so type "S 2,0.01 Enter P Enter." The steering wheel angle remains fixed at its position after 10.4 s because that is when the driver stopped paying attention to steering. The right front tire gets on the shoulder when TrueYRF exceeds 6 ft, half the lane width. At this time, psid increases slightly due to the higher drag of the shoulder on the tire. This is a brief example of the plotting capabilities of VDANL, there are significantly more options available. See the VDANL documentation, especially the VDANL.HLP, the online help file, under the section, "DISPLAY GRAPHICS."

A-6 Advanced Features: Simulating A Countermeasure System

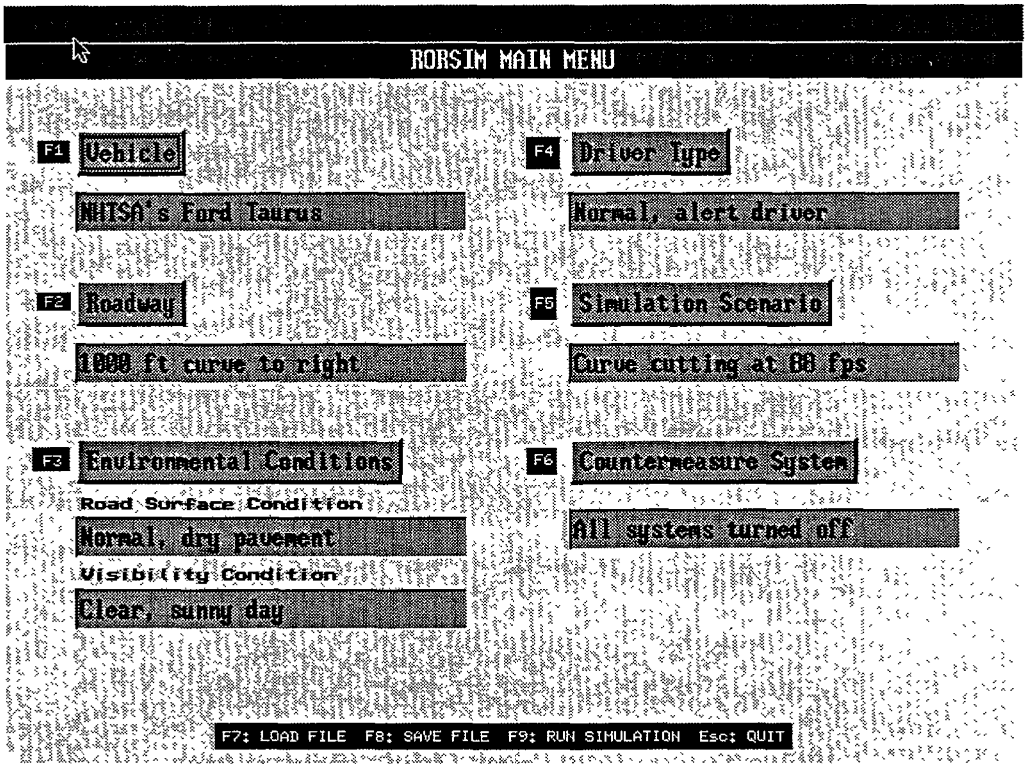
The two cases run have simulated a normal driving situation and a crash. We will add a countermeasure system to the simulation. From the DOS prompt, start RORMENU again. It will return to the state where you left it, which should be the driver who stops steering part way through a curve at 88 fps. Click on the Countermeasure System button. The area at the top of the screen lists the name and description of all of the countermeasure systems that have been defined. Click once on the file named TTD88.CMS, then click the pink Select button at the right. That selects the new file. Clicking the blue button labeled, "CMU Look Ahead (TTD) System," will display the setup of the TTD system for this case: click Retum to get back to the countermeasure system menu. In the countermeasure system menu, click Accept. Since this is not a new file, you will not be prompted for a new file name. On the main menu, click RUN! to start the simulation. At about 15 1/2 s into the simulation, when the vehicle is about to run off the road, the TTD system will "sound" a warning to the driver. The driver's attention will be restored to the task at hand, and steering corrections will again be made. The vehicle remains in its lane. Display the variables Warn1 and CurvER. The value of Warn1 will go from 0 to 1 when the TTD system issues the warning to the driver. CurvER is the curvature error in the closed-loop steering model. It is forced to zero when the driver is inattentive, and it returns to a finite value when the driver resumes steering. Note that there is a slight delay between the warning and the resumption of steering corrections because of the human's reaction time.

A-7 You’re Ready!

Nearly all of RORSIM’s capabilities are variations on what you have seen in this quick start. Detailed descriptions of the variables in the menus are in the tables of Section 3 of this manual. Explanations of the plotting variables are in VDANL.HLP, under the heading “PLOTTING VARIABLES FILE”. If you want to run several simulations together, see Section 3.6. More information on the models, including the countermeasure systems, is in Section 5 of Volume I of this report. Source listings and standard cases are in the accompanying 3-ring binder. The simulation program itself is actually a version of VDANL with Battelle’s enhancements. For help getting around within the program, please refer to the VDANL documentation.

One final installation tip: calculations in BASIC are considerably faster when EMM386 is not installed. If you’re planning to do a lot of simulating and you know your way around inside CONFIG.SYS, boot your machine without EMM386.

Appendix B. Menu Screens



Vehicle: NHTSA's Ford Taurus

Driver Type: Normal, alert driver

Roadway: 1000 ft curve to right

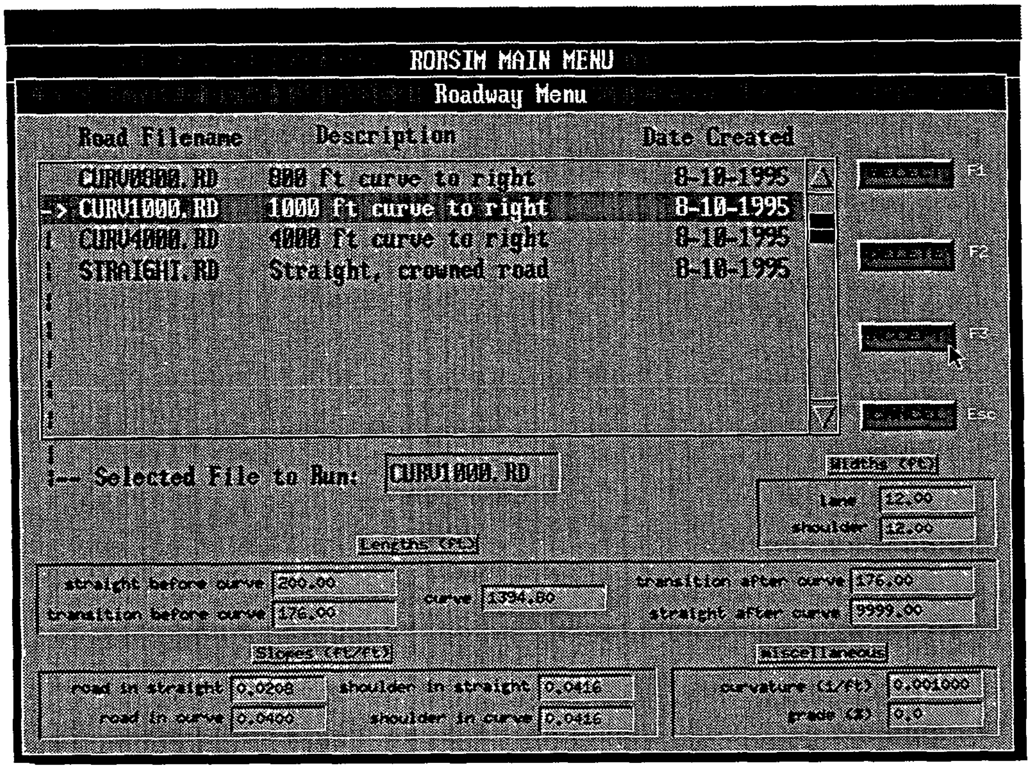
Simulation Scenario: Curve cutting at 88 fps

Environmental Conditions

Road Surface Condition: Normal, dry pavement

Visibility Condition: Clear, sunny day

Countermeasure System: All systems turned off



Roadway Menu

Road Filename	Description	Date Created
CURV0800.RD	800 ft curve to right	8-10-1995
-> CURV1000.RD	1000 ft curve to right	8-10-1995
CURV4000.RD	4000 ft curve to right	8-10-1995
STRAIGHT.RD	Straight, crowned road	8-10-1995

Selected File to Run: CURV1000.RD

Widths (ft)

lane 12.00
shoulder 12.00

Lengths (ft)

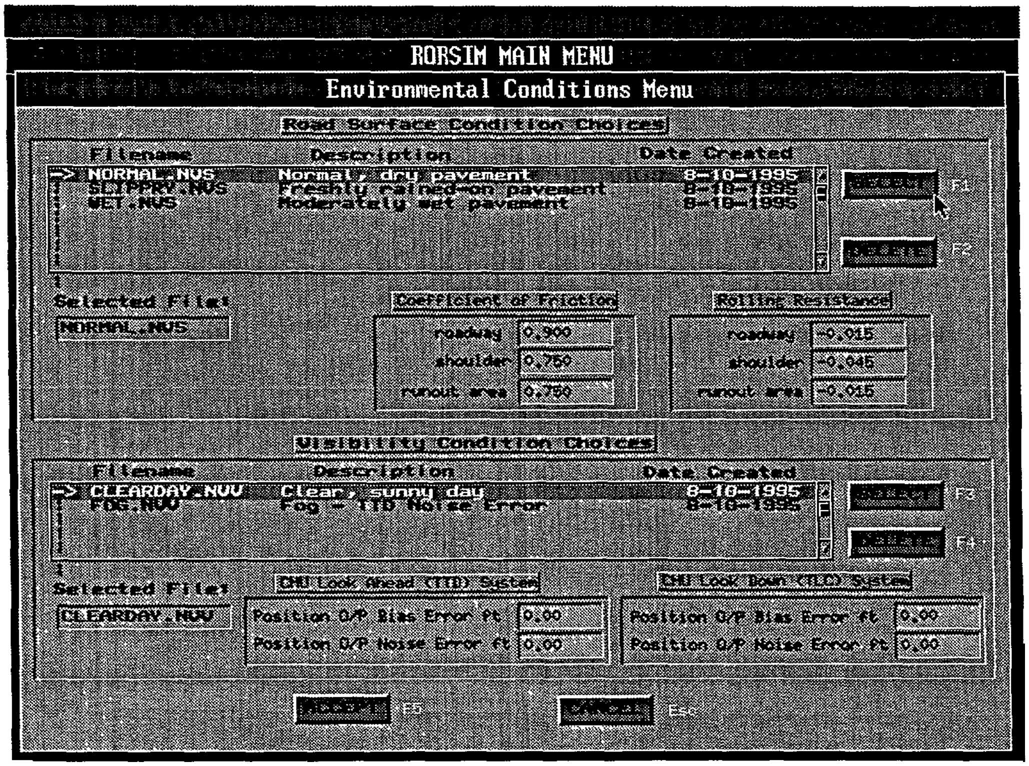
straight before curve 200.00
transition before curve 176.00
curve 1394.80
transition after curve 176.00
straight after curve 9999.00

Slopes (ft/ft)

road in straight 0.0208
road in curve 0.0400
shoulder in straight 0.0416
shoulder in curve 0.0416

Miscellaneous

curvature (1/ft) 0.001000
grade (%) 0.0



Environmental Conditions Menu

Road Surface Condition Choices

Filename	Description	Date Created
-> NORMAL.NVS	Normal, dry pavement	8-10-1995
SLIPPRY.NVS	Freshly rained-on pavement	8-10-1995
WET.NVS	Moderately wet pavement	8-10-1995

Selected File: NORMAL.NVS

Coefficient of Friction

roadway 0.900
shoulder 0.750
runout area 0.750

Rolling Resistance

roadway -0.015
shoulder -0.045
runout area -0.015

Visibility Condition Choices

Filename	Description	Date Created
-> CLEARDAY.NVV	Clear, sunny day	8-10-1995
FOG.NVV	Fog - TTD Noise Error	8-10-1995

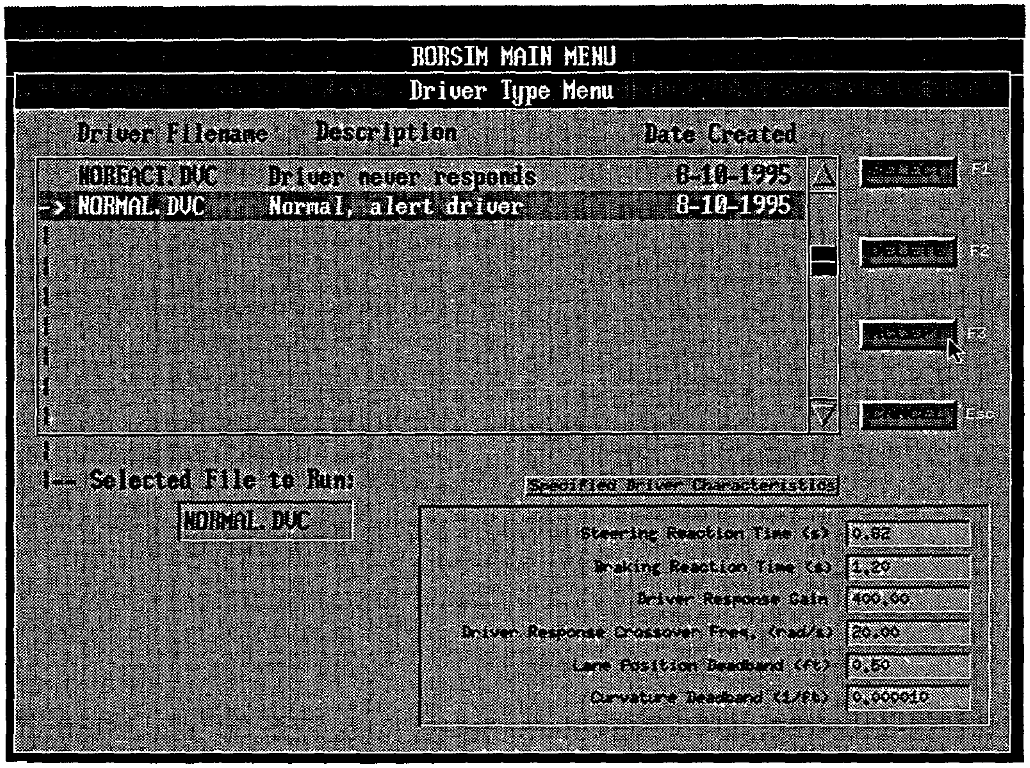
Selected File: CLEARDAY.NVV

CMU Look Ahead (TTD) System

Position O/P Bias Error ft 0.00
Position O/P Noise Error ft 0.00

CMU Look Down (TLC) System

Position O/P Bias Error ft 0.00
Position O/P Noise Error ft 0.00



RORSIM MAIN MENU

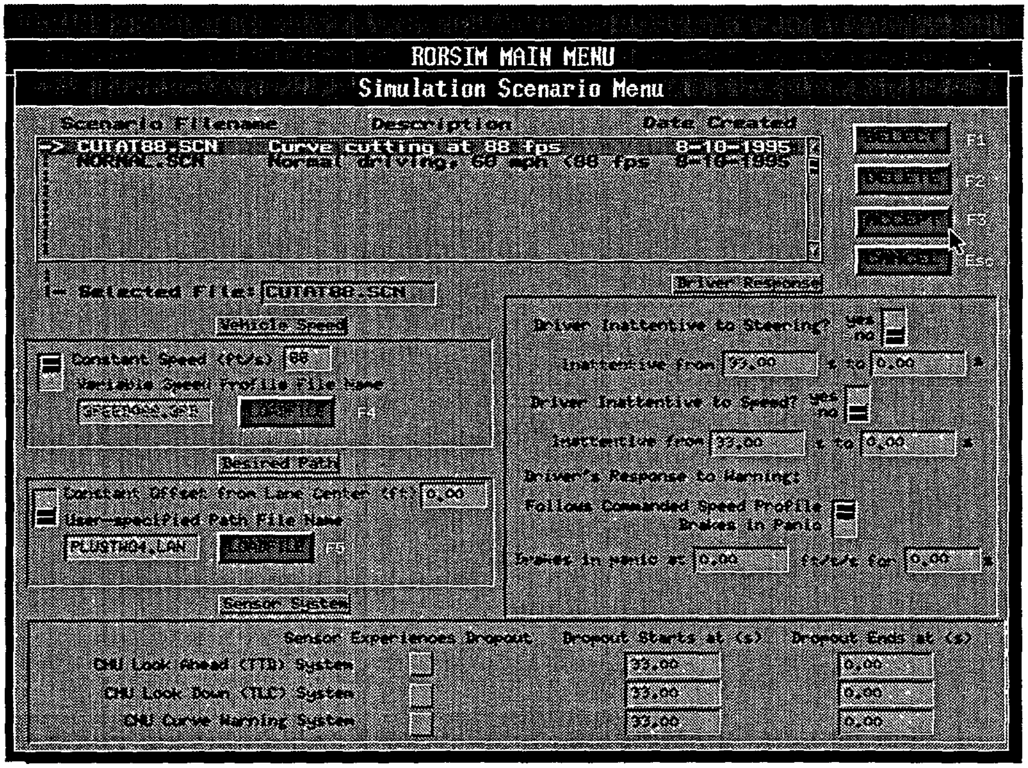
Driver Type Menu

Driver Filename	Description	Date Created
NOREACT.DVC	Driver never responds	8-10-1995
-> NORMAL.DVC	Normal, alert driver	8-10-1995

|-- Selected File to Run: NORMAL.DVC

Specified Driver Characteristics

Steering Reaction Time (s)	0.82
Braking Reaction Time (s)	1.20
Driver Response Gain	400.00
Driver Response Crossover Freq. (rad/s)	20.00
Lane Position Deadband (ft)	0.50
Curvature Deadband (1/ft)	0.000010



RORSIM MAIN MENU

Simulation Scenario Menu

Scenario Filename	Description	Date Created
-> CUTAT88.SCN	Curve cutting at 88 fps	8-10-1995
NORMAL.SCN	Normal driving, 60 mph (88 fps	8-10-1995

|-- Selected File: CUTAT88.SCN

Vehicle Speed

● Constant Speed (ft/s) 88
○ Variable Speed Profile File Name SPEED[illegible].SPF

Desired Path

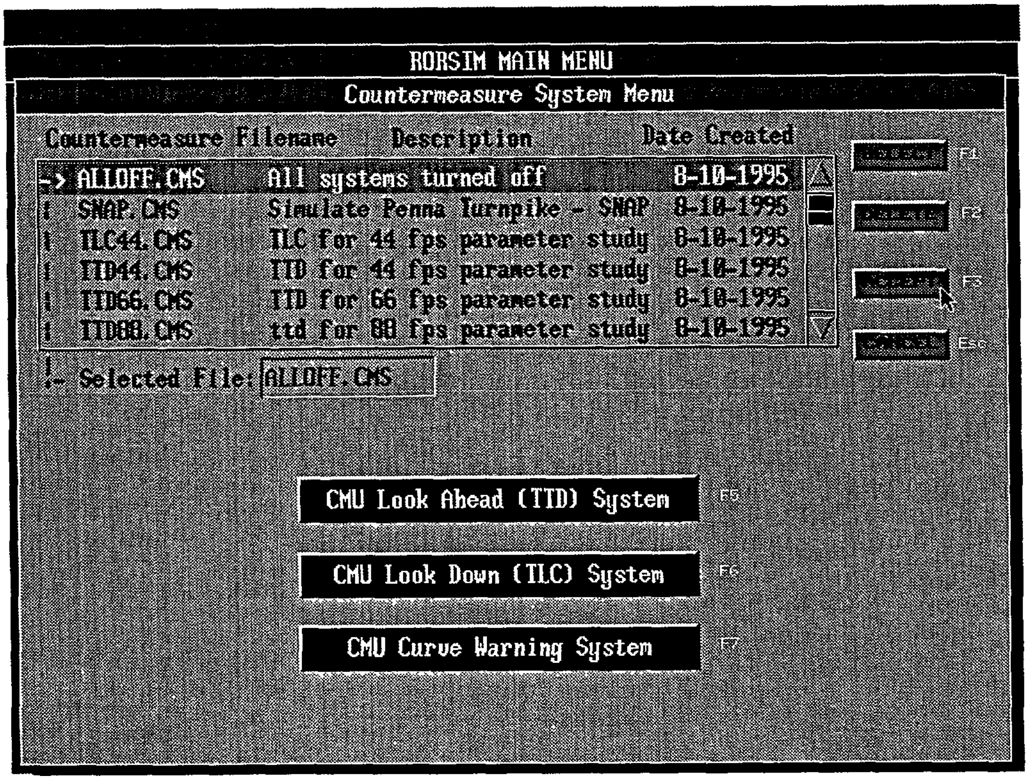
○ Constant Offset from Lane Center (ft) 0.00
● User-specified Path File Name PLUSTWO4.LAN

Driver Response

Driver Inattentive to Steering? yes ○ no ●
Inattentive from 33.00 s to 0.00 s
Driver Inattentive to Speed? yes ○ no ●
Inattentive from 33.00 s to 0.00 s
Driver's Response to Warning:
Follows Commanded Speed Profile ●
Brakes in Panic ○
Brakes in panic at 0.00 ft/s/s for 0.00 s

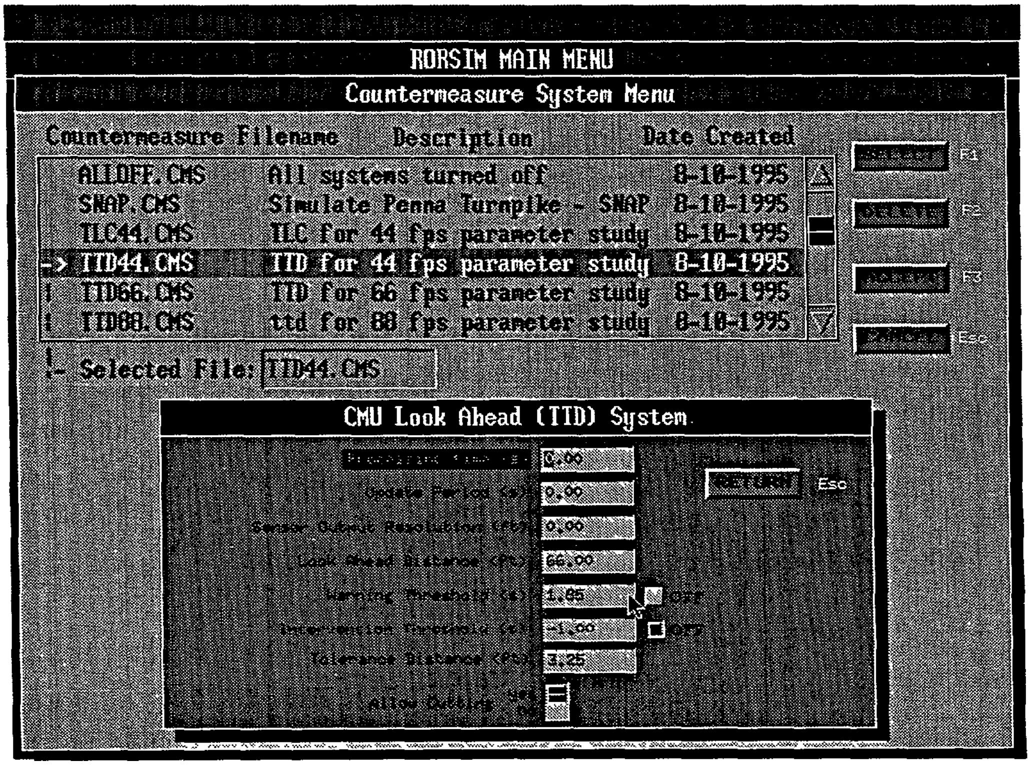
Sensor System

	Sensor Experiences Dropout	Dropout Starts at (s)	Dropout Ends at (s)
CHU Look Ahead (TTB) System	□	33.00	0.00
CHU Look Down (TLC) System	□	33.00	0.00
CHU Curve Warning System	□	33.00	0.00


RORSIM MAIN MENU
Countermeasure System Menu
SELECT F1
DELETE F2
ACCEPT F3
CANCEL Esc
CMU Look Ahead (TTD) System F5
CMU Look Down (TLC) System F6
CMU Curve Warning System F7

Countermeasure Filename	Description	Date Created
-> ALLOFF.CMS	All systems turned off	8-10-1995
SNAP.CMS	Simulate Penna Turnpike - SNAP	8-10-1995
TLC44.CMS	TLC for 44 fps parameter study	8-10-1995
TTD44.CMS	TTD for 44 fps parameter study	8-10-1995
TTD66.CMS	TTD for 66 fps parameter study	8-10-1995
TTD88.CMS	ttd for 88 fps parameter study	8-10-1995

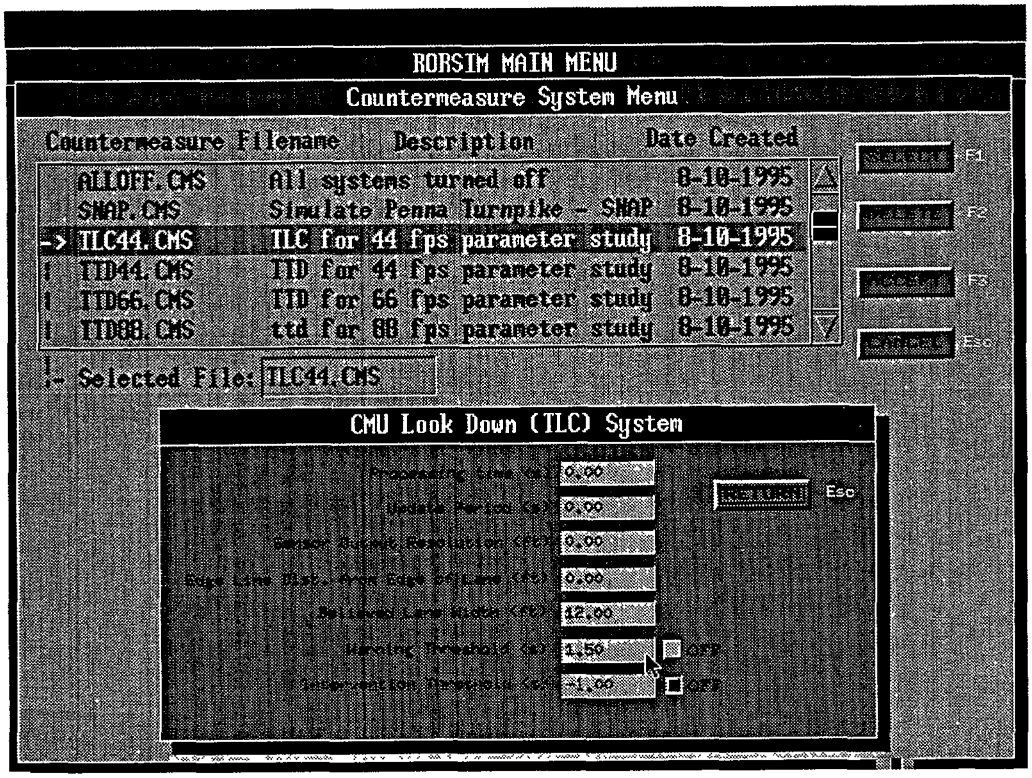
Selected File: ALLOFF.CMS



Countermeasure Filename	Description	Date Created
ALLOFF.CMS	All systems turned off	8-10-1995
SNAP.CMS	Simulate Penna Turnpike - SNAP	8-10-1995
TLC44.CMS	TLC for 44 fps parameter study	8-10-1995
-> TTD44.CMS	TTD for 44 fps parameter study	8-10-1995
TTD66.CMS	TTD for 66 fps parameter study	8-10-1995
TTD88.CMS	ttd for 88 fps parameter study	8-10-1995

Selected File: TTD44.CMS

Processing Time (s) 0.00
Update Period (s) 0.00
Sensor Output Resolution (ft) 0.00
Look Ahead Distance (ft) 66.00
Warning Threshold (s) 1.85 □ OFF
Intervention Threshold (s) -1.00 □ OFF
Tolerance Distance (ft) 3.25
Allow Cutting yes no



RORSIM MAIN MENU

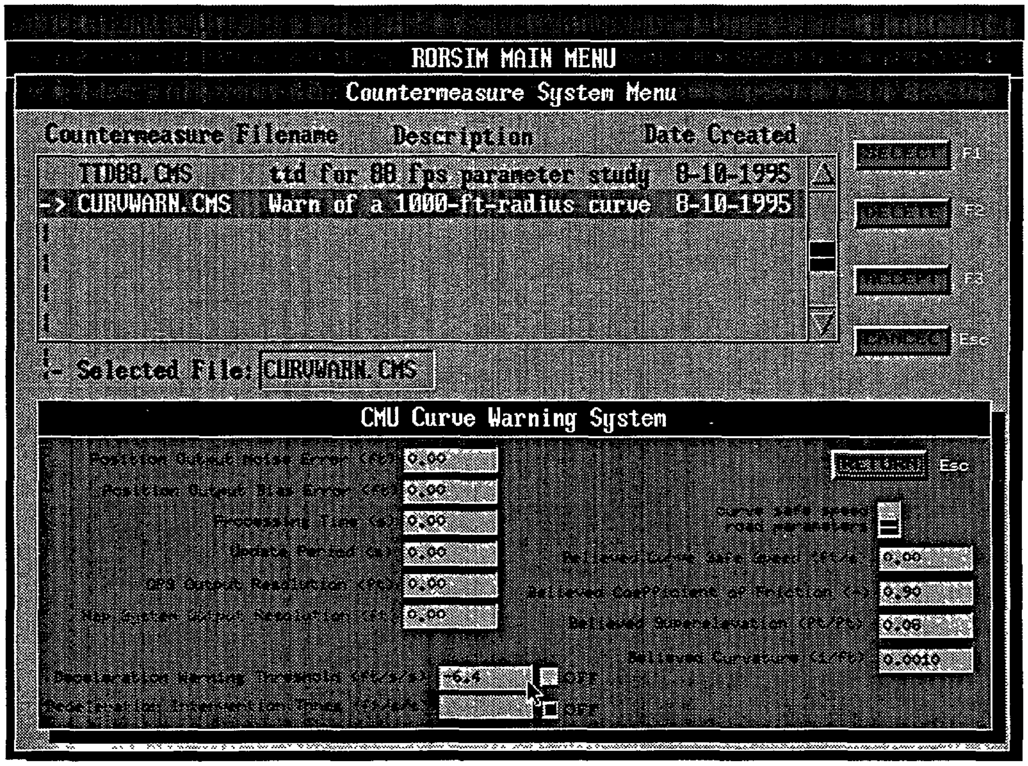
Countermeasure System Menu

Countermeasure Filename	Description	Date Created
ALLOFF.CMS	All systems turned off	8-10-1995
SNAP.CMS	Simulate Penna Turnpike - SNAP	8-10-1995
-> TLC44.CMS	TLC for 44 fps parameter study	8-10-1995
TTD44.CMS	TTD for 44 fps parameter study	8-10-1995
TTD66.CMS	TTD for 66 fps parameter study	8-10-1995
TTD88.CMS	ttd for 88 fps parameter study	8-10-1995

Selected File: TLC44.CMS

CMU Look Down (TLC) System

Processing Time (s) 0.00
Update Period (s) 0.00
Sensor Output Resolution (ft) 0.00
Edge Line Dist. from Edge of Lane (ft) 0.00
Believed Lane Width (ft) 12.00
Warning Threshold (s) 1.50 □ OFF
Intervention Threshold (s) -1.00 ■ OFF



RORSIM MAIN MENU

Countermeasure System Menu

Countermeasure Filename	Description	Date Created
TTD88.CMS	ttd for 88 fps parameter study	8-10-1995
-> CURVWARN.CMS	Warn of a 1000-ft-radius curve	8-10-1995

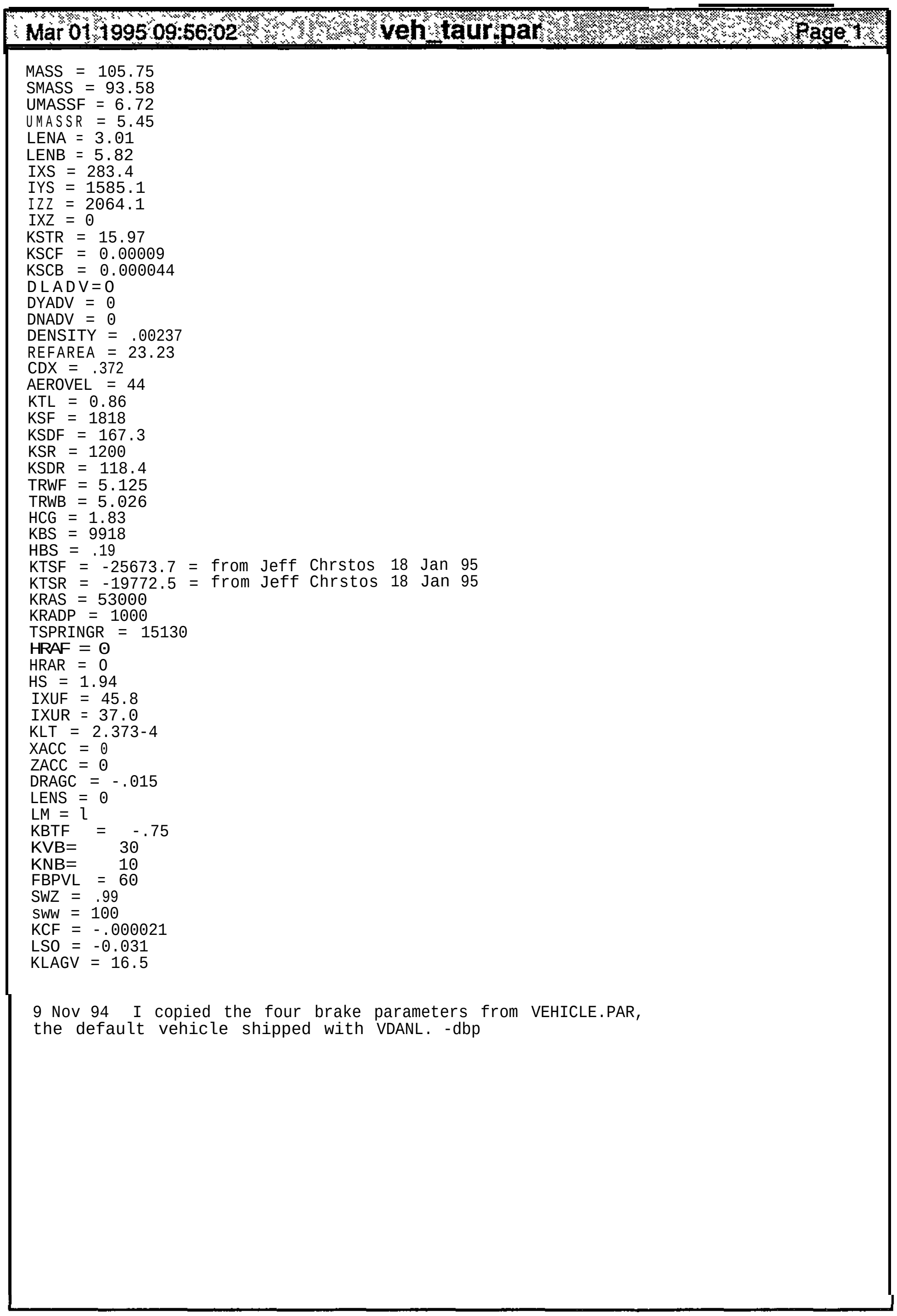
Selected File: CURVWARN.CMS

CMU Curve Warning System

Position Output Noise Error (ft) 0.00
Position Output Bias Error (ft) 0.00
Processing Time (s) 0.00
Update Period (s) 0.00
GPS Output Resolution (ft) 0.00
Map System Output Resolution (ft) 0.00
Deceleration Warning Threshold (ft/s/s) -6.4 □ OFF
Deceleration Intervention Thres (ft/s/s) ■ OFF

Curve safe speed
road parameters
Believed Curve Safe Speed (ft/s) 0.00
Believed Coefficient of Friction (-) 0.90
Believed Superelevation (ft/ft) 0.06
Believed Curvature (1/ft) 0.0010

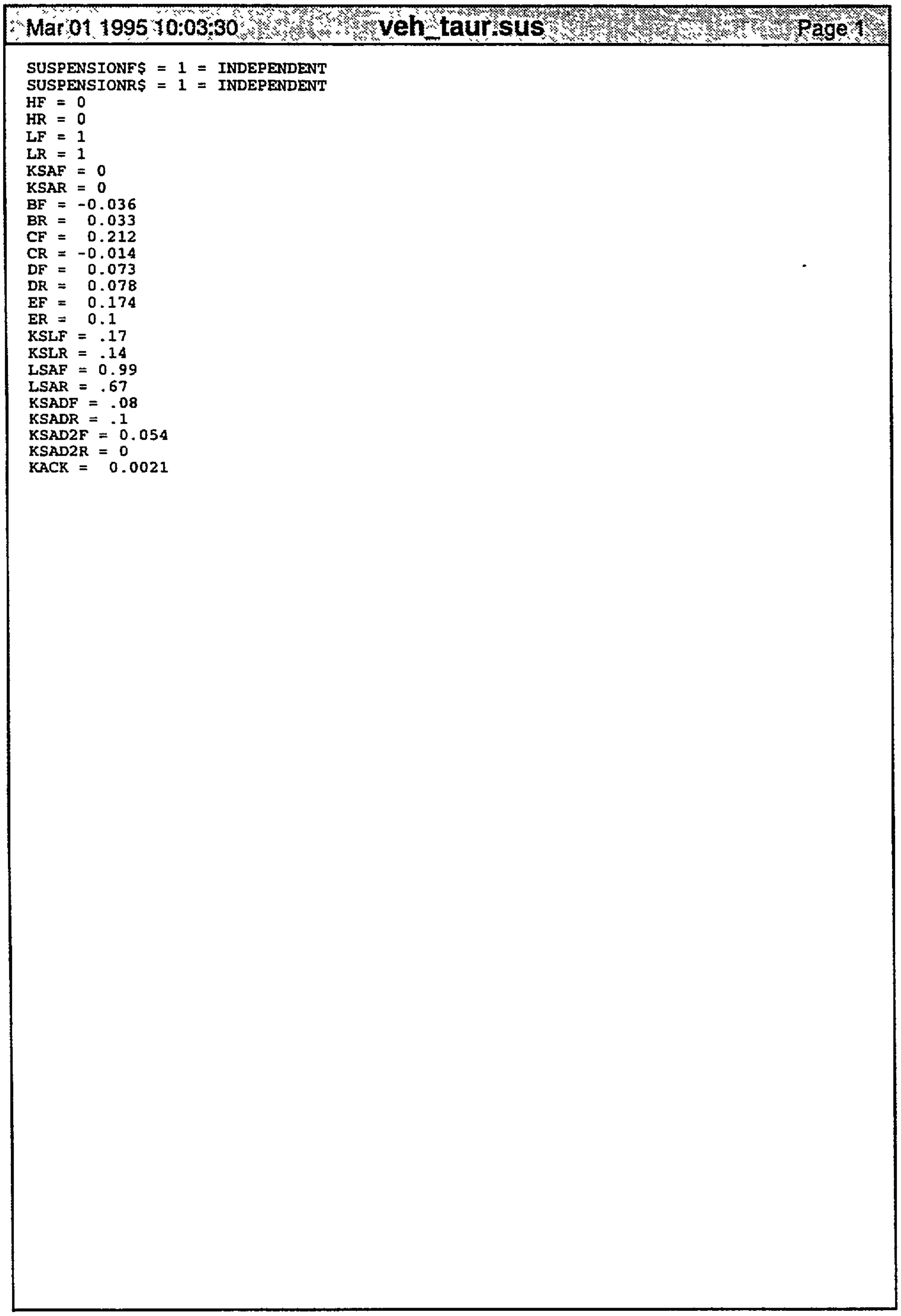
Appendix C. Vehicle Parameter List



Mar 01 1995 09:56:02 veh_taur.par Page 1

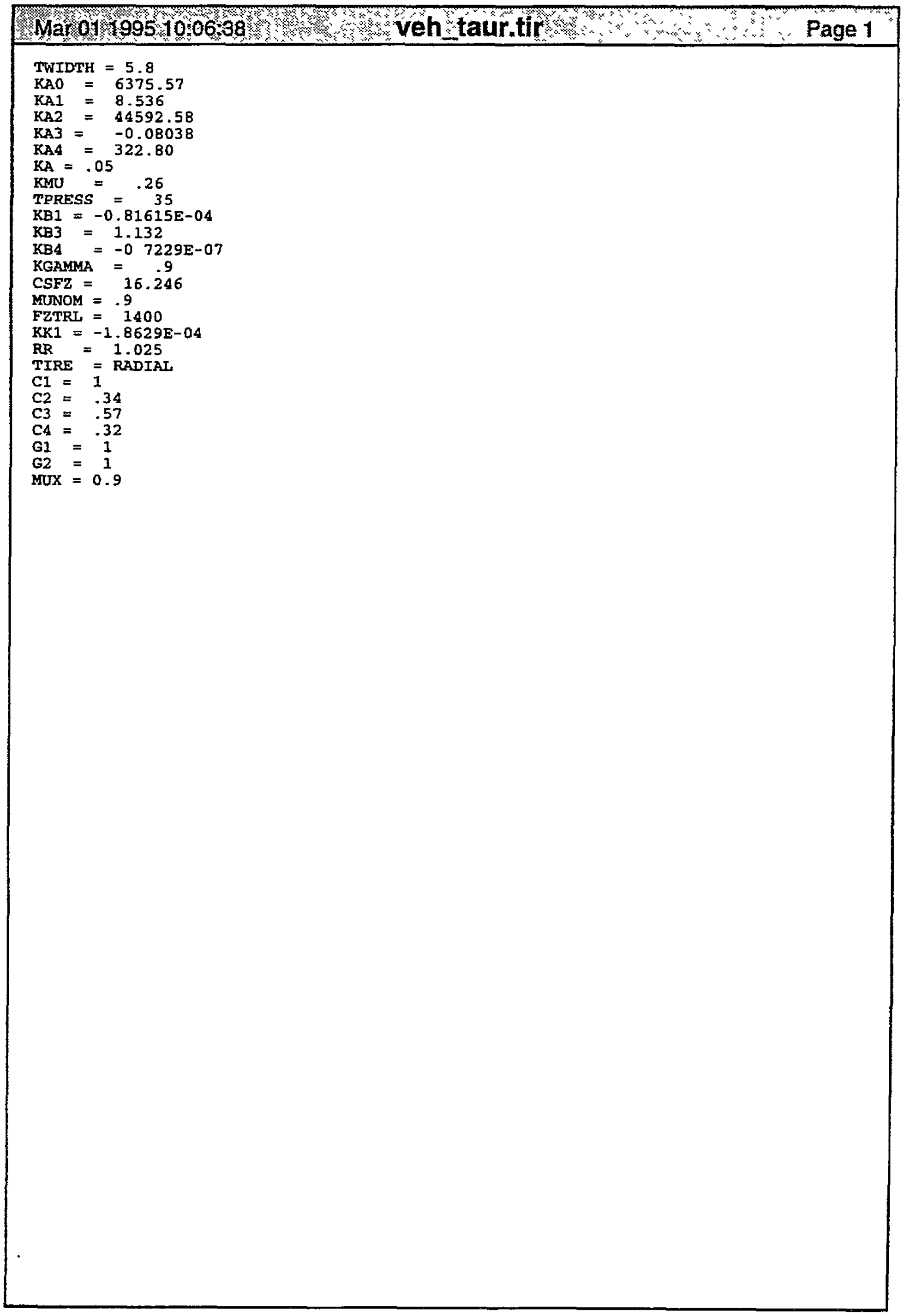
```
MASS = 105.75
SMASS = 93.58
UMASSF = 6.72
UMASSR = 5.45
LENA = 3.01
LENB = 5.82
IXS = 283.4
IYS = 1585.1
IZZ = 2064.1
IXZ = 0
KSTR = 15.97
KSCF = 0.00009
KSCB = 0.000044
DLADV=0
DYADV = 0
DNADV = 0
DENSITY = .00237
REFAREA = 23.23
CDX = .372
AEROVEL = 44
KTL = 0.86
KSF = 1818
KSDF = 167.3
KSR = 1200
KSDR = 118.4
TRWF = 5.125
TRWB = 5.026
HCG = 1.83
KBS = 9918
HBS = .19
KTSF = -25673.7 = from Jeff Chrstos 18 Jan 95
KTSR = -19772.5 = from Jeff Chrstos 18 Jan 95
KRAS = 53000
KRADP = 1000
TSPRINGR = 15130
HRAF = 0
HRAR = 0
HS = 1.94
IXUF = 45.8
IXUR = 37.0
KLT = 2.373-4
XACC = 0
ZACC = 0
DRAGC = -.015
LENS = 0
LM = l
KBTF   =   -.75
KVB=    30
KNB=    10
FBPVL = 60
SWZ = .99
sww = 100
KCF = -.000021
LSO = -0.031
KLAGV = 16.5


 9 Nov 94  I copied the four brake parameters from VEHICLE.PAR,
 the default vehicle shipped with VDANL. -dbp
```



Mar 01 1995 10:03:30 **veh_taur.sus** Page 1

```
SUSPENSIONF$ = 1 = INDEPENDENT
SUSPENSIONR$ = 1 = INDEPENDENT
HF = 0
HR = 0
LF = 1
LR = 1
KSAF = 0
KSAR = 0
BF = -0.036
BR =  0.033
CF =  0.212
CR = -0.014
DF =  0.073
DR =  0.078
EF =  0.174
ER =  0.1
KSLF = .17
KSLR = .14
LSAF = 0.99
LSAR = .67
KSADF = .08
KSADR = .1
KSAD2F = 0.054
KSAD2R = 0
KACK =  0.0021
```



Mar 01 1995 10:06:38 veh_taur.tir Page 1

```
TWIDTH = 5.8
KA0  =  6375.57
KA1  =  8.536
KA2  =  44592.58
KA3 =   -0.08038
KA4  =  322.80
KA = .05
KMU   =   .26
TPRESS  =   35
KB1 = -0.81615E-04
KB3  =  1.132
KB4   = -0 7229E-07
KGAMMA  =   .9
CSFZ =   16.246
MUNOM = .9
FZTRL =  1400
KK1 = -1.8629E-04
RR   =  1.025
TIRE  = RADIAL
C1 =  1
C2 =  .34
C3 =  .57
C4 =  .32
G1  =  1
G2  =  1
MUX = 0.9
```

Appendix D. Lists of Files

The next page has the complete list of files that you get when you install RORSIM. To run RORSIM, you need a VDANL hardware key, which you get when you buy VDANL Version 4.??, November 1994. (Yes, 4.?? really is the version number.) That will be enough for you to run simulations and get time traces of variables and ASCII output files. If you want to make a plot of the vehicle at different places along the road, you'll need AutoCAD release 10 or 12. Fortunately, AutoCAD doesn't have to be on the same computer as RORSIM; you can transfer the .LSP file by floppy disk.

```
FILE LIST FOR RORSIM INSTALLATION

 PROGS.EXE          Installs to .  /RUN directory

 Length     Date     Time   Name
 ------     ----     ----   ----
   5168  03-10-92   03:10  VGAOEM.FON      VDANL screen driver
   8862  06-27-94   08:31  TJRTOOLS.OVL    VDANL overlay
   7226  08-30-94   10:11  VDANLSTR OVL    VDANL overlay
     21  09-06-94   12·23  DUMMY.FIL       data file with zeroes
     16  12-06-94   12:42  VDANLSET.UP     VDANL setup
     30  02-09-95   17:09  PLOTTING.PAR    VDANL plotting parameters
 521980  07-12-95   15:12  RORSIM.EXE      Main RORSIM (VDANL) executable
  44498  07-13-95   13:19  RORMENU.UIR     Menu structure for RORMENU
    329  08-08-95   14:53  FILES.DUM       List of vehicle definition files
 441187  07-13-95   13:33  RORMENU.EXE     Menu executable


 CASES.EXE          Installs to .. /RUN directory

 Length     Date     Time   Name
 ------     ----     ----   ----
    874  08-10-95   10:00  ALLOFF.CMS      (See the tables in Section 3 of
    192  08-10-95   10:00  BADDRIVR.DVC     the manual for descriptions.)
    156  08-10-95   10:00  CLEARDAY.NVV
    222  08-10-95   10:00  CURV0800.RD
    250  03-09-95   15:54  SPDCNTRL.SCF
    107  07-11-95   15:45  ROR30.MAC       VDANL macro file
    221  08-10-95   10:00  CURV1000 RD
    223  08-10-95   10:00  CURV4000.RD
    273  08-10-95   10:00  CURVAT88 AUX
    508  08-10-95   10:00  CUTAT88.SCN
    175  08-10-95   10:00  FOG NVV
    196  08-10-95   10:00  LONGREAC.DVC
    191  08-10-95   10·00  NOREACT.DVC
    256  08-10-95   10:00  NORMAL.DVC
    137  08-10-95   10:00  NORMAL.NVS
    513  08-10-95   10:00  NORMAL.SCN
    272  08-10-95   10:00  PLAIN.AUX
    147  08-10-95   10:00  SLIPPRY.NVS
    888  08-10-95   10:00  SNAP.CMS
    277  08-10-95   10:00  STRAIGHT.RD
    886  08-10-95   10:00  TLC44.CMS
    886  08-10-95   10:00  TTD44.CMS
    886  08-10-95   10:00  TTD66.CMS
    886  08-10-95   10:00  TTD88.CMS
    144  08-10-95   10:00  WET.NVS
    273  08-10-95   10:00  CONFIG.AUX
    113  03-16-95   14:26  PLUSTWO4.LAN
     38  07-12-95   13:54  SPEED088.SPD
     36  07-12-95   13:54  LAN0000 LAN
    878  08-10-95   10:00  CURVWARN.CMS


 SOURCE.EXE         Installs to .../SOURCE directory

 Length     Date     Time   Name
 ------     ----     ----   ----
   5361  08-18-95   09:47  DECIDE1 BAS     Decision algorithm for system 1
  11634  07-12-95   14:34  DECIDE2.BAS     Decision algorithm for system 2
   4969  07-12-95   14:34  DECIDE3 BAS     Decision algorithm for system 3
   4189  01-09-95   09:04  DEGRADE.BAS     Degrades sensor output
   4021  05-12-95   17:19  READFILE.BAS    Reads parameter files
  17797  05-12-95   08:40  ROAD.BAS        Calculates road height, friction
   7852  07-12-95   14:05  SENSOR1.BAS     Sensor simulator for system 1
   4736  07-12-95   14:05  SENSOR2.BAS     Sensor simulator for system 2
   6945  07-12-95   14:05  SENSOR3.BAS     Sensor simulator for systme 3
  35720  07-12-95   15:11  UMROR.BAS       Primary USERMODULE function
   3300  05-12-95   09:32  RORCM.INC       Include file for RORSIM functions
    211  10-17-94   15:58  USERMOD.INC     VDANL Open Module include file
   6574  07-10-95   11:49  VDANLVAR.INC    Subscripts to VARYY and VDANLINP
    736  07-13-95   13:21  ROR LNK         Link commands


 TAURUS95.EXE       Installs to .../TAURUS95 directory
```

```
 Length     Date     Time   Name
 ------     ----     ----   ----
    295  03-01-95   09:56  NONLIN.PAR      (FILES.DUM in the RUN directory
    519  07-11-95   17:30  PARAM.INP        points to these files.  For more
    792  07-16-92   15:45  TRA: ER.PAR      information, look at the section
    322  08-02-93   14:30  TRAILER.TIR      on FILES DUM in VDANL.HLP, which
    332  11-02-94   09:48  VEH_DBP.DRI      came with VDANL.)
    561  04-20-94   16:23  VEH_FWD.DRT
    907  03-01-95   09:56  VEH_TAUR.PAR
    339  03-01-95   10·03  VEH_TAUR.SUS
    380  03-01-95   10.06  VEH_TAUR TIR


 RORPLOTZ.EXE       Installs to .../RORPLOT directory

 Length     Date     Time   Name
 ------     ----     ----   ----
1127072  08-21-95   10:10  RORPLOT.EXE     Generates AutoCAD script files
```

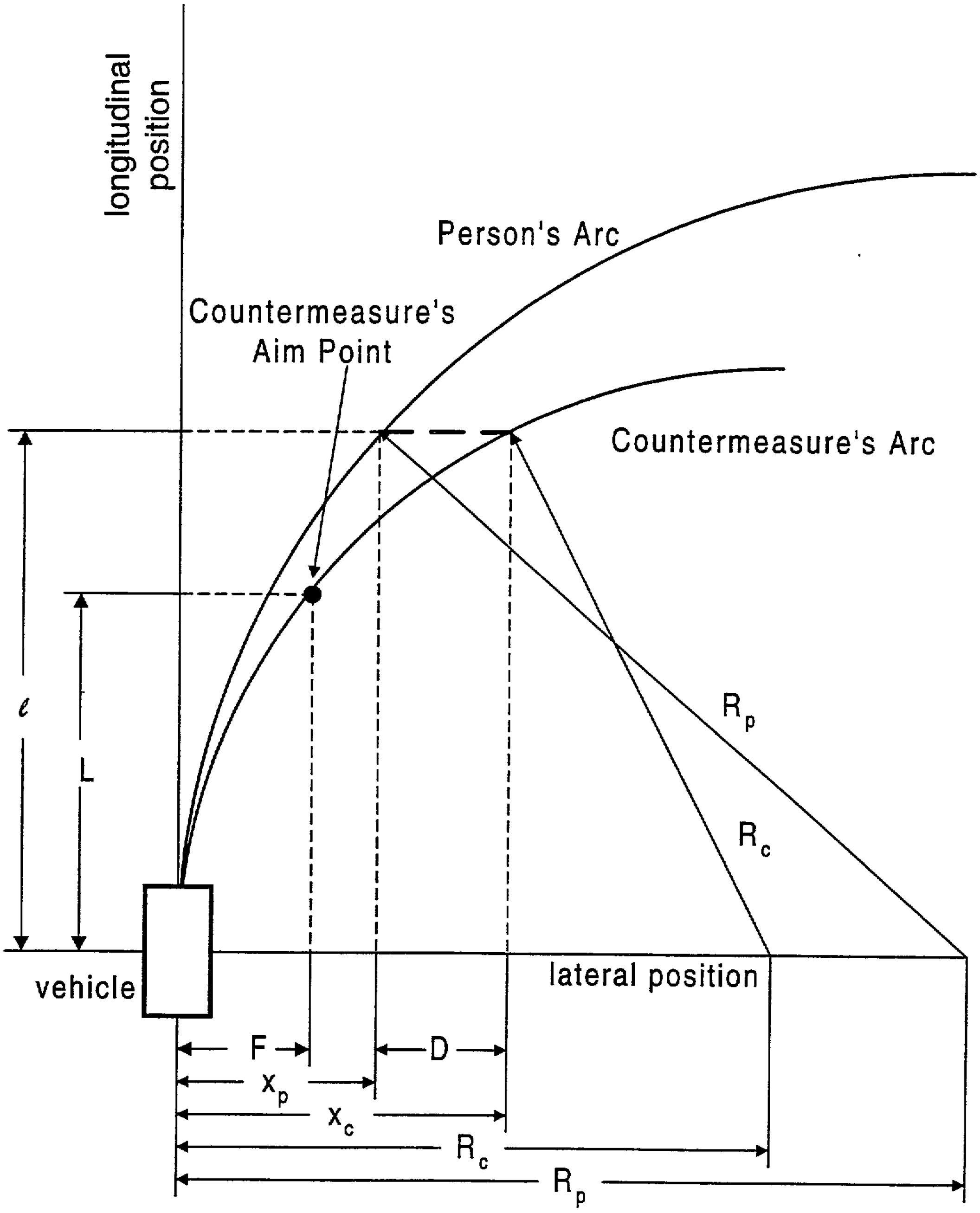
Appendix E. Plotting Variables in RORSIM

This is a list of the 30 variables that are saved by RORSIM to be plotted or written as output. The units of VDANL (and RORSIM) are lb (force), ft, s, and radians. The global coordinate system has positive X in the direction the vehicle points at the beginning of the simulation, positive Y is to the right, and positive Z is down. Lateral positions in the local (road-based) coordinate system (such as Yi and the tire positions) are measured to the right from the lane center. The code number is the subscript in the array VARYY in the BASIC code.

			Special	
		Native to	for	
Code	Name	VDANL	RORSIM	Description
311	Trip		X	Distance down the road traveled by the vehicle (odometer)
71	Yi	X		Distance from the lane center to the vehicle cg
314	PsiOff		X	Angle from the lane tangent to the vehicle longitudinal axis
305	Lwidth		X	Width of the travel lane
306	AheadRPsi		X	Global angle of the road, at the look-ahead point
307	AheadRX		X	Global x position of the road, at the look-ahead point
308	AheadRY		X	Global y position of the road, at the look-ahead point
57	longp	X		Global x position of the vehicle center of gravity (cg)
313	Y	X		Global y position of the vehicle center of gravity (cg)
49	longvel	X		Longitudinal velocity of the vehicle
54	ax	X		Longitudinal acceleration of the vehicle
144	Athrot	X		Throttle position (0 to 1)
36	FBP	X		Force on the brake pedal
29	6sw	X		Handwheel (steering) angle
42	ϕs	X		Roll angle of the sprung mass
44	psid	X		Yaw rate (time derivative of psi)
45	psi	X		Angle from the global x axis to the vehicle longitudinal axis
301	TrueYlf		X	Distance from the lane center to the vehicle's left front tire
302	TrueYrf		X	Distance from the lane center to the vehicle's right front tire
331	warn1		X	Warning signal from Countermeasure system #1[1]
332	warn2		X	Warning signal from Countermeasure system #2[1]
333	warn3		X	Warning signal from Countermeasure system #3[1]
75	Ye	X		Error in commanded lane position
343	F		X	Offset to lane center at the look-ahead point[2]
344	TTD		X	Time to Trajectory Divergence function
345	TLC		X	Time to Line Crossing function
346	vc		X	Estimated safe speed for an upcoming curve
347	DistAway		X	Estimated distance to the entry of an upcoming curve
91	CURVEC	x		Instantaneous road curvature
73	CurveER	X		Curvature error in the closed-loop steering model

[1] See Table 2-l for an explanation of the warning outputs.

[2] F is the distance from the vehicle's longitudinal axis to the lane center point, measured at the look-ahead distance. See the figure on the following page.



Geometry for calculating TTD (Time to Trajectory Divergence). The look-ahead distance, L, and the tolerance distance between the arcs, D, are fixed before the measurement. The sensor returns the lane center offset, F. At distance *l* ahead of the vehicle, the two arcs are D apart.

Index