DEVELOPMENT AND VALIDATION OF THE
PENNSYLVANIA TRUCK DRIVING SIMULATOR

A Thesis in
Mechanical Engineering
by
Andrew H. Hoskins

© 2002 Andrew H. Hoskins

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2002
I grant The Pennsylvania State University the nonexclusive right to use this work for the University's own purposes and to make single copies of the work available to the public on a not-for-profit basis if copies are not otherwise available.

Andrew Howard Hoskins
We approve the thesis of Andrew H. Hoskins.

Moustafa El-Gindy
Senior Research Associate
Thesis Advisor

H. J. Sommer III
Professor of Mechanical Engineering

Richard C. Benson
Professor of Mechanical Engineering
Head of the Department of Mechanical Engineering
ABSTRACT

The Penn State Truck Driving Simulator (PTDS) was donated to the university by a consortium of companies. The simulator consists of eight personal computers, a Mack Trucks cab, a Moog INC. motion base and real-time simulation software developed by Renault and Systems Technology, INC. Validating the simulator is the focus of the work herein.

The simulator incorporates a 6 DOF motion base that moves the entire truck cab. The motions are created in response to driver input (gas pedal, brake pedal, clutch, gear shift, steering wheel) and the terrain. The motions created by the motion base are intended to simulate the forces the driver would undergo during driving. The extent that these motions represent real life accelerations within a vehicle of the same dimensions was evaluated. Further, a study conducted to evaluate the simulator as a training tool for heavy vehicle operators allowed the comparison of measurements generated by the simulator data collection device with professional heavy vehicle driving instructor assessments of driver performance. The results of this comparison are included in this document.

The simulator has been upgraded many times since its construction through introduction of new hardware and new software. The upgrades to the simulator have not always been documented. Accurate documentation of the current architecture of the simulator is included in this thesis to ensure useful operation by other investigators in the future.
TABLE OF CONTENTS

LIST OF FIGURES ........................................................................................................................... VII

LIST OF TABLES ............................................................................................................................... X

ACKNOWLEDGMENTS ...................................................................................................................... XI

CHAPTER 1 LITERATURE SURVEY ON DRIVING SIMULATOR VALIDATION STUDIES .......................................................... 1

Introduction ........................................................................................................................................ 1

Simulator Validation Studies ........................................................................................................... 3
  Driver Behavior Validation Studies ................................................................................................. 4
  Model Validation ............................................................................................................................. 13
  Overview of VDANL ......................................................................................................................... 17

Heavy Vehicle Driver Training Scenario Development ............................................................... 21

CHAPTER 2 HISTORICAL BACKGROUND AND DEVELOPMENT OF THE PENNSYLVANIA TRUCK DRIVING SIMULATOR .................................................................................. 23

Historical Background .................................................................................................................... 23

Simulator Upgrades ....................................................................................................................... 25
  First Use of the Simulator in a Funded Project ............................................................................. 26
  Projection upgrades and tuning ........................................................................................................ 26
  Front projection tuning .................................................................................................................. 27
  Rear Projection Installation and Tuning ......................................................................................... 30
  Visual Generation Computer Upgrade ......................................................................................... 32
  Backup Alarm ................................................................................................................................. 33
  Pressurized Air Connection ............................................................................................................ 34
  CCD Camera Installation ............................................................................................................... 34
  Data Recording ............................................................................................................................... 36

Current Simulator Architecture ..................................................................................................... 37
### CHAPTER 3 SIMULATOR VALIDATION

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>39</td>
</tr>
<tr>
<td>Motion Base Control</td>
<td>39</td>
</tr>
<tr>
<td>Validation Methodology</td>
<td>39</td>
</tr>
<tr>
<td>Equipment Configuration</td>
<td>42</td>
</tr>
<tr>
<td>Timer Synchronization</td>
<td>44</td>
</tr>
<tr>
<td>Experimental Procedure</td>
<td>46</td>
</tr>
<tr>
<td>Data Recording During Trials</td>
<td>52</td>
</tr>
<tr>
<td>VDANL Simulations</td>
<td>55</td>
</tr>
<tr>
<td>Results</td>
<td>56</td>
</tr>
<tr>
<td>Straight-Line Braking</td>
<td>59</td>
</tr>
<tr>
<td>Double Lane Change Maneuver</td>
<td>68</td>
</tr>
<tr>
<td>Discussion</td>
<td>73</td>
</tr>
<tr>
<td>Straight-Line Braking</td>
<td>74</td>
</tr>
<tr>
<td>Double Lane Change Maneuver</td>
<td>76</td>
</tr>
</tbody>
</table>

### CHAPTER 4 SCENARIO DEVELOPMENT WITHIN THE PENNSYLVANIA TRUCK DRIVING SIMULATOR

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>77</td>
</tr>
<tr>
<td>Real-time Simulation Components Within the PTDS</td>
<td>77</td>
</tr>
<tr>
<td>Configuration Hierarchy Within SCANeR® II</td>
<td>78</td>
</tr>
<tr>
<td>Developing Scenarios in Mice</td>
<td>80</td>
</tr>
<tr>
<td>Traffic Generation</td>
<td>81</td>
</tr>
<tr>
<td>Developing Other Scenario Elements</td>
<td>85</td>
</tr>
<tr>
<td>Recording Data With the Simulator</td>
<td>87</td>
</tr>
<tr>
<td>Limitations of Recording Data</td>
<td>89</td>
</tr>
<tr>
<td>Training Scenario Development</td>
<td>90</td>
</tr>
<tr>
<td>55 MPH Scenario</td>
<td>93</td>
</tr>
<tr>
<td>Bridge Turn Scenario</td>
<td>96</td>
</tr>
<tr>
<td>Train Track Scenario</td>
<td>98</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>99</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1.1: The TNO Driving Simulator during a driving simulation (TNO, 2002)........ 5
Figure 1.2: The Daimler-Benz Simulator. (Riemersma et al, 1990).......................... 7
Figure 1.3: Data flow for simulation validation study. (Heydinger et al, 1990)......... 16
Figure 1.4: The National Advanced Driving Simulator, shown here, incorporates a
hexapod motion base mounted on a translating frame with a complete,
interchangeable vehicle cab mounted in the projection dome on top of the motion
system (NADS, 2002). ........................................................................................................ 19
Figure 2.1: Pennsylvania Truck Driving Simulator during a simulation ............ 24
Figure 2.2: Screen and viewpoint parameter illustration ......................................... 29
Figure 2.3: Rear projection viewpoint and screen configuration......................... 31
Figure 2.4: CCD camera installed in the simulator cab ........................................ 35
Figure 3.1: Pitch motion contributing to longitudinal acceleration ..................... 41
Figure 3.2: Simulator validation methodology diagram ........................................ 44
Figure 3.3: DMU mounted in the simulator’s cab .................................................. 45
Figure 3.4: Dual output keyboard for synchronizing the DMU and the PTDS ...... 48
Figure 3.5: Mechanical stop installed in the PTDS for use in straight-line braking
maneuvers ...................................................................................................................... 50
Figure 3.6: Markers placed near the PTDS steering wheel for double lane change driving
maneuvers ...................................................................................................................... 51
Figure 3.7: Calibration window for CabinIO.exe .................................................. 57
Figure 3.8: Select braking input profiles for VDANL simulations and simulator trials for
36% braking effort (top) and 68% braking effort (bottom) ................................. 60
Figure 3.9: Select velocity profiles for VDANL simulations and simulator trials for 36%
braking effort (top) and 68% braking effort (bottom) ............................................. 61
Figure 3.10: Pitch angles predicted by VDANL and measured in the simulator cab for 36% braking input.................................................................................................................. 63

Figure 3.11: Pitch angles predicted by VDANL and measured in the simulator cab for 68% braking input.................................................................................................................. 64

Figure 3.12: Pitch velocities predicted by VDANL and measured in the simulator cab for 36% braking input.................................................................................................................. 65

Figure 3.13: Pitch velocities predicted by VDANL and measured in the simulator cab for 36% braking input.................................................................................................................. 66

Figure 3.14: Longitudinal accelerations predicted by VDANL and measured in the simulator cab for 36% braking input.................................................................................................................. 67

Figure 3.15: Longitudinal accelerations predicted by VDANL and measured in the simulator cab for 68% braking effort.................................................................................................................. 68

Figure 3.16: Steering input for lane change maneuver trials................................................................. 69

Figure 3.17: Speed profiles for lane change maneuver trials................................................................. 70

Figure 3.18: Roll angle predicted by VDANL and measured in the simulator cab............................... 71

Figure 3.19: Roll velocity predicted by VDANL and measured in the simulator cab............................... 72

Figure 3.20: Lateral accelerations predicted by VDANL and measured in the simulator cab.......................... 73

Figure 4.1: Supervisor GUI.................................................................................................................... 79

Figure 4.2: Vehicle configuration window within the traffic generation portion of scenario development.................................................................................................................. 82

Figure 4.3: Mice GUI showing input format ............................................................................................ 86

Figure 4.4: Scenario for recording data open in Mice showing necessary context ......................... 88

Figure 4.5: View behind truck during “Tunnel Backup” scenario ............................................................ 92

Figure 4.6: View from simulator cab exiting tunnel onto snow-covered road in the “Winter Driving” scenario .................................................................................................................. 93

Figure 4.7: View from simulator cab approaching poorly marked turn in the “55 MPH” scenario .................................................................................................................. 94

Figure 4.8: 55MPH scenario results (Vance et al, 2002 and Hoskins et al, 2002 a and b) 95
Figure 4.9: View from simulator cab approaching one lane bridge in the “Bridge Turn” scenario ........................................................................................................................................... 96

Figure 4.10: Bridge Turn performance results (Vance et al, 2002) ................................................................. 97

Figure 4.11: View from simulator cab approaching railroad crossing in the “Train Track” scenario ........................................................................................................................................... 98

Figure 4.12: Train Track scenario results (Vance et al, 2002) ........................................................................ 99
LIST OF TABLES

Table 2.1: Computers of the Pennsylvania Truck Driving Simulator ....................... 37
Table 2.2: Simulator hardware model numbers ......................................................... 38
Table 4.1: Directories in the PTDS computers used for simulations ......................... 78
Table A.1: IP Addresses for computers in the TruckSim workgroup ......................... 112
Table C.1: Files required for a complete dynamics model within VDANL ............... 117
ACKNOWLEDGMENTS

The author would like to thank Dr El-Gindy for his support and input to all of the research described in this document. Many thanks are given to Brian Kleback and James Metzgar for their tireless volunteer efforts in improving and working with the simulator throughout the work described within. Special thanks are given to Sam, Peg, and Matt Hoskins for their financial and emotional support without which this work would not have been possible.
CHAPTER 1
LITERATURE SURVEY ON DRIVING SIMULATOR VALIDATION STUDIES

Introduction

Before attempting to measure the validity of a simulator it is important to identify what is meant by “validity”. Rose et al (1987) describe validity as “the extent to which a model serves its purpose for a particular training device or devices.” Because vehicle dynamics simulations are critical elements of all simulators, validity of the simulation is paramount. Garrott et al (1997) state that a simulation “will be considered valid if, within some specified operating range of the physical system, a simulation’s predictions of the system’s responses of interest to specified input(s) agree with the actual physical system’s responses to the same input(s) to within some specified level of accuracy.”

All simulators are based on mathematical models of the systems they represent. It is the accuracy of these models and how they are implemented that determines the validity of the simulator. Jamson (1999) states that there exist two primary areas of simulator validation, “behavioral” and “physical” validation. The first refers to a simulator’s ability to induce the same response from a driver as would be performed in the same situation in real life (Jamson, 1999). Physical validity measures the degree to which the simulator dynamics and visual system reproduce the vehicle being simulated (Jamson, 1999).

Kaptein, Theeuwes, and van der Horst (1996) use four different types of validity. These are absolute validity, relative validity, internal validity, and external validity. A
simulator is said to have absolute validity for a given task if the effect seen within the simulator is of the same scale as the effect seen in real life. A simulator has relative validity if the same trend of an effect for a given task is seen within the simulator and in real life. A simulator with internal validity allows researchers to exclusively associate a given effect on a person’s driving with a specific cause. External validity is the extent to which the results of a task at a certain time and place with a certain subject can be generalized to hold true for other subjects at other places and times.

Kaptein, Theeuwes, and van der Horst (1996) state that the validity of a simulator must be considered task dependent. Task dependency of validity is associated with the cues provided by the simulator and the extent to which these cues provide adequate information to the driver for the purpose of completing the task. An example of a simulator that is not valid for a given task is a simulator that has a 90-degree lateral field of view in front projection simulating the scenario of a 4-way intersection with crossing traffic that does not stop. The driver would not have the necessary cues of seeing the traffic approaching from either side due to the geometry of the visual generation equipment. This simulator might however be considered valid for completing a serpentine course that traverses a road adequately depicted by the visual generation equipment.

Garrott et al (1997) when describing the validation of vehicle dynamics models similarly noted that a dynamics model would usually only be accurate in predicting the vehicle response within some subset of the possible input conditions. Within driving simulators, vehicle dynamics models generate many of the cues a driver is exposed to
while operating the simulator. This can be illustrated by the following example. For a simple driving input of making a 90-degree turn at an intersection, a dynamics model is configured such that the road-tire coefficient of friction is significantly reduced from what is normally experienced on dry pavement. A driver attempts to make the turn but proceeds to slide through the intersection. In this case the visual generation system showed the driver continuing to travel straight, the motion platform reflected reduced lateral acceleration from that experienced on dry pavement and the corrective steering torque was significantly less than is expected in the dry road condition. All of these responses occurred because of the vehicle dynamics predictions for the driving input. Without validation of the vehicle dynamics there is only speculation that a given model accurately predicts a vehicle response.

**Simulator Validation Studies**

A large number of studies have attempted to assess the validity of specific simulators and simulation models. The sections that follow present several of these studies to illustrate methods that have been employed for the purpose of simulator and/or model validation. Similar to the report by Jamson (1999), a distinction has been made between studies that focused on the driver responses within simulators and studies that focused on the accuracy of dynamics model predictions within the simulator. While there are a large number vehicle dynamics models available today, only a select few that are implemented in driving simulators will be considered here. By no means should this be considered a complete review of all such studies on driving simulators.
Driver Behavior Validation Studies

The TNO driving simulator is a fixed base car-driving simulator located in the Netherlands. The driver interface is a Volvo 240 cabin shown in Figure 1.1. Kaptein, Theeuwes, and van der Horst (1996) presented a survey of the validation studies that had been conducted on this simulator. The methodology of the studies discussed is comparing driver behavior for real-life driving trials with behavior exhibited in the simulator for a similar driving experience. A study that follows this approach examined lateral position on the road and speed control for drivers of an instrumented car and the simulator. A comparison of the data showed that subjects drove the simulator faster and had a greater variation in lane position than did the subjects on the real road (Blaauw, 1984). For this study the simulator exhibited relative validity with respect to speed control and lateral position variation (Kaptein, Theeuwes, and van der Horst, 1996).

Another study with the TNO driving simulator evaluated the visual information available to the driver in attempting to identify what information was critical to the driver’s performance (Kaptein, van der Horst, and Hoekstra, 1996). In this study the effect of scene complexity and field of view were examined while subjects executed a braking maneuver. The subjects were told to refrain from braking until the last possible moment while approaching a parked vehicle so that a collision could be avoided. Driver performance varied only slightly with changes in field of view and scene complexity (Kaptein, van der Horst, and Hoekstra, 1996). Drivers did however exhibit better control with a larger field of view (Kaptein, van der Horst, and Hoekstra, 1996). Typically there
was relative validity when the results of the field study were compared to the simulator study (Kaptein, van der Horst, and Hoekstra, 1996).

Figure 1.1: The TNO Driving Simulator during a driving simulation (TNO, 2002)

A study presented by Blaauw (1982) using the TNO driving simulator looked at another aspect of driving in regard to variations in the driving population. In this study a group of subjects that were classified as being either experienced or inexperienced with regard to driving were selected from the population. The subjects then drove an instrumented vehicle on the highway and the TNO driving simulator on a simulated highway. During the experiments the subjects were placed into one of four groups. These groups were distinguished by the instructions to pay attention to lateral position, speed, both speed and lateral position or neither speed nor lateral position. Vehicle speed and steering wheel angle were recorded during each of the experiments. Comparisons
between the subsets of subjects in the car and simulator tests revealed the simulator had relative validity for most experimental conditions.

A validation study using the Daimler-Benz simulator (Figure 1.2) located in Berlin, Germany was reported by Riemersma et al (1990). This simulator has a complete car enclosed in a circular projection screen. This entire unit is mounted on a 6 degrees-of-freedom (DOF) motion platform. The study evaluated effects that different road markings and signs had on driver speed choice entering a village, Weiteveen, The Netherlands. Speed measurements were taken on real traffic entering the town prior to the implementation of the signs and road markings. The speed measurements were repeated in the same locations after the speed reducing measurements had been introduced. For the simulation part of the study the landscape of the entrance to Weiteveen was recreated within the simulator. Speed reduction was evaluated for the cases of no speed reducing measures and with the speed reducing measures. The results of these studies showed that the drivers in the simulator reduced their speed more than the drivers did in real life representing relative validity for the task studied.
Several validation studies have been completed investigating different aspects of the Leeds Advanced Driving Simulator (LADS) located at the University of Leeds, UK. LADS is a fixed base simulator with a complete Rover 216GTi car for the user interface. A study by Carsten et al (1997) used a similar approach to that of the study by Riemersma et al (1990) described above. For this particular study speed and lateral position were measured at 21 different locations along an 8km section of road. A road with the same profile was then created within LADS. 100 drivers (50 male, 50 female) then drove the simulator over the road while the lane position and speed were recorded. Comparisons were then made between the real and simulated results. The mean speeds at
the locations did not differ significantly suggesting absolute validity in regard to speed. The lateral position data showed relative validity between the simulator and the real world results.

Groeger et al (1999) performed two studies examining LADS in the context of speed and distance estimation. In the first of these studies, subjects were positioned in the driver’s seat of the simulator and viewed the screen as they would when driving the simulator. The research team then introduced an object into the viewing area. The object was either a familiar object (red London bus) or a control object (red box of similar dimensions to the bus). The subject then provided researchers with two forms of information about the scene they were viewing: an estimate of the distance from their vehicle to the object; and a bisection of the distance either by driving the simulated vehicle or positioning a marker halfway between the vehicle and the object. The second study by Groeger et al (1999) investigated speed perception in LADS. The subjects were positioned as an operator would be in the simulator and were presented with a scene depicting the vehicle moving at a certain speed. The subjects then had to estimate how fast they were traveling and either increase the speed to twice the original or decrease the speed to half the original. The results of both studies were compared with field tests of other researchers. The researchers concluded that the simulated environment produced similar judgment errors in speed and distance perception as found in normal driving.

A study by Jamson (1999) using LADS investigated the effect of driver experience on negotiating curves. The experiment utilized two different groups of subjects: inexperienced subjects (drivers that had not passed the UK driving test) and
experts (drivers with a mean of 17 years driving experience). The subjects drove a
course that included two right-handed, circular curves. Speed, position and driver inputs
were recorded for each trial. This information was used to calculate several driver
performance measures such as number of centerline crossings and standard deviation in
steering position. Comparison of the results for the inexperienced and expert drivers
showed that the inexperienced drivers displayed less control while negotiating the turns.
These results supported a field study that had previously investigated issues of vehicle
control and driver experience.

The use of a telephone while driving was used as a means to validate the Driver
Interface Research Simulator at the University of Michigan Transportation Research
Institute (Reed, 1995). This simulator is a fixed base simulator with a 1985 Chrysler
Laser cab for the driver station. Subjects of this study performed driving tasks in an
instrumented car as well as in the simulator. While driving specified routes the subjects
dialed phone numbers that were displayed on cards next to the instrument panel of the
car. While completing these tasks, driver inputs, vehicle speed and lane position were
recorded. The results of the simulator trials and on-road trials were compared in order to
validate the simulator and showed that the simulator had absolute validity with respect to
vehicle speed but was less valid with respect to lane position.

A study by Klee et al (1999) performed at the University of Central Florida (UCF)
examined the validity of the UCF driving simulator with respect to driver speed. The
UCF driving simulator is a fixed-base driving simulator with a 1983 Dodge Aries station
wagon for the user interface. In this study speed measurements were taken at 16
locations along a route within the UCF campus road system. The route for the experiment was recreated in the simulator. Subjects first drove the route on the UCF campus and repeated the test loop in the simulator. A statistical analysis of speed data revealed that at a majority of the locations drivers drove the real car and the simulator at similar speeds suggesting relative validity for speed control in this simulator.

Carter and Laya (1998) investigated the validity of a fixed base simulator at the French National Institute For Transport and Safety Research (INRETS) with respect to the driver’s visual targets, termed “visual search”. The study implemented a head-mounted device that tracked the position of the subject’s eye. Subjects completed driving tasks both on the road and in the simulator while wearing the head-mounted device. The results of this particular study were somewhat inconclusive as to the validity of using simulators for studies on visual search in regard to driving condition. However, the authors note some limitations of the simulator that may have contributed to the results of the study such as the lack of motion feedback and engine noise. The authors conclude that there is potential for simulators to exhibit relative validity for studies on eye motion in the context of “spatial distributions of fixations on the visual field” (Carter and Laya, 1998).

Harms (1996) reported a validation study using the Swedish Road and Transport Research Institute (VTI) Driving Simulator. This simulator has a motion base to incorporate vehicle motion into the simulated experience. The study incorporated an instrumented car for on-road trials. Speed and lane position were recorded for all trials. Comparison of the results showed a strong correlation for speed and a weak correlation
for lane position between the on-road and simulator trials. This result is consistent with other experiments of this kind.

Staplin (1995) reported a study investigating the effect of driver age on perceived safe minimum distance from oncoming traffic in the act of making a left-handed turn across the path of the oncoming traffic. The study included controlled field experiments as well as experiments for three different presentations of the visual environment: television, video projection, and cinematic. The effects of the differences in size and resolution on driver performance were evaluated along with the effect of driver age. The results obtained with the cinematic projection and those of the field experiments showed relative validity for the simulator over the different age groups. Not surprisingly, the results showed that both size and resolution were important factors to be considered in designing laboratory experiments to evaluate simulators.

A report by Duncan (1998) on calibration studies of the Transport Research Laboratory (TRL) simulator in Crowthorne, UK revealed several approaches to behavioral validation studies of driving simulators. Within subject comparisons of 47 subjects driving results were made for data collected when driving an instrumented car on a test track and driving the simulator on a simulated version of the test track. Within the study, several experiments were performed in the simulator and in the instrumented car. One of these was speed estimation. For this measure the speedometer was obstructed so that the drivers could not see the speedometer and were asked to drive the test track at 45 mph. Speed was recorded for each task. The subjects then drove the same route with the speedometer exposed with the same instructions of maintaining a speed of 45 mph.
Finally another trial was executed with the same conditions as the first. These trials were then run traveling the route in the opposite direction. Subjects also completed a braking test where they were asked to travel at 45 mph into a series of 3 sets of cones placed such that the first set was 33 m from the last and the second was 22 m from the last. The subjects were first instructed to apply braking pressure beginning at the first set of cones and bring the vehicle to a complete stop even with the last set. This was completed three times. The next trial was to complete the same test but not start braking until the second set of cones was reached with the objective of bringing the vehicle to a stop at the third set of cones. This was also completed three times. A preferred speed and lateral position test was also part of the study. Subjects were instructed to traverse a route on the test track at a speed that they deemed safe so long as it was below the 60mph maximum speed for the track. Lateral position of the vehicle was recorded, as was the preferred speed as a way to get a candid representation of the subjects’ impressions for comparison between the car and simulator. Next the subjects’ were asked to drive at 45 mph and instructed to drive as close as possible to the center of the lane. Once this was finished the subjects were asked to complete visual tests presented on a screen mounted on the dashboard. The tests were simply looking at a group of 8 numbers and identifying if the number 2 was in the group. The sets of numbers were displayed for 5 seconds and there were 12 sets of numbers. The subjects’ response to the set was given through the left and right turn signals. Each of these was completed 3 times and then the direction of travel was reversed for three more trials. Following distance was investigated by having subjects follow a vehicle with the instructions of maintaining a constant of 30m between
the vehicles. This distance was illustrated to the subjects by the position of the vehicles at the start of the trial. The subjects were also told that the vehicle they were following would travel between 30 and 45mph and abrupt speed changes could be expected. The brake lights of the vehicle that the subjects were following were disconnected so that brake signals would not be a definitive clue to warn that the vehicle was slowing down. The subjects were then instructed to follow the same vehicle over the same course and maintain a safe following distance. These two trials were repeated traveling in the opposite direction. The speed estimation was found to be similar in both environments with more variability between subjects in the simulator. The braking test results showed that performances in the simulator varied more than those in the real car. The following distances tended to be greater in the simulator than in the real world trials. The subjects generally had more variation in lateral position in the simulator than in the real world driving exercises. The overall conclusions were that the TRL driving simulator was valid for inducing driver responses for most situations.

Model Validation

This section describes work that has been done to validate vehicle dynamics simulation models, specifically two models used in real-time driving simulators, NADSDyna and Vehicle Dynamics Analysis Non-Linear (VDANL). A background of model development and validation is also presented.

A primary objective of modeling mechanical systems is predicting system responses accurately through mathematical representations of system behaviors.
Simulations of vehicle dynamics were developed as early as the 1950’s (Bernard and Clover, 1994). Validation work investigating early simulations often included adjusting parameter measurements to increase the accuracy of the simulation prediction as well as neglecting to investigate frequency response characteristics (Heydinger et al, 1990). Furthermore, many of the early validation studies did not include testing vehicle maneuvers that resulted in vehicle responses into the nonlinear range of performance characteristics (Heydinger et al, 1990). While adjusting vehicle parameters might give a more accurate response prediction for certain driving maneuvers, this approach does not constitute validation, as the ultimate goal of validation is to identify the reliability of a model to predict a response that has been measured experimentally (Heydinger et al, 1990).

Heydinger et al (1990) present a methodology to validate vehicle simulation models. Figure 1.3 is a flow diagram that shows the primary steps in this methodology. The flow chart of this methodology has two distinct branches both starting from the “physical system”.

1) Experimental (left)

2) Simulation (right)

The experimental component involves testing the system (a vehicle in this case) and recording not only all responses of interest but also all inputs used to achieve these responses. Heydinger et al (1990) suggest multiple trials to determine the random effects in the data. Data reduction is performed on the raw experimental data followed by ensemble averaging. Ensemble averaging takes the means of the multiple trials over the
independent variable (for example: time) to allow for calculating “experimental
repeatability” through statistical analysis (Heydinger et al, 1990). The result of this step
is then compared to the results from the simulation component of the validation. Within
the simulation section of this methodology, the first step is taking detailed measurements
of the system such as geometry, mass, inertial properties, damping and compliance
(Heydinger et al, 1990). The data collected is then used to create a model of the system
and is fed into the simulation along with the inputs recorded in the experimental testing
phase. Heydinger et al (1990) propose running simulations with inputs matching those of
each of the experimental trial runs and conducting ensemble averaging of the results of
the simulation. However note that the ensemble averaging can be done prior to running
the simulation if there is not too much variation in the input data. As with the
experimental data, the simulation results may need to be reduced prior to the final step of
making comparisons between the two sets of data.

In this methodology, both qualitative and quantitative comparisons of the final
forms of the simulator and experimental data sets are recommended (Heydinger et al,
1990).
Figure 1.3: Data flow for simulation validation study. (Heydinger et al, 1990)
Overview of VDANL

VDANL was developed during the mid 1980’s for the National Highway Transportation Safety Administration (Christos and Heydinger, 1997) and is described in volumes II (Allen et al, vol II, 1988), III (Allen et al, vol III, 1998), and IV (Allen et al, vol IV, 1998) of “Analytical Modeling of Driver Response in Crash Avoidance Maneuvering”. A version of VDANL is implemented in the Pennsylvania Truck Driving Simulator at the Pennsylvania Transportation Institute, of the Pennsylvania State University. In addition, the vehicle dynamics model for STISIM, a commercially available driving simulator from Systems Technology, Inc., is derived from VDANL (STISIM, 2002).

Early validation studies of VDANL include work presented in Heydinger et al (1990) where four vehicles were evaluated in accordance with the methodology discussed above. Two simulation packages were used in this study, VDANL and Improved Digital Simulation Fully Comprehensive (IDSFC). Results of this study showed VDANL to be more capable of predicting vehicle responses for a wider range of vehicles than IDSFC as well as noted benefits from the tire models that are included in VDANL and not in IDSFC (Heydinger et al, 1990).

Work presented in Christos and Heydinger (1997) is related to work summarized in Garrott et al (1997). In this study the parameters used to model a 1994 Ford Taurus within the NADSdyna simulation package are used in two separate modeling packages, VDANL and VDM RoAD, along with the experimental test data gathered, to validate each of these models. Christos and Heydinger (1997) concluded that, “Both simulations
do a good job of predicting expected vehicle responses in the linear range.” Further, “Both simulations are shown to predict the trends in the non-linear understeer plots.”

An improved version of the methodology described above with the same overall approach was implemented in a study presented by Garrott et al (1997) investigating the vehicle dynamics model NADSdyna, which will be used in the National Advanced Driving Simulator (NADS). NADS (Figure 1.4) has been developed at the University of Iowa and is claimed to be the “most sophisticated research driving simulator in the world” (NADS, 2002). The paper by Garrott et al (1997) reports on work done at the National Highway Transportation Safety Administration’s (NHTSA’s) Vehicle Research and Test Center to further develop and evaluate NADS. A more detailed description of certain parts of the work described in the paper by Garrott et al (1997) is given in the companion papers (Chrstos and Grygier, 1997), (Salaani, Chrstos et al, 1997), (Salaani, Heydinger et al, 1997), and (Chrstos and Heydinger, 1997). A detailed description of the NADS can be found at (NADS, 2002).

Garrott et al (1997) detail important steps to validating a vehicle dynamics model in listing the three major parts of the work described in this paper. The work pertinent to the current document takes the form of “experimental data collection”, “vehicle parameter measurement”, and “comparison of simulation predictions with experimental data”.
Figure 1.4: The National Advanced Driving Simulator, shown here, incorporates a hexapod motion base mounted on a translating frame with a complete, interchangeable vehicle cab mounted in the projection dome on top of the motion system (NADS, 2002).

Christos and Grygier (1997) describe the methods used for collecting the experimental data needed for the study detailed by Garrott et al (1997). The 1994 Ford Taurus used in the study was equipped with more than 40 different transducers to record different vehicle parameters during testing. The 75 different driving experiments resulted in almost 600 trial runs. Modifications to the vehicle prior to testing were implemented to allow for consistent driver inputs such as a mechanical stop on the steering wheel and actuators for the brake and accelerator pedals. Examples of the driving experiments used in this study include straight-line acceleration maneuvers (percent of wide open throttle and gear input are specified for trials that start from a specific initial velocity and end at a predetermined final velocity) and braking in turn maneuvers (traveling at the specified initial velocity, the brake actuator is initiated when the steering wheel hits the mechanical
stop that has been set to allow a certain amount of steering wheel rotation). The raw data from these experiments was filtered and analyzed to provide a baseline to compare with simulation results.

Salaani, Chrstos et al (1997) describes the development of data set necessary to create the model of a 1994 Ford Taurus for the NADSdyna simulation. This data set consists of very precise measurements of part dimensions and orientations for input into the multi-body dynamics formulation. Salaani, Chrstos et al (1997) state that multi-body dynamics is not sufficient for creating an accurate model of a vehicle and all of its components so other subsystems were introduced including engine, power train, wheels, tires, braking, aerodynamic forces, and a terrain database. Initially, Dynamics Analysis and Design System (DADS) from CADSI, a multi-body modeling software package, was used to test the data set that was eventually implemented in NADSdyna.

The results of the validation study described above are presented in Salaani, Heydinger et al (1997). The study confirmed that the simulation package was able to predict the trends of the vehicle dynamics on the whole.

A similar validation study to the one described above is presented in the report by Salaani and Heydinger (2000) in which a 1997 Jeep Cherokee model is validated within NADSdyna using a methodology that is very similar to that described in Garrott et al (1997).
Simulators have been used for training operators of many different devices including tanks, planes, and nuclear reactors, dating back to World War II (Emery et al, 1999). Truck and car driving simulators have been developed to train operators of police vehicles (GE Capital I-Sim, 2002; FAAC, Inc., 2002), tractor-trailers (GE Capital I-Sim, 2002), snowplows (FAAC, Inc., 2002) and a host of other vehicles in both the civilian and military sectors (For more information on simulators see (INRETS, 2002). The report by Emery et al (1999) presents a methodology to validate the effectiveness of simulator training for tractor-trailer drivers. Of particular importance to the current author is the development of training scenarios within a simulator. Training effectiveness is closely related to scenario design (Farmer et al, 1999). A scenario in a simulator is made up of several components. Farmer et al (1999) describe the training scenario as “the environment in which the training activities are to be executed.” Each scenario commonly has a road network or terrain associated with it. Further specifications can be made for other variables that the simulator might be capable of simulating such as traffic (including pedestrians), weather, time of day, and road friction for examples. Scenario designers need to arrange these elements into a scheme that supports the training objectives.

Scenarios developed in the Emery et al (1999) report are intended to meet a goal of training a specific set of skills required by the Professional Truck Driver Institute so that a direct comparison between the trainees with simulator training can be made with those who trained on the real equipment only. While duplicating real world training
activities might be possible within a specific simulator, not all simulators will be capable of simulating all events that trainers want to train. This fact requires scenario designers to account for simulator limitations when developing training scenarios. Without this consideration, training scenario designs can be useless due to the fact that elements (for examples: rain, fog, or traffic) designed into a scenario may not be producible within a given simulator.

Training scenario detail needs to be sufficient for trainees to complete the tasks that are required (Farmer et al, 1999). The amount of detail necessary will vary with training activity (Farmer et al, 1999).

The use of simulator-based measures to evaluate trainee performance is an important part of simulator-based training. Due to the fact that the specification of these measures within the PTDS is a part of the scenario design, a short discussion of this topic is included here. Within simulator training there is a need to be able to assess trainee performance (Farmer et al, 1999). Vreuls and Obermayer (1985) recognize that very advanced simulators often cannot sufficiently measure operator performance (cited in Farmer et al (1999)). A distinction must be made between subjective and objective performance measures. Subjective measures are based on an individual’s opinion (instructor, trainee, etc.) whereas objective measures are based solely on performance (Farmer et al, 1999). Farmer et al (1999) cite sources that have investigated problems encountered when trying to establish simulator-based (objective) performance measures and concludes that both subjective and objective measures are needed in simulator training performance assessment.
CHAPTER 2
HISTORICAL BACKGROUND AND DEVELOPMENT OF THE PENNSYLVANIA TRUCK DRIVING SIMULATOR

Historical Background

The Pennsylvania State Truck Driving Simulator (PTDS), shown in Figure 2.1, is the product of continued work and development that began in 1997 with the undertaking of the 2TS (Truck Training Simulator) project by a consortium of four companies, Moog, Inc., Systems Technology, Inc. (STI), Mack Trucks, and Renault (Delahaye, 1999). The simulator driver station consists of a Mack Trucks CH600 series truck cab mounted on a six-degrees-of-freedom motion platform (Moog, Inc.). STI and Renault developed the two software components that drive the real-time driving simulation, VDANL and SCANeR® II, respectively.

The simulator arrived at the Pennsylvania Transportation Institute (PTI), located at the University Park Campus of the Pennsylvania State University, in July of 1999. At this time the simulator belonged to the members of the consortium and the research staff at PTI were permitted to use the simulator after Mack Trucks had completed initial testing. The members of the consortium officially donated the simulator to the university later that year with the agreement that the members of the consortium could use the simulator.
At the time of installation, the simulator had three visual channels that provided a $130^\circ$ wide by $35^\circ$ high field of view. The software that was running the simulator was SCANeR® II v1.3 and VDANL v6.0 (Delahaye, 1999). The original configuration of the vehicle dynamics model (VDANL) was for a Ford Taurus only. Dr. Moustafa El-Gindy (PTI) and Nicolas Delahaye (Renault) changed this model to simulate a straight truck (Loaded: 10,800kg; Unloaded: 7200kg). The computers used to run the simulator at this time were 3 Dell OptiPlex GXPro computers running dual Pentium Pro 200MHz processors (drivers right side visual generation, the host computer, vehicle dynamics and
Input/Output to the cab) and 2 E-machines Etowercs 333 computers running Cyrix MII
233MHz processors (center and driver’s side front visual generation computers)
(Delahaye and Kemeny, 1999).

Simulator Upgrades

Upgrading the simulator has been a consistent effort since its donation to the university. In an initial extension of the capabilities of the simulator three Dell Dimension 4100 computers were added. These computers operate Pentium III, 800MHz processors and were designated as two rear visual generation computers and a sound generation computer. With the addition of the sound computer came an amplifier that sends sound cues to the cab to simulate road and engine noise. While the installation of the two rear visual generation computers had taken place, the introduction of the necessary projection equipment did not occur until work began on the Pennsylvania Department of Transportation (PENNDOT) Simulator Training Evaluation Program (STEP) in March of 2001. Prior to the initiation of the STEP project, representatives from both STI and Renault were contracted to upgrade the software from their respective companies. These software upgrades represent the most recent changes to the simulator software at the time this document was prepared. At the time of printing the simulator was operating on version 6.0.31 of VDANL and version 1.53 of SCANeR© II.
First Use of the Simulator in a Funded Project

The STEP project represents the first project funded outside the university using the simulator since the original creation and subsequent donation of the simulator to the university. During this one and a half year project many additions to the simulator were necessary to accommodate the needs and desires of PENNDOT management. As a graduate student funded by the STEP project, the current author was responsible for implementing the enhancements described here to the simulator with the assistance of Moustafa El-Gindy, PhD and PTI staff.

Projection upgrades and tuning

The visual projection system of the PTDS contains sets of parameters that can be adjusted to tune the projected images of a simulation. These parameters are stored in the ‘observer.cfg’ file located on the host computer “Marseille” in the directory D:\cats\data\config. Parameters for dimensions, position and orientation of each screen can be adjusted through viewing this file with a text editor. These parameters are “PositionOffset”, “RotationOffset”, “screenTop”, “screenRight”, “screenLeft”, “screenBottom”, and “screenDist”. Figure 2.2 illustrates the physical meaning of many of these parameters. The “Viewpoint” in Figure 2.2 represents the eye of the driver when the motion base has elevated the cabin to its initial level position. All distance specifications (in meters) are made with reference to this point. The projection point is specified as that point on the screen from which a line can be drawn perpendicular to the
plane of the screen and intersect the viewpoint. The parameter “screenDist” is the distance between the projection point and the viewpoint. The dimensions of the screen are then specified as the distance from the projection point to the respective edge (screenTop, screenRight, screenLeft, screenBottom) of the screen. As can be seen in Figure 2.2 the height of the viewable screen area is equal to the sum of the distances screenTop and screenBottom. Similarly the width of the viewable screen area is equal to the sum of screenRight and screenLeft.

**Front projection tuning**

At start of the STEP project the simulator caused many users to experience simulator sickness and the roadways being simulated appeared to be too narrow. This resulted partially because the projection parameter settings did not accurately represent the physical dimensions of the simulator projection equipment. The parameters were tuned to better represent the simulator configuration. To get an accurate measure of the screenTop and screenBottom parameters, a measurement was taken to find the height of the drivers eye above the floor while the simulator was in its neutral position (not on the floor but elevated such that each actuator is at half extension, this corresponds to the position of the simulator when it has been initiated but has not been taken off pause (see Appendix B). This dimension was taken to be 77 inches. While this dimension will vary for each driver, a 6ft individual has been used to establish a working dimension. The net viewing area of the front projection screens is 68” x 92” (W x H). A large framing square, similar to those used by carpenters, was used to find the projection point to
determine screenLeft and screenRight. For the center screen, the screenLeft is approximately 43”, leaving 49” for screenRight. For the screenBottom, the height for the driver’s eye is subtracted from the height of the bottom of the screen. The bottom of the screen is found to be 55.75” from the floor making screenBottom 21.75” (all distance measurements were converted into meters for use in the observer.cfg file).

While these adjustments improved the simulated environment, there were still problems with simulator sickness and narrow roads. To improve the projected image the parameter screenDist was adjusted for the center screen and a slight change in perspective resulted. To measure this change, objects within the projected image of the original configuration were marked on the screen with pieces of tape. Then the parameters were changed and the same image was projected onto the screen. Any movement in the marked items could be attributed to changes in the observer.cfg file. By increasing the screenDist parameter, the desired effect of an enlarged image was attained.

The parameters for the side projection computers were adjusted in a similar manner. The values for screenTop and screenBottom are the same as used for the center screen. While the position of the driver is shifted slightly to the left, an approximation was made so the values for left screen are reversed for the right screen. In other words, screenLeft on the left screen is the same value as screenRight on the right screen. The screenDist parameter for each side screen was increased by the same percentage as the center screen was from its original value.
Adjustments to the RotationOffset parameter were necessary to improve the mesh of the three front projection channels after the tuning described above was completed. While each of these channels functions independently, the ultimate goal of projecting an unbroken scene requires fine-tuning of the RotationOffset once all other parameters are set. The RotationOffset is based on the plane of the center screen. Measures of heading, pitch and roll angles are necessary to describe the RotationOffset. For our purposes only the heading angle is important and the roll and pitch angle should always be zero (i.e. the screens remains vertical with the bottom edge parallel to the floor). The heading angle corresponds to a rotation about a vertical axis that is perpendicular to the plane of the floor. For this reason, a screen in the plane of the center screen has a RotationOffset of zero about all axes or in the case of the rear projection channels a heading rotation of 180 degrees. To accomplish tuning, measurements were taken to assess the angle of the
screens in the room. These measurements were used as a baseline and small changes to the heading rotation were made to achieve the desired projection. It is important to note that this parameter is extremely sensitive and experience has shown resolution of at least one half a degree to be necessary for proper tuning.

**Rear Projection Installation and Tuning**

The installation of two additional screens and projectors for the rear view visual generation channels as mandated by the STEP project deliverables to PENNDOT was completed in June of 2001. Two InFocus LP 260 projectors were selected for the image projection along with two Da-Lite, Perm Wall screens. The tuning of the rear projection is accomplished through the same parameters described above for tuning the front projection with changes in location of viewpoint due to the use of mirrors.

The details of the configuration for the rear projection channels is shown in Figure 2.3. The observer referred to in this figure is the driver. A virtual observer, positioned in relation to the mirror and the observer, is used to establish the projection point that is in turn used to specify screenTop, screenRight, screenLeft, screenBottom as illustrated by Figure 2.3. The parameter PositionOffset represents a 3-dimensional position vector of the virtual observer in the right-handed, orthogonal, Cartesian coordinate frame of the observer’s eye. This reference frame is oriented such that the x-axis is positive in the forward direction of vehicle travel, parallel to the plane of the floor, the y-axis is positive out the drivers side door, parallel to the plane of the floor leaving the z-axis positive in the vertical direction, perpendicular to the floor. The value for
screenDist is found by measuring the distance from the virtual observer to the projection point.

Measurements were taken to establish the geometry of the screen and observer position based on virtual observers for each side of the simulator. To get an accurate impression of what can be seen in a rear view mirror of a truck, photographs were taken while sitting in the driver’s seat of one of PENNDOT’s dump trucks. The projection parameters were varied from the measured distances until the projected image represented the view from within the PENNDOT truck. The values for RotationOffset for the two rear screens were set to be 180 degrees.

Figure 2.3: Rear projection viewpoint and screen configuration
Visual Generation Computer Upgrade

Further enhancements to the PTDS were necessary as the STEP project began in the form of upgrading the two E-machines originally implemented as visual generation computers for the front, driver’s side screen and the center screen. The intense computational demands of running the simulator became nearly impossible for the two computers and made the simulator unreliable. A typical problem the research team faced was failure at the beginning or middle of a simulation in which the visual generation computer would crash. The architecture of the simulator mandates that all programs be initiated at the beginning of a simulation so each time this occurred the simulation had to be restarted. These reoccurring problems were of such severity that research would have been nearly impossible. The E-machines were replaced with a Dell Dimension 4100 running a Pentium III, 900 MHz processor for the front, driver’s side and a Dell Dimension 4100 running a Pentium III, 1000 MHz processor for the center screen. The steps necessary to accomplish these replacements are detailed in Appendix A. The use of the new Dell computers not only made the simulator more reliable but also decreased the time it took to start up the simulator. This had been a concern because the amount of down time a research team experienced between initiating a simulation and being able to begin the experiment (45 – 75secs) needed to be accounted for in experimental procedure. The new computers decreased this lag time to 5 – 10 seconds from when a simulation is initiated.

The projectors used on the three front visual channels are InFocus model LP435Z. These projectors have upgradeable software stored on a flash card within the projector.
Prior to the STEP project each of the three projectors was running a different version of this software and problems such as distorted images arose. The software for the LP435Z is available through the InFocus website at [http://www.infocus.com/service/lp435z/software.asp](http://www.infocus.com/service/lp435z/software.asp). At the time this document was published each of the three projectors had been upgraded to the latest version of the software available, v2.0B. The problems associated with distorted imaging were overcome by this software upgrade. It is necessary to use a special adapter provided by InFocus when updating the software. The projector is connected to the computer’s serial port with this adapter connected and a null modem cable. This adapter is kept in the simulator room in a box marked 435Z. New software versions include instructions on how to flash the memory card within the projector. This process can result in permanent damage to the projector if it is not done properly so it is advised to exercise caution when performing the upgrade.

**Backup Alarm**

A recommendation from training supervisors at PENNDOT that an audible backup alarm be installed for use in backing operations prompted the installation of a alarm that sounds when the gearshift is in the reverse position. This was accomplished with the addition of a micro-switch to the plate designed to hold the micro-switches for detecting the position of the gearshift and a commercially available backup alarm. The alarm is powered by the same power supply used to power the cab and there is no interaction between the I/O computer and the switch used for the backup alarm.
Pressurized Air Connection

Another enhancement made to the simulator in the months prior to the actual training portion of the STEP project (October – November, 2001) was the installation of a compressed air line to the simulator cab. This involved making use of the central compressed air system (previously installed in the building) and the distribution valve body on the cab that distributes compressed air from the reservoirs on a real truck to components such as the breaks, seats, and horn. A flexible air hose was connected to the air supply and the truck cab with brass air hose couplers. The air distribution valve body has outlets for more components than are necessary for the simulator. Brass plugs were inserted into these extra outlets so the system remains pressurized. The addition of compressed air enabled the use of the air horn, the pneumatic cylinders in the seats that allow for vertical seat adjustment, and the parking break light to turn off when the appropriate button is pressed. Within the data collection function of SCAnEr® II (Chapter 4), there is a function for recording when the horn is being used. There is, however, no input signal from the horn to the I/O computer. Therefore horn use cannot be recorded within SCAnEr® II.

CCD Camera Installation

Another addition to the simulator for use in the STEP project was the installation of the closed circuit camera shown in Figure 2.4. This camera is mounted over the right shoulder of the person driving the simulator and sends an image to any device that
accepts a coaxial television cable. For the STEP project, a television was placed in front of the table from which the instructors observed the drivers. The incidence of simulator sickness is more common when riding in the passenger seat of the simulator making the long periods of time required to complete the driving trials for the project difficult for the instructors to endure as a passenger. The camera allowed the instructors to observe the students’ feet and hand motions from outside the cab while rating the different driving exercises.

Figure 2.4: CCD camera installed in the simulator cab
Data Recording

The simulator has the ability to record data, and prior to the STEP project this function had not been utilized. Programming the simulator to record data is described in detail in Chapter 4. There is a large selection of variables that can be recorded during any given simulation, including driver inputs and vehicle responses. Each variable a researcher needs to record is specified separately within a scenario file. For each simulation where data is recorded, a binary file is written to the d:/cats/data/record directory on the host computer “Marseille”. The files are named in accordance with the scenario name and the time and date the simulation was started. The binary data does not follow a traditional format making it necessary to use a binary to ASCII converting program provided by Renault. At the current author’s request, Renault furnished an updated version of this program after the original was found to be defective. The program was originally stored on the host computer but the updated version would not function properly within the Windows NT 4.0 operating system that is running on all the simulator computers except the sound computer. Therefore, the program was placed on the desktop of the sound computer, “Toulouse” (running the Windows 98 operating system), and has proven to function properly. The use of this program is described in more detail in Chapters 3 and 4.
Current Simulator Architecture

The computers implemented on the simulator at the time of publication are summarized in table 2.1 with the programs run by the respective computer during a simulation given in the last column. The computer models are given in the third column with the speed of the Pentium processor in the computer given in parentheses.

Table 2.1: Computers of the Pennsylvania Truck Driving Simulator

<table>
<thead>
<tr>
<th>Computer Name</th>
<th>Function(s)</th>
<th>Computer Model</th>
<th>Programs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marseille</td>
<td>Host</td>
<td>Dell OptiPlex Gxpro</td>
<td>Scenario, Traffic, DataRecord, Mice, Error, Supervisor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(dual 200MHz)</td>
<td></td>
</tr>
<tr>
<td>Lyon</td>
<td>Dynamics, Cabin I/O</td>
<td>Dell OptiPlex Gxpro</td>
<td>VDANL, MOOGStrat, Cabin I/O</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(dual 200MHz)</td>
<td></td>
</tr>
<tr>
<td>Toulouse</td>
<td>Sound</td>
<td>Dell Dimension 4100</td>
<td>Audio mixer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(800MHz)</td>
<td></td>
</tr>
<tr>
<td>Grenoble</td>
<td>Center Visual</td>
<td>Dell Dimension 4100</td>
<td>Visual</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1000MHz)</td>
<td></td>
</tr>
<tr>
<td>Paris</td>
<td>Right Front Visual</td>
<td>Dell Dimension 4100</td>
<td>Visual</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(dual 200MHz)</td>
<td></td>
</tr>
<tr>
<td>Nantes</td>
<td>Left Front Visual</td>
<td>Dell Dimension 4100</td>
<td>Visual</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(900MHz)</td>
<td></td>
</tr>
<tr>
<td>Bordeaux</td>
<td>Right Rear Visual</td>
<td>Dell Dimension 4100</td>
<td>Visual</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(800MHz)</td>
<td></td>
</tr>
<tr>
<td>Avignon</td>
<td>Left Rear Visual</td>
<td>Dell Dimension 4100</td>
<td>Visual</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(800MHz)</td>
<td></td>
</tr>
</tbody>
</table>

A summary of hardware that makes up the simulator is presented in table 2.2.
### Table 2.2: Simulator hardware model numbers

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Manufacturer</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rear Projectors (2)</td>
<td>InFocus</td>
<td>LP260</td>
</tr>
<tr>
<td>Front Projectors (3)</td>
<td>InFocus</td>
<td>LP435Z</td>
</tr>
<tr>
<td>Motion Platform</td>
<td>Moog, Inc</td>
<td>170E122A</td>
</tr>
<tr>
<td>Cab</td>
<td>Mack Trucks</td>
<td>CH600series</td>
</tr>
<tr>
<td>Rear Screens (2)</td>
<td>Da-Lite</td>
<td>Perm Wall (64”H x 84”W viewing area)</td>
</tr>
<tr>
<td>Front Screens (3)</td>
<td>Da-Lite</td>
<td>Fast Fold (68”H x 92”W viewing area)</td>
</tr>
<tr>
<td>Ethernet switch</td>
<td>3-COM</td>
<td>Super Stack II – 3C16611</td>
</tr>
</tbody>
</table>
CHAPTER 3
SIMULATOR VALIDATION

Introduction

Simulator validation can be accomplished in many different ways as illustrated in Chapter 1. This chapter presents a methodology for validating the motion rendering within the cab of a simulator with motion cueing. Motion cueing refers to the movement a driver experiences inside the simulator cab while driving. The implementation of this methodology in the validation of the PTDS is presented. Results are given for straight-line braking and dual lane change driving maneuvers.

Motion Base Control

Motion platforms contribute to real-time driving simulation by exposing the driver to movement during a simulation. The literature discusses a variety of devices employed to provide this type of sensory feedback on simulators. Examples include devices ranging from large speakers mounted under the driver’s seat to simulate vibration or more complex six-degree-of-freedom motion platforms, like the one used in the PTDS that moves the entire simulator cab.

The settings for the motion platform control within the PTDS have not been documented and are the subject of further examination within this chapter. It has been
noted prior to this investigation that changing certain gain settings within the configuration files affects the magnitude of the angular displacements the motion platform produces; however, the motivation for the gain settings is not clear.

The gain settings for the motion control are located in the configuration file on the computer “Marseille” in the directory D:\cats\data\config\platform.cfg. The gain settings for the pitch and roll motion are found in the lines “LowPass1[1.5, clock, -1*mmm*ax] -> [pFromAcc]” and “LowPass1[1.5, clock, nnn*ay] -> [rFromAcc]” respectively. The gain settings are represented here by “mmm” for pitch and “nnn” for roll. The default settings were mmm=0.01 and nnn=0.001.

The translational capabilities of the motion platform only allow for approximately 8 inches of travel in any direction from the neutral position. Operating within these constraints with smooth motion to avoid user discomfort would require that translational motion to simulate linear accelerations only utilize half of this travel. The remaining half of the travel would be needed to stop the cab without reaching the physical limitations of the motion platform. Anything less would result in unacceptably high accelerations. This operating range does not allow for appreciable representation of linear accelerations because of the scaling necessary to keep the platform in the normal operating range.

Within the configuration file mentioned above, it is noted “pitch to simulate x acceleration” and “roll to simulate y acceleration”. These statements make reference to the component of gravitational acceleration acting on the simulator operator when the cab is rotated in either pitch or roll motion that simulates the longitudinal or lateral acceleration respectively. One hypothesis for the motion control gain settings is that the
accuracy of simulated longitudinal and lateral accelerations should be the motivating factor for these settings. An illustration of pitch motion simulating longitudinal acceleration is shown in Figure 3.1.

![Figure 3.1: Pitch motion contributing to longitudinal acceleration](image)

The picture on the left of Figure 3.1 shows the truck cab sitting level with the force due to gravity, $F_g$, acting on a body placed in the driver's seat. The directions of $Z_g$ and $X_g$ show the global reference frame, which is the same as the body fixed reference frame for the cab in the picture on the left. The picture on the right of Figure 3.1 shows the truck cab pitched at an angle theta ($\theta$). The force of gravity ($F_g$) acting on the mass in this picture has been resolved into components $F_{pg}$ and $F_{pl}$ in the body fixed reference frame of the cab, now shown by the orientation of $Z_c$ and $X_c$. $F_{pl}$ represents the longitudinal force on the body due to the pitch angle $\theta$. $F_{pg}$ is the perceived gravitational force for the body moving with the frame of the cab. The equation for the simulated
longitudinal force, $F_{pl}$, as a function of the pitch angle, $\theta$, is given in equation 3.1 in terms of the variables described above.

$$F_{pl} = F_g \times \sin(\theta)$$  \hspace{1cm} (3.1)

A result of using angular positioning to simulate linear accelerations for a vehicle is that the normal rotations of the sprung mass with respect to the unsprung mass during driving maneuvers in the simulated vehicle are not directly considered in the motion control. While pitch and roll in a real vehicle result from the linear accelerations of the sprung mass center of gravity, the motion cueing control as described above would be focused on recreating the sprung mass linear accelerations, not the sprung mass rotations. A second hypothesis for the motion control gain settings is that the angular motions of the simulated vehicle are more important to the motion cueing than the linear accelerations recreated by the simulator cab rotations. Thus the motion control settings would be based on the sprung mass rotations and the linear accelerations would be neglected.

**Validation Methodology**

The procedure outlined below was developed to provide experimental support for the gain settings of the motion cueing within the PTDS and is represented in the diagram presented in Figure 3.2.
This methodology is based on comparing the simulator cab motion during a simulation with the motion a vehicle of the same dimensions undergoes during the same driving event. To accomplish this a Dynamic Measurement Unit (DMU) was installed in the cab of the PTDS. The DMU used here is a device capable of measuring three mutually orthogonal linear accelerations as well as angular rates about each axis. During a simulation both driver inputs and measurements from the DMU are collected. After the simulation has been completed in the simulator, the driver inputs recorded during the simulator trial are used to recreate the driving maneuver within a stand-alone simulation package, VDANL. The measurements gathered from the DMU are then compared to the predictions from the VDANL simulations through data analysis. Relationships between the motion of the cab and the motion of the sprung mass of the actual vehicle are found based on these comparisons. The results for different gain settings are then used to tune the motion cueing parameters. The results for different settings are used to tune the motion cueing parameters so that the motion in the cab during a simulation best represents that of the actual vehicle’s sprung mass for similar driving inputs.
Equipment Configuration

The simulator was equipped with a Dynamic Measurement Unit (DMU) prior to the investigation, as mentioned above. The DMU used for this study was a Crossbow Technology Inc., DMU-VGX. This instrument communicates with a computer through a
serial connection and uses an external power source of 12 volts. Version 1.8 of Accel-View, the software that runs on the computer connected to the DMU, was used for this study. The DMU was mounted with screws to the center console of the simulator cab as shown in Figure 3.3. The cigarette lighter in the dashboard of the simulator was used to power the DMU. An extended data cable was assembled to permit using a computer outside of the simulator cab for collecting data. Measurements from the DMU are logged into a file at approximately 100Hz.

Figure 3.3: DMU mounted in the simulator’s cab
The data collected for this study included data taken from the DMU and driver inputs taken from the simulator. Driver inputs were recorded by writing specific scenarios for each of the two types of driving maneuvers conducted for this research, namely straight-line breaking and the double lane change. In this way, two separate devices were recording data for a single experiment. The DMU software (commercial version was used and the only adjustments made were those necessary to configure the software for use with the DMU-VGX) was written to record elapsed time at each interval. Similarly, the simulator was programmed to record the elapsed time at each recording step. Thus, these two time measures were made on two separate computers running two separate software packages and neither had a constant duration between steps. In order to correlate the events of driver input with the motion cueing responses it was necessary to synchronize these two time measurements.

**Timer Synchronization**

The initiation of a test run on the DMU can be accomplished by highlighting but not selecting the “start” button in the Graphical User Interface (GUI) of the Accel-View software and pressing the spacebar on the computer running the software. Similarly, a simulation in the PTDS is initiated by un-pausing the simulation by pressing the spacebar while the window for the traffic program WCarto is selected. In both devices pressing the spacebar on the respective computer a second time halts a test run. The data recording for each device, and thus the timers, start and stop with these keystrokes.
The synchronization of the two timers was accomplished by constructing a keyboard capable of sending information to two computers simultaneously. The wiring for this keyboard included the standard wiring for a keyboard with data lines of an additional cable soldered to those of the original cable. The power wire of the additional cable was not connected to avoid placing twice the necessary voltage across the logic of the keyboard that would most likely result in permanent damage to the keyboard. It was discovered that when the two computers are connected to the keyboard a conflict in the clocking signal resulted after a short time and the computer connected to the added keyboard cable did not receive the keyboard output. This was overcome by placing a switch in line with the clocking wire of the added keyboard cable. The switch was wired so that the keyboard only receives the second clocking signal when the switch is pressed. For this reason the computer plugged into the added cable only registered a keystroke if the switch was pressed. If a conflict in clock signals occurred at any time while using the keyboard, simply releasing the switch and then repressing it allowed operation to continue.

The underside of this switch and the wires coming into the dual output keyboard are shown in Figure 3.4. For the purpose of data collection only one keystroke was required at any one time and the keyboard functioned in this capacity for all experiments described in this document.

It was noted that the typical variation in the simulator timer and the DMU timer when using the keyboard shown in Figure 3.4 was on the order of ± 0.1 s for trials lasting 50 s. The accuracy of the simulator timer is noted to be on the order of 0.1 s within the
simulator help pages. The accuracy of the DMU timer was not identified in the literature. The clock synchronization was assumed to be accurate enough to represent the time recorded by simulator timer with the time recorded by the DMU timer.

![Dual output keyboard for synchronizing the DMU and the PTDS](image)

Figure 3.4: Dual output keyboard for synchronizing the DMU and the PTDS

The straight-line braking and double lane change driving maneuvers were used as driver inputs because the primary vehicle response is planar. With straight-line braking a vehicle experiences pitch motion and longitudinal acceleration while the double lane change primarily results in roll and lateral acceleration.
The straight-line braking maneuvers were conducted with a mechanical stop installed in the simulator as shown in Figure 3.5. This allowed drivers to apply consistent braking of less than one hundred percent. The coupling device is mounted to two pieces of cable. One of these cables was fastened to the brake pedal and the other was secured to a bracket supporting the steering column. The coupler between these two cables allows the user to vary the length of the assembly at full extension by turning the centerpiece that has been threaded for the two eyebolts. This coupling device was purchased at a local hardware store and is easily replaced.

Braking inputs of 68% and 36% were used for this study. The upper value was chosen because trials showed that braking events with braking effort that exceeded this level resulted in vehicle behavior that corresponded to a locking of the rear and/or front wheels. Six settings of the gain for pitch described above were used for the 68% trials while 5 gain settings were used for the 36% trials.
The simulator was equipped with markers shown in Figure 3.6. These markers allowed the driver to produce consistent steering inputs over several trials. This was important because investigation targeted the effects of variations in gain settings with all other aspects held constant. Tape was placed on the steering wheel so the driver could identify the steering input during each trial.
The driving experiments for both straight-line braking and the double lane change maneuvers were performed on a section of road that was both straight and level. This simplified input into the stand-alone vehicle dynamics simulation package. Programming the simulator to measure these parameters and then traversing the route verified these road characteristics. This is important to ensure cab motion is based exclusively on driver input and not terrain variations.
The monitor to the computer, Lyon, was placed next to the computer running the DMU for the experiments completed in this study. This allowed the researcher running the simulator and DMU to view a digital readout of all driver inputs and vehicle speed through the active window for “CabinIO_2TS.exe”. This is the program that was originally installed to take input from the driver controls and send information out to the cab when the simulator was developed under the 2TS project mentioned in the previous chapter.

**Experimental Procedure**

Prior to starting the braking tests the brake stop was adjusted for the desired braking effort. To accomplish this the driver was positioned in the simulator and the program CabinIO_2TS.exe was started. The driver then pressed the brake until it engaged with the stop while the researcher running the computer observed the braking input readout in the CabinIO_2TS.exe window. Adjustments to the length of the cable were made and the process was repeated until the desired braking effort was attained in successive trials.

The simulation was then initiated by pressing “Launch All” in the simulator supervisor menu and the DMU was set to show a graph of angular rates. It was important for the researcher running the simulator and DMU to make certain the data recording function for the simulator was activated at this stage. Prior to opening the scenario “Andy_brake” in WCarto in the simulator, the DMU was zeroed by clicking the “DMU zeroing” button in the graph GUI. The scenario was opened and the dual output
keyboard was connected to the laptop computer running the DMU and the host computer of the simulator, Marseille. The original keyboard output was always plugged into the laptop because of problems with using the added cable in that computer. With all systems initiated, the start button in the Accel-View was highlighted using the “tab” key and the window for WCarto, now open in Marseille, was selected. The researcher then needed to check the window for the CabinIO_2TS for any residual braking input. Residual braking input is a manifestation of the software configuration of the optical encoder used to detect braking input. This problem is discussed in more detail in the final chapter.

Once all steps described above had been completed, the driving portion of the test began. The researcher operating the computers then depressed the clocking switch mounted on the dual output keyboard and pressed the spacebar. This started the driving simulation and the data recording of the DMU simultaneously. The driver drove the simulator down the road and the researcher at the computers relayed the vehicle speed in the CabinIO_2TS window to the driver vocally as the speed approached the desired speed of 65mph. When the simulated vehicle was traveling at a speed of 65mph the driver applied the brake as fast as possible in order to achieve the desired step input braking. The researcher at the computers monitored the braking input to ensure the desired level was being applied. If for any reason the desired braking effort was not reached during a trial, that trial was repeated. The braking effort was maintained until the vehicle had stopped and the cab had stopped moving, at which time the researcher pressed the switch
on the keyboard and again pressed the spacebar. This stopped both the data collection of
the DMU and the simulation.

After each trial the researcher stopped the Accel-View graphical window and
pressed “kill All” in the Supervisor GUI of the simulator. It was necessary to stop the
DMU program to reset the clock because while renaming the data file stops data from
being written to that file, the new file would start at a time step one greater than the last
recorded in the original file. Data analysis was simplified by having all data sets start
with a time of zero.

The procedure followed for the double lane change maneuvers was similar to that
described above. The simulations were initiated in the same manner as for the braking
maneuvers except that a different scenario was opened. This scenario programmed the
simulator to record different parameters than were needed in the braking trials. Both
scenarios took place on the same road.

The driving maneuver consisted of the driver proceeding down the road with the
researcher at the computers dictating speed read from the CabinIO_2TS window. At a
speed of 50mph the driver steered the vehicle into the opposite lane and immediately
back into the original lane. All trials started in the right hand lane and the driver
attempted to have a consistent steering input over trials. Five different values of the gain
setting for roll were used to characterize the effects this setting has on cab motion.
**Data Recording During Trials**

The data files written by the DMU were text files with values for 9 different measures given in columns. Each row corresponded to data taken at the time step, specified in the first column, given in seconds. Columns 2, 3, and 4 were angular velocities about the x, y, and z-axes respectively. Columns 5, 6, and 7 were linear accelerations along each axis of the DMU given in multiples of the acceleration due to gravity or G's. The DMU was oriented in the cab such that the positive x-axis was toward the front of the truck, the y-axis was positive out the passenger side door leaving the z-axis positive downward. Column 8 was the temperature in Celsius degrees and column 9 was a timer not important to the current study.

The data recorded by the simulator was originally written into a binary file that was converted to text using the binary to ASCII converting program mentioned in Chapter 2. The text files created by this program were similar to that of the DMU output, with data for specific measures in columns and data for each time step given in each row. For the braking trials these files consisted of six columns. Each row corresponded to a record integer (starting from zero and continuously increasing throughout the file) given in column 1. The remaining columns were configured so column 2 was time (s), column 3 was braking input on a scale from 0 – 1, column 4 was vehicle speed (kph), column 5 was distance from a parked vehicle (m), and column 6 was the steering wheel input.

The data recorded by the simulator for the lane change maneuvers included eight columns. Similar to the braking trials, column 1 represented a record number. Column 2 was time (s), column 3 was vehicle speed (kph), column 4 was steering wheel input in
multiples of full revolutions (i.e. 1 = 360 degrees), column 5 was distance from a parked vehicle (m), column 6 was the lateral acceleration (m/s²), column 7 was road slope (change in height (m) for traversing 100m) and column eight was the distance from the right side of the vehicle to the right side of the road (m).

**VDANL Simulations**

Simulations were run in VDANL following the simulator trials described above. The motivation for the implementation of the stand-alone simulation package is two fold. VDANL drives the real-time simulation of the simulator and therefore all things related to vehicle dynamics in the simulator, including the cab motion. At first glance, a comparison could be made between the predictions from a real-time simulation and the cab motions. While this is possible, it affords the researcher limited access to the dynamic output parameters calculated by the dynamics model due to the architecture of the software running the simulator. This is described in more detail in the next chapter. Secondly, the difference between the real-time simulation and the stand-alone simulation package is not well documented. It is reasonable to expect that the integration time step used by the stand-alone simulation is significantly smaller than that used in the real-time simulation. This is supported by the fact that simulations run on the stand-alone simulation package are not completed in real-time. The stand-alone simulation should result in a more complete and more accurate solution for vehicle motion for this reason.

The input to the VDANL simulations was set to mirror the input to the simulator for each trial. The input to the simulator was recorded for each trial as described above.
Unit conversions were necessary to achieve the correct input into VDANL. The data recorded by the simulator can be converted using the calibration settings in the CabinIO.exe program and basic unit conversions. The calibration values in CabinIO are set to convert raw input data into a form that can be used by VDANL in the simulation. The raw input takes the form of counts on optical encoders for the steering wheel and pedal inputs and bit numbers for gear inputs. A screen capture of the calibration window is shown in Figure 3.7.

![Calibration Values](image)

Figure 3.7: Calibration window for CabinIO.exe

The brake pedal in the simulator is calibrated such that 19 clicks on the optical encoder represents full braking ("counts at FS" = counts at full scale, see Figure 3.7).
Braking is input into the simulator on 0 – 1 scale with zero being no brake and 1 being full braking such that only increments of 1/19 can be achieved. The braking input into VDANL is in pounds. The conversion is a linear scale of the brake input by the parameter NL.BRAKTMAX found on the computer Lyon in the file truck_PTI_loaded_NonLin.vpf found in the directory C:\modelVDANL\Parameter_Files\truck_PTI_loaded\. This parameter represents the maximum force applied to the pedal by the driver. The braking for in the simulated vehicle is given in equation 3.2:

\[ F_b = \frac{1}{19} \times (counts) \times NL.BRAKTMAX \]  

(3.2)

Where: 
Fb = the applied braking force
Counts = the number of optical encoder counts
NL.BRAKTMAX = 50 lbs (for truck_PTI_loaded)

The data output from VDANL can be selected by the user and is specified in the parameter file on Lyon in C:\modelVDANL\Parameter_Files\xxx\xxx_Param.vpf; where “xxx” specifies the vehicle model. The vehicle model used in this study was truck_PTI_Loaded. The VDANL output for the braking simulations consisted of brake input (lbs), center of gravity (CG), longitudinal and lateral accelerations (ft/s^2), CG longitudinal velocity (ft/s), sprung mass (SM) longitudinal and lateral accelerations (ft/s^2), pitch angle (rad), pitch velocity (rad/s), pitch acceleration (rad/s^2), time (s), and longitudinal position (ft). The VDANL output for the lane change simulations consisted of steering input (rad), center of gravity (CG), longitudinal and lateral velocities (ft/s), sprung mass (SM) longitudinal and lateral accelerations (ft/s^2), roll angle (rad), roll
velocity (rad/s), roll acceleration (rad/s^2), time (s), and longitudinal and lateral positions (ft). The data was saved as MAT files (extension - *.mat) directly by VDANL.

Results

The data analysis for this study was completed using MATLAB. An individual program was written to analyze the braking and lane change data sets. The program used to complete the data analysis for the lane change maneuvers is given in Appendix D. The results for the braking and lane change maneuvers are presented separately below.

Straight-Line Braking

An important step in analyzing the data for this study was ensuring that the input to VDANL and the input to the simulator was the same. To do this plots of the braking input were created for both the VDANL and simulator trials. The time scale for the simulator trials was shifted so that braking input profiles for the VDANL simulations and the simulator trials occurred at the same time. Examples of the braking profiles for two simulator trials and the corresponding VDANL simulations are illustrated in the plots of Figure 3.8.
Figure 3.8: Select braking input profiles for VDANL simulations and simulator trials for 36% braking effort (top) and 68% braking effort (bottom)

A comparison of the velocity profile in the VDANL and the simulator was conducted for each set of braking data after the time scales had been fixed based on the braking input. As can be seen in Figure 3.9, there are negligible differences in vehicle speed as recorded by VDANL and the simulator.

It should be noted that there are “holes” in some of the simulator data output presented below. An example of this appears in the top plot of Figure 3.9 where the red curve drops to zero for a short time and then returns to follow the expected profile. Time
periods where data recorded by the simulator has holes like the one described here represent the simulated vehicle crossing from one road section to the next. The trials for this study were conducted on a stretch of road in the simulator where two sections are joined together. The vehicle behavior does not reflect the data recorded for these sections and therefore will not impact the DMU measurements.

Figure 3.9: Select velocity profiles for VDANL simulations and simulator trials for 36% braking effort (top) and 68% braking effort (bottom)

The similarities in the vehicle speed and braking input for each trial, coupled with the timer synchronization discussed above allowed vehicle responses predicted by
VDANL to be compared directly to the motion response in the simulator cab. The pitch angles achieved by the sprung mass in the VDANL simulations and the pitch angles of the simulator cab are shown in Figure 3.10 and Figure 3.11. The curves labeled “Measured” within these figures are the pitch angles derived by integrating the angular velocity data collected by the DMU. The measured curves were shifted so that the starting angle was approximately zero to allow for more direct comparisons. The shift in the original data represents a cab rotation that occurred prior to the braking event and is therefore not important to this study.
Figure 3.10: Pitch angles predicted by VDANL and measured in the simulator cab for 36% braking input.

The pitch velocities of the sprung mass in the VDANL simulations and the pitch velocities of the simulator cab for the various gain settings investigated in this study are shown in the plots of Figure 3.12 and Figure 3.13.
Figure 3.11: Pitch angles predicted by VDANL and measured in the simulator cab for 68% braking input
Figure 3.12: Pitch velocities predicted by VDANL and measured in the simulator cab for 36% braking input.
Figure 3.13: Pitch velocities predicted by VDANL and measured in the simulator cab for 36% braking input

The longitudinal accelerations of the sprung mass in the VDANL simulations and the longitudinal accelerations measured in the simulator cab for each trial are presented in Figure 3.14 and Figure 3.15.
Figure 3.14: Longitudinal accelerations predicted by VDANL and measured in the simulator cab for 36% braking input
Double Lane Change Maneuver

The same input into the simulator and the VDANL simulations was necessary for the same reasons discussed for the braking trials. The similarity in the steering input for the lane change maneuvers in the simulator and VDANL can be verified graphically.
through the plots of Figure 3.16. The velocity profiles for these maneuvers when performed in the simulator and simulated in VDANL are shown in Figure 3.17.

Figure 3.16: Steering input for lane change maneuver trials
The primary response rendered in the simulator cab for the lane change maneuvers was roll. The roll angles of the simulator cab and sprung mass in the VDANL simulations for trials with different gain settings in the simulator are shown in the plots of Figure 3.18. The roll velocities for these trials are shown in Figure 3.19.
Figure 3.18: Roll angle predicted by VDANL and measured in the simulator cab.
The roll angle of the simulator cab causes the simulator driver to experience lateral acceleration in the same way the pitch motion causes the driver to experience longitudinal acceleration. The lateral accelerations measured by the DMU for trials with different gain settings and for the sprung mass in the VDANL predictions are shown in Figure 3.20.
Discussion

The plots presented above detail many findings. Of primary importance is the support for further implementation of the methodology implemented herein. The ability to duplicate the driver input from the simulator trials in the VDANL simulations is demonstrated in Figure 3.8 and Figure 3.9 for the braking experiments and in Figure 3.16 and Figure 3.17 for the lane change experiments.
Inspection of all plots depicting vehicle and cab response show delayed cab response. This behavior is expected in a simulator due to transport and computation delays. These delays, however, do not show a consistent trend over trials as might be expected. Part of this deviation is attributed to imperfections in the timer synchronization described above. While every effort was made to ensure the timers were synchronized, the accuracy of the timers coupled with deviations in the differences between the two timers from trial to trial do not allow for an accurate assessment of the delay displayed by the simulator because this delay is typically on the order of $10^{-1}$ s.

**Straight-Line Braking**

The effect of the gain settings for pitch motion within the motion base control file is well defined by the plots in Figure 3.10 through Figure 3.13. It is clear that increasing the gain results in greater angular displacements for the same driving input up to the point where the motion base is constrained by the control algorithm. The constraint of the cab response is represented by the plateau in angular displacement in the plots for 0.025 and 0.03 gain settings in Figure 3.11. Similarly, the angular velocities increase with increases in the gain setting to a predefined limit. This is illustrated by the maximum velocity occurring at approximately the same value in the plots for 0.025 and 0.03 gain settings in Figure 3.13. The measured values for longitudinal acceleration shown in Figure 3.14 and Figure 3.15 reflect the limitations of the pitch motion in rendering longitudinal acceleration. The measured accelerations are directly related to angular displacement and therefore reflect equation 3.1.
There appears to be a tradeoff between increasing the simulated longitudinal acceleration and maintaining a reasonable recovery time for the straight-line braking maneuver. The recovery time can be defined as the time it takes the simulator cab to return to a neutral position after the vehicle has stopped. Close inspection of the pitch angle curves in Figure 3.10 and Figure 3.11 show an increase in the recovery time for increases in the gain setting. The result of increased recovery time is prolonged, simulated longitudinal acceleration and is therefore a detriment to the validity of the simulated environment. It is also important to recognize that while the acceleration due to pitch motion is always a fraction of the acceleration predicted by the VDANL simulations, there should be a correlation between the acceleration in the cab and the predicted acceleration. Choosing a gain setting that is too high results in a consistent acceleration in the simulator cab for significantly different predicted accelerations. This is illustrated by the 0.03 gain setting in that the maximum pitch angle for the 36% and 68% braking input is nearly the same and therefore the longitudinal acceleration in the cab is also comparable.

From this discussion it is apparent that gain settings for the pitch motion in the PTDS need to be selected based on the longitudinal acceleration achieved for a given driving input, the trends for variations in that driving input and the recovery time. After reviewing the results for the braking maneuvers presented above it is the current author’s recommendation that the pitch gain in the PTDS be set at 0.01.
Double Lane Change Maneuver

The effect of the gain settings for roll motion within the motion base control file is well defined by the plots in Figure 3.18, Figure 3.19 and Figure 3.20. While it is not as direct a comparison due to variations in the steering input, similar to the pitch motion discussed above the roll angle and velocity each increase with increases in the gain setting. Again the deficiency of simulating linear accelerations using angular displacements is shown in the plots of Figure 3.20. Unlike the longitudinal acceleration, the lateral acceleration rendered by the motion platform is not clearly representing the predicted acceleration. For this reason the continuity of the roll motion should be used as the deciding factor for the gain setting used in the PTDS without serious consideration for the accelerations achieved.

Based on this discussion this author recommends setting the roll gain to 0.001.
CHAPTER 4

SCENARIO DEVELOPMENT WITHIN THE PENNSYLVANIA TRUCK DRIVING SIMULATOR

Introduction

Most experimental work performed using the PTDS requires the ability to program and develop driving scenarios. Scenarios represent a way for researchers to interact with and control the simulated environment. Traffic events, terrain specifications and data collection are all accomplished through programming scenarios. Therefore training exercises are developed through scenarios. The lack of documentation on the current architecture and operation of the PTDS available to researchers has required the current author to expend considerable effort to solve problems associated with simulator operation and programming. The information presented here represents information the current author has acquired through personal experience with the software described herein and personal correspondence with the engineers who have worked on the simulator since its original development.

Real-time Simulation Components Within the PTDS

A large number of individual components are needed to run a simulation with an interactive driver in the PTDS. Throughout the development of the simulator, new software versions have been introduced to increase performance. The implementation of
new software versions has not always been accompanied by a complete removal of old versions. The directories of files used to run simulations at the time of publication are given for each of the simulator computers in Table 4.1. This information is not intended to represent all data and configurations necessary to conduct a simulation, but instead to provide a basis for navigating through the programs and data used by the simulator. It should be recognized that not all files or programs run on a specific computer are stored in that computer. The sound computer, “Toulouse”, runs programs from the D:\cats\bin directory on the host computer “Marseille”, for example. Furthermore, all the sound clips used in a simulation are found in the D:\cats\data\sound directory on Marseille.

Table 4.1: Directories in the PTDS computers used for simulations

<table>
<thead>
<tr>
<th>Computer</th>
<th>Directory</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marseille (host)</td>
<td>D:\cats\data</td>
<td>Data files</td>
</tr>
<tr>
<td></td>
<td>D:\cats\bin</td>
<td>Programs and executables</td>
</tr>
<tr>
<td>Lyon</td>
<td>C:\modelVDANL\CabinIO_2TS</td>
<td>Cab Input/Output executable</td>
</tr>
<tr>
<td></td>
<td>C:\modelVDANL\Parameter_Files</td>
<td>Vehicle Dynamics data</td>
</tr>
<tr>
<td>Toulouse</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Paris</td>
<td>C:\Cats</td>
<td>Data files</td>
</tr>
<tr>
<td>Grenoble</td>
<td>C:\Cats</td>
<td>Data files</td>
</tr>
<tr>
<td>Avignon</td>
<td>C:\Cats</td>
<td>Data files</td>
</tr>
<tr>
<td>Bordeaux</td>
<td>C:\Cats</td>
<td>Data files</td>
</tr>
<tr>
<td>Nantes</td>
<td>C:\Cats</td>
<td>Data files</td>
</tr>
</tbody>
</table>

**Configuration Hierarchy Within SCANeR® II**

SCANeR® II is the software package that currently controls all programs used in an interactive simulation. The program, “Supervisor”, has been designed to allow an operator to interact with all of these programs during a simulation. The Graphical User
Interface (GUI) for this program is shown in Figure 4.1. A shortcut to this program is located on the desktop of the computer Marseille. The Supervisor is the program used to start a simulation in the simulator. Appendix B contains instructions for starting the simulator and includes a description of using the Supervisor in that capacity.

Figure 4.1: Supervisor GUI
Developing Scenarios in Mice

The Supervisor can be used in a capacity other than running simulations because it contains links to SCANeR® II programs not used during simulations. The most important of these is “Mice”. Mice is the program in SCANeR® II that allows a researcher to develop scenarios. This program can be started in two ways, by double clicking on the square to the right of the name “Mice” in the Supervisor, or by running it from D:\cats\bin directory on Marseille. The program “Daemon” must be running prior to running Mice. Mice can only be run on Marseille.

Within Mice the user will be working directly with scenario files that are stored in D:\cats\data\scenario on Marseille. These files have the extension *.sc and can be viewed in text format with a text editor. The information contained in these files is in a computer language that is difficult to understand in its text form so it is recommended that work be done on them through Mice.

To illustrate the functions of Mice, a description of creating a scenario from the beginning is presented here. Modifying existing scenarios is a function of using the basic navigation tools described below. To create a new scenario in Mice once the program is started, select “File” from the menu bar at the top of the window and choose “New”. A window will appear with options for terrain profiles. Each of these profiles has set roadways on set topography that can represent the setting for a scenario. The “Mountain” terrain contains roads on and around mountains. The “Industrial” terrain has roads surrounding an industrial park. “Simulia” represents the most diverse roads in any terrain profile with towns, snow covered roads, a bridge and tunnels. The specific characteristics
of the different terrain profiles can be investigated through driving a scenario previously created on that terrain.

**Traffic Generation**

Once the terrain has been selected, the traffic of the scenario should be inserted. The traffic must include the interactive vehicle. The interactive vehicle is the vehicle that will be controlled through the control inputs of the simulator cab. Choosing “View” from the menu bar and selecting “Traffic” allows the user to see a map of the terrain file. The user can then insert a vehicle into the scenario by right clicking on the window with the map of the terrain file and choosing “Insert vehicle” within the popup window. When this is done a window pops up with options for specifying characteristics of the new vehicle. A snapshot of this window configured for an interactive vehicle using the “truck_PTI_loaded” vehicle model is shown in Figure 4.2. The designation for “Model” in this window can take values for vehicles that are either interactive or autonomous. At the time this document was prepared, three models represented stable vehicle configurations that could be used as interactive vehicles, “truck_PTI_loaded”, “Taurus”, and “truck_PTI_loaded_snow”. The other models cannot be used as interactive vehicles because there are not dynamic models associated with them. In the event that the existing vehicles are not adequate for research, new interactive vehicles can be created. A detailed description of the steps for creating a new vehicle within the PTDS is given in Appendix C. The general procedure followed to accomplish this involves copying the files that define an existing vehicle and renaming them to represent the new vehicle.
Changes to the dynamic model can be done to the copied files without fear of damaging the existing files or omitting necessary components.

Figure 4.2: Vehicle configuration window within the traffic generation portion of scenario development

The section of this window labeled “Initial Status” is used for specifying the characteristics of a vehicle at the start of a simulation. An activated vehicle assumes the behavior specified within the other fields of this window, where a vehicle that is not activated will remain parked from the beginning of the scenario. This value can be changed during a simulation through programming in Mice. This is described in more
The “Visible” specification for a vehicle indicates the graphics for that vehicle will be displayed from the beginning of the scenario. A researcher may want to have a vehicle invisible in a scenario if the vehicle is being used as a marker for distance measurements, for example. In this case the vehicle would be invisible to the driver of the simulator and therefore not be a distraction while the driver is completing the driving tasks.

Positioning a vehicle within the map of the terrain is achieved using the following controls within the traffic window:

- Zoom - press the roller on the mouse and move the mouse up to zoom out and down to zoom in
- Shift image - press control, right click and move the mouse up or down to shift the image up or down respectively
- Move vehicle - position arrow over the orange dot representing the vehicle, press control, left click and move to desired position. For a simulation, the vehicle must be placed on one of the roads.

The other options contained in this window are used within the scenario for functions that primarily deal with the interaction of vehicles. The specific functions in SCANeR® II are defined in a help function described in more detail below.

Autonomous vehicles are created in the same way as an interactive vehicle with the distinction of autonomous made within the vehicle properties window. The properties of a given vehicle can be viewed by double left clicking on the vehicle within
the traffic window of Mice. In this way a researcher can make adjustments to vehicle
 specifications while developing scenarios.

The details of the traffic configurations for a given scenario are contained in a traffic file that is automatically generated within the D:\cats\data\traffic directory on Marseille when the scenario is saved in Mice. The name of the file is the same as that of the scenario with the extension *.trf instead of *.sc. These files can be viewed using a text editor but using Mice to make changes to the vehicles will help avoid making errors that can make the scenario not function.

The intimate relationship of the scenario files and traffic files during a simulation is apparent through the procedure for running the simulator. As described in Appendix B, a simulation is initiated within the program Wcarto by selecting a traffic file. The scenario associated with this file is run automatically. It should be noted that most training scenarios are dependent on the configuration of the traffic at the beginning of the scenario and within Wcarto there is an option to “Save” in the “File” pull down menu. If a traffic file is saved at the end of running the scenario, the vehicles will start in that position the next time the scenario is run. This could be a catastrophic result because the scripting within a scenario often hinges on vehicles reaching certain locations and the starting position of each vehicle ultimately determines the sections of road they will be traveling. For this reason a backup of all scenario and traffic files used in a training or research capacity should be created and stored in a safe place.
Developing Other Scenario Elements

Scenarios in the PTDS contain traffic elements and a script with user defined actions for controlling the simulated environment. The script has a hierarchical structure with the scenario being divided into tasks, tasks divided into rules, with the rules defined by conditions and actions. Figure 4.3 shows a screen shot of the Mice GUI with a scenario being edited. The first task has been named “Start” and the first rule has been named “Start_2nd_group”. To begin programming the first task of a scenario, a user merely right clicks on the small green box at the top of the main scenario window in Mice. A popup window will prompt the user to insert a task, rule, variable or list of files. The use of the latter two will not be discussed further.

A set of rules can be organized into a task to facilitate programming. Each rule is framed in the form of an IF-THEN statement. Each IF statement is referred to as a condition. Right clicking on the green rule box and selecting “Insert Condition” from the popup window inserts a condition. There exist a multitude of conditional statements for a user to choose from including “isAtPosition” used in the rule shown in Figure 4.3.
The conditions are commonsensical and this one corresponds to the specified vehicle reaching a specific location in the terrain file. The vehicles that conditions and actions apply to are specified by double clicking on the green box of the statement in question and using the pull down menus in the pop up window to select the correct vehicle. The vehicles are referred to by the name given to them in the traffic development window described above. The position for the “isAtPosition” argument is specified through the same popup window used for specifying the vehicle. A function called the “Map Tool” allows the user to place a crosshair on a specific location within the map of the scenario to specify the location desired. Particular attention should be
paid to the value of the conditional statements. Values vary from versions of “is true/false”, “has been true/false”, “becomes true/false”, and “has been true/false for”. These values are also selected in the popup window used for defining the other parameters of the argument.

The actions are inserted in the same manner as the conditions. The actions represent the events that will happen in the event the preceding condition is satisfied. A list of actions is presented to the user in a popup window. A complete description of each action can be found by inserting the action into the scenario, double left clicking on the green icon within the Mice GUI, left clicking on the question mark in the upper right hand corner and then clicking on the popup window or pressing “F1”. A help window with the description of the action will be opened. This window can be used to navigate through the pages of the help function once it has been opened in the manner described here.

**Recording Data With the Simulator**

The ability to record data from a simulation is critical to many forms of research and training with the simulator. In the PTDS this is accomplished through programming within a scenario. Prior to the STEP project no data had been collected with the PTDS since the most recent software upgrade. This was apparent because of certain missing folders and malfunctioning programs at the start of that project. These included the problems with the binary to ASCII converting program described in Chapter 2 and the lack of the D:\cats\data\record directory on Marseille. This directory is where the binary
data is recorded during a simulation. Without this directory there is no possibility of
recording data and it should never be removed from Marseille.

The first step to recording data with the simulator is to create a rule with a
condition that specifies when the data should begin being recorded. The condition should
be configured to suit the researchers needs and it will most likely reflect the type of work
being conducted. Training exercises often require data be recorded for a portion of the
entire scenario, for example in the scenarios described below. The scenario created for
collecting data presented in Chapter 3 lane change maneuvers is shown open in Mice in
Figure 4.4.

Figure 4.4: Scenario for recording data open in Mice showing necessary context
Once the condition has been set, a researcher needs to use the function “connectChannel” in the first action of the rule. The value for this operation should be set to (ALL_CHANNELS, ALL_DEVICES, ON). The next action should be set to the function “exportChannel”. The value for this operation represents a parameter or combination of parameters to be recorded. This parameter can take a number of different values and entering the help pages for functions accesses complete descriptions of these parameters. This is accomplished in the same manner as used to enter help for actions as was described above. Figure 4.4 shows several possibilities for the value of exportChannel. When defining the export channel it is important to remember to change the number associated with each successive channel. The channels in Figure 4.4 for example range from 0 to 6. These values ultimately determine the column the data for the parameters appears in within the output file. If two channels have the same number the first will be written over by the second in the output file and the values for the parameter occurring first in the rule will be lost.

The last step to ensure data is recorded during a simulation is running the program “DataRecord.exe” from the Supervisor. This program is not preset within the Supervisor to be selected at startup, so the researcher must manually “check” it.

Limitations of Recording Data

As mentioned in Chapter 2, SCANeR® II is a commercially available software package. The newest versions of this software are comprehensive simulator packages with all programs necessary to run a simulator included. The PTDS however, uses
VDANL for vehicle dynamics and a program called “CabinIO_2TS” to communicate with the cab; both programs are not part of SCANeR® II. Many of the parameters a researcher might be interested in are direct outputs from the dynamics model or inputs from the cab. The data recording function in VDANL does not operate when the simulator is running. Therefore, the data recording capabilities during a real-time simulation are limited. This is one motivating factor for using the stand-alone simulation package for the experiments described in Chapter 3. As an example, researchers are unable to record the angular rates about the axes of the vehicle coordinate frame.

Training Scenario Development

The final task of this study is to implement the scenarios described in the following sections in regular driver training using the PTDS in the STEP project. Five scenarios were designed by the author of this thesis to train selected driving skills. The scenarios were developed based on input from PENNDOT equipment operator instructors and the capabilities of the PTDS. Each scenario required the use of different driver skills.

The research design for this study included having students practice three of the five scenarios three times each and then get tested on all five scenarios as described in Vance et al (2002) and Hoskins et al (2002 a and b). The subjects were separated into three groups distinguished by the scenarios they practiced and whether or not they received instruction during the practice runs. There were two types of data recorded during each run of the study, PENNDOT equipment operator instructor ratings and data recorded by the simulator. The instructors were provided rating forms modeled after the
Commercial Drivers License test for each scenario. These forms divided the longer scenarios into multiple segments that roughly corresponded to individual skills. The instructors recorded individual skills ratings (use of mirrors and covering the brake pedal for examples) for each segment as well as overall segment ratings (0 – 5 scale) for each scenario. The data recorded by the simulator was programmed into each scenario by the processes previously described. The author has made an attempt to select measures that accurately represent the driver’s performance allowing a comparison of the simulator measures with the equipment operator instructor ratings. The skills targeted in each scenario were different and therefore required different measures within the simulator as described below. The simulator was programmed to record data for a portion of each scenario due to the nature of the data output and problems associated with prolonged periods of data collection.

The scenarios were named based on aspects of the respective scenarios. For the purpose of this discussion, only the portions of the scenarios where data was being collected by the simulator will be described in detail.

The results of the statistical analysis of the data taken (Vance et al, 2002) while students drove the scenarios developed for this study are shown in the figures of the following sections. Results that did not reach statistical significance are not presented. This was the case for the simulator measures collected for two of the five scenarios. The “Tunnel Backup” scenario placed students inside a tunnel with a turn that they were instructed to back out of while remaining in their lane. Still images of the rear projection during this scenario are shown in Figure 4.5. The simulator measure for this scenario
was the distance traveled with the left rear wheel over the centerline while exiting the tunnel.

![Passenger’s side approaching turn](image1) ![Driver’s side exiting tunnel](image2)

Figure 4.5: View behind truck during “Tunnel Backup” scenario

The second scenario for which there weren’t significant results was “Winter Driving”. This scenario exposed students to driving on snow by reducing the coefficient of friction between the tires and ground. The scenario began inside a tunnel that exited onto a snow-covered road as shown in Figure 4.6. The simulator measures taken in this scenario were whether or not people crashed, braking effort, and the change in speed. All measures were collected while the student drove the first snow covered turn in the scenario. The braking effort and speed change were rated based on instructor performance for the same driving condition.
55 MPH Scenario

Students in all three groups practiced the scenario “55 MPH”. This scenario started with the driver driving through a small town. A van then ran a stop sign and crossed the road on which the student was driving. The student was then instructed to bring the truck up to a speed of 55 mph. After negotiating several turns at this speed the driver came to the final section, “rollover avoidance”. During this section the road makes an abrupt turn that is poorly marked. Figure 4.7 shows the simulator approaching this turn. The simulator was programmed to record speed, lateral acceleration and distance from a parked vehicle while the student completed this section. The recording was initiated by using an “isAtPosition” command for the student’s vehicle prior to entering the final turn. This scenario was intended to target looking ahead and vehicle speed control. Lateral acceleration is directly proportional to the force that causes rollover. For
this reason the lateral acceleration measurement was used as a way of determining vehicle stability throughout the turn. Multiple equipment operator instructors drove this scenario while data was being recorded. This data was used as a baseline for assessing the students by making the maximum lateral acceleration achieved by the instructors the cutoff for the pass-fail rating of this measure.

Figure 4.7: View from simulator cab approaching poorly marked turn in the “55 MPH” scenario

The first plot of Figure 4.8 shows the average of the segment ratings by trial (three practice runs and one test run) for each of the three groups. Group C is the only group that received instruction during the practice runs. The second plot shows the percentage of each group that passed the lateral acceleration criterion for each trial.
Figure 4.8: 55MPH scenario results (Vance et al, 2002 and Hoskins et al, 2002 a and b)

The two plots of Figure 4.8 show that the general trend of the simulator measures agreed with the instructors’ ratings. Furthermore, Group C performed consistently better than Group A and Group B according to both measures.
Bridge Turn Scenario

Groups A and C practiced this scenario and the members of Group C received instruction while practicing. The skills targeted in this scenario included braking while completing a down hill turn and avoiding obstacles in the road. One section required drivers to negotiate a one-lane bridge with oncoming traffic as shown in Figure 4.9.

Figure 4.9: View from simulator cab approaching one lane bridge in the “Bridge Turn” scenario

Simulator measures were taken for a section of road with multiple turns and an embankment that came close to the right shoulder. This segment was termed the “Hill avoidance” segment. The simulator measures for this section included a measure of distance the student traveled with the left rear wheel over the centerline. Another measure was taken for the distance traveled with the right rear wheel off the side of the road. Figure 4.10 shows the results of the segment ratings for the instructors and the results for the “right side distance” simulator measure.
The two plots of Figure 4.10 show a similar trend for both measures. It should be noted however that Group C performed consistently worse than group A in the Simulator measures but performed better in the instructor assessments for the most part.

Figure 4.10: Bridge Turn performance results (Vance et al, 2002)
Train Track Scenario

The “Train Track” scenario required students to negotiate stopped traffic at a railroad crossing. Figure 4.11 shows the simulator approaching the stopped traffic at the railroad crossing. After crossing the railroad, the drivers encountered a slow moving vehicle in their lane and the decision to pass with oncoming traffic was assessed. The scenario also exposed the drivers to driving through a small town.

Figure 4.11: View from simulator cab approaching railroad crossing in the “Train Track” scenario

The simulator measures for the “Train Track” scenario included distance traveled with the left rear tire over the centerline while passing the slow moving vehicle. The option to pass this vehicle was left up to the student. In the event that the student passed the slow moving vehicle, the measure was equivalent to the distance the student remained in the opposite lane. Measures for instructors were taken to acquire a baseline performance and student performance was assessed in relation to this baseline. The
results for the simulator measurements for this scenario are presented in Figure 4.12. The instructor measures for this segment failed to reach statistical significance and are not shown here. However, it should be noted that the trend of the simulator measures for this segment was opposite to that of the instructor measures.

![Figure 4.12: Train Track scenario results (Vance et al, 2002)](image)

**Data Analysis**

The data analysis for the simulator data collected was done in MATLAB using similar functions as those used to obtain the results shown in Chapter 3 (See Appendix D). The methodology used to reduce each set of raw data included producing a graphical representation of each parameter recorded. These plots were analyzed to identify unique characteristics in the data sets that might result in inaccurate output from the computer program. A different program was written to analyze data taken from each scenario. The analyzed data took the form of numeric driver assessments that could easily be placed
into the predetermined scoring scale. The standard for this scale was determined either by measuring instructor performance for the same driving maneuver or by recognizing an optimal performance for a given maneuver. Considering the Tunnel Backup scenario can provide an example of the latter. The students were instructed to remain in their own lane while backing out of the tunnel and there is no reason other than lack of vehicle control that would result in crossing the centerline. For these reasons the optimal performance for this scenario measure was to not cross the centerline.
CHAPTER 5
CONCLUSIONS

It is clear from the material presented in Chapter 1 that there are many ways to validate driving simulators. The Pennsylvania Truck Driving Simulator has been improved through multiple upgrades as described in Chapter 2. Many of these upgrades have been instrumental in the success of the research described in Chapter 3 and Chapter 4.

A methodology for validating the motion cueing for the PTDS was developed and implemented. The gain settings for the pitch and roll motion of the PTDS have been set based on the results from straight-line braking and double lane change maneuvers.

The development of training scenarios within the PTDS has been described in Chapter 4. The implementation of these scenarios in the Simulator Training Evaluation Program as described in Chapter 4 gives support for the usefulness of the PTDS as a training device. The use of data recorded by the simulator for assessing driver performance received tentative support from the results presented here. While some measures collected by the simulator in some scenarios agreed with instructor ratings such as the 55MPH scenario, it is clear that further research is needed to identify measures that consistently reflect instructor’s measures for certain driving maneuvers before they can be relied on with confidence. There appear to be certain driving maneuvers where
subjective measures are imperative to capturing an accurate assessment of a driver’s performance as noted by Farmer et al (1999).

**Future Work**

While working with the PTDS many possibilities for improvements and research have presented themselves. Brief descriptions of these are presented below.

**Increased Maneuver Characterization**

The work presented in Chapter 3 included two driving maneuvers. For a more accurate characterization of the effect the gain settings have of the cab response more maneuvers could be investigated. The work presented in Heydinger et al (1990) and Chrostos and Grygier (1997) can provide insight into driving maneuvers that can be used in the methodology described in Chapter 3.

**Frequency Response Evaluation**

The motions of the simulator cab were investigated for angular displacements and velocities and the contribution of the angular displacements to linear accelerations. Another significant component of vehicle sprung mass motion is the frequency response. An investigation into the frequency response of a vehicle similar to the simulated vehicle could include vertical vibrations experienced in the driver seat of the simulator for
different driving maneuvers such as straight–line driving at different speeds and hitting an obstacle of known dimensions. This work could be accomplished using a similar methodology to the one described in Chapter 3 with the comparisons being made between measured cab responses in the simulator with those measured within a real vehicle of similar dimensions. During the proposed investigation a more complete characterization of the motion cueing could be found through altering the filtering parameters within the motion cueing configuration files.

**Characterization of Cab Response Delay**

The use of the synchronized clocks in the validation study presented in Chapter 3 allowed for reasonable comparisons of the cab response in the simulator with the sprung mass response predictions from VDANL. With more accurate timers for both the driving input and the response measuring device that are well synchronized, the effect the gain settings have on the perceived delay of the cab motion can be characterized and considered in the gain settings.

**Increased Characterization of Simulator Collected Data for Driver Assessment**

As mentioned above, the use of simulator-collected data in assessing driver performance in a training program needs further investigation. The ultimate goal of an investigation of this type would be a set of simulator-collected data that can be relied on for accurately measuring driver performance.
Transmission Improvements

The transmission within the PTDS is currently represented by micro-switches and there is no interaction between the clutch and simulated transmission. For this reason a driver can shift without using the clutch. This is not an accurate representation of a manual transmission. The PENNDOT instructors noted that meshing gears is a difficult skill for novice drivers and a more accurate representation of the transmission within the PTDS would increase its value as a training tool. Within the dynamics model there exists provisions for three more forward gear positions so the simulated transmission could be developed to accommodate a more complex shifting pattern than the “H” pattern currently used.
BIBLIOGRAPHY


http://www.inrets.fr/ur/sara/Pg_simus_e.html


Reed, M. “Validation of a Low-Cost Driving Simulator Using a Telephone Dialing Task.” The Link Foundation Fellowship in advanced Simulation and Training, Transportation Research Institute, University of Michigan, Ann Arbor, Michigan, 1995.


PUBLICATIONS


APPENDIX A

STEPS FOR CONFIGURING A VISUAL GENERATION COMPUTER FOR THE PTDS

The software for the simulator was originally set up to be run using Windows NT 4.0. In an effort to preserve as many system configurations as possible, this operating system was retained in the upgrade of the visual generation computers.

Much of the following information was taken from a correspondence from Loic Joly who was working for Renault at the time of this upgrade. Mr. Joly has been very influential in helping overcome software problems that have arisen with the simulator.

1. Install the operating system Windows NT 4.0. Make the appropriate adjustments to the computer name, IP address and login passwords. Table A.1 contains all IP addresses for the TRUCKSIM Workgroup.

Table A.1: IP Addresses for computers in the TruckSim workgroup

<table>
<thead>
<tr>
<th>Computer Name</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grenoble</td>
<td>128.118.33.94</td>
</tr>
<tr>
<td>Marseille</td>
<td>128.118.33.95</td>
</tr>
<tr>
<td>Lyon</td>
<td>128.118.33.96</td>
</tr>
<tr>
<td>Paris</td>
<td>128.118.33.97</td>
</tr>
<tr>
<td>Nantes</td>
<td>128.118.33.98</td>
</tr>
<tr>
<td>Toulouse</td>
<td>128.118.33.99</td>
</tr>
<tr>
<td>Avignon</td>
<td>128.118.33.100</td>
</tr>
<tr>
<td>Open</td>
<td>128.118.33.101</td>
</tr>
<tr>
<td>Open</td>
<td>128.118.33.102</td>
</tr>
<tr>
<td>Bordeaux</td>
<td>128.118.33.103</td>
</tr>
</tbody>
</table>

2. Map a network drive to the D:\ drive on Marseille.
3. Set up an environment variable named CATS_PATH that contains the path of the directory that contains the datafile.cfg file. All other visual generation
computers will contain the same variable so to view the correct path look on another computer. Environment variables are found in the System Properties. The path to the *.cfg file was m:\cats\data\config where m:\ is the mapping to the D:\ drive on Marseille.

4. For the sake of better loading time some data necessary for visual generation is copied locally. This data is defined in the datafile.cfg file. This file is located on Marseille in D:\cats\data\config. The necessary files are located in C:\cats on the visual generation computers. This folder contains the folder data that contains the folders terrain and vehicle. The entire cats folder should be recreated in the C:\ drive of the new computer.

5. Icons to launch the deamon programs need to be added. A command line parameter that states the name of the computer is required to run the program correctly (-p DEMON_XXX, where XXX is the computer name). If the computer name changes, you must update the process.cfg file located in m:\cats\data\config where m:\ is the D:\ drive on Marseille.

6. The file Hosts must be added to the new computer. This file should be located in the following directory C:\Winnt\system32\drivers\etc on the new PC.
APPENDIX B

PENNSYLVANIA TRUCK DRIVING SIMULATOR OPERATING INSTRUCTIONS

This file contains a list of steps to follow when starting the simulator with everything completely shutdown.

1. Turn on power strip under the hood that powers the motion base computer
   a. Listen for fan; If fan is inaudible the motion base should not be run

2. Turn on motion base computer; (switch is at the back of the machine)

3. Turn on Projectors
   a. The power for the two rear projectors is the gray button on top of the machine and it must be pressed twice consecutively
   b. The power for the front projectors is near where the power cord connects

4. Check each of the PC’s to see if they are running the program “Daemon”
   a. This program must be running on each computer
   b. The program can be seen to be running by a minimized window at the bottom of the screen
   c. The icon for the program is a tree

5. Make sure that none of the PC’s are running a screen saver

6. Close all windows programs outside of “Daemon”

7. Make sure all kill switches are released
   a. There are 3, under the hood, in the cab, and by the host computer
8. Go to the host computer (Marseilles) and start the program “Supervisor” from the desktop

9. Within the GUI for the Supervisor make certain all necessary programs are checked on the left side of two columns
   a. All programs necessary for normal operation will be checked automatically
   b. “RECORD” will not be checked and the simulator will not record any data unless done so manually

10. Click on the “Launch All” button within the Supervisor GUI
    a. All programs to run a simulation should be started
    b. Go to the supervisor window and check to see if there are any “Miss” programs
    c. If anything missed, double click on the black “Miss” box to start that program or click “Launch All” again

11. If everything is running go to the window of “Wcarto”
    a. This controls the traffic of a simulation
    b. Select “File” from the top toolbar and scroll to “Open”
    c. A list of the traffic files corresponding to the scenarios that have been previously developed will appear
    d. Select the appropriate file and open it

12. Select the Wcarto window by clicking on it and press the space bar once all screens show a projected image to begin the simulation
APPENDIX C

INSTRUCTIONS FOR GENERATING A NEW VEHICLE WITHIN THE PTDS

Background

Within the Penn State Truck Driving Simulator there exist a large variety of parameters that can be changed to affect the vehicle’s performance. While several models exist within the simulator including a variety of vehicles ranging from loaded dump trucks to passenger sedans, it is often beneficial to create new vehicle models. The following text details the process necessary to duplicate an existing vehicle and rename the duplicate to create a new vehicle. By duplicating the files for an existing vehicle model you avoid omitting any necessary files from the directory and in the event that each file were created independently, the omission of important vehicle parameters. This process may sound trivial, but it is involved and this list will help avoid the omission of crucial steps.

Duplicating Parameter Files:

The directory that contains the vehicle parameter files is found in Lyon_C:\modelVDANL\Parameter_Files. There exist other parameter files within Lyon_C:\VDANL_2TS\Data_set. These files are no longer in use! The files in this
directory were used by the original version of VDANL that has been upgraded to a newer version.

The best way to proceed having identified the correct extension for the existing vehicle parameter files is to select a vehicle that has similar characteristics to the vehicle to be modeled. The vehicle model for the ‘Taurus’ is the model of a sedan and the model ‘truck_PTI_loaded’ is a loaded dump truck with a single rear axle. At the time of this report, the dynamics model was not capable of supporting two rear axles. Each folder of vehicle parameters has a complete set of files necessary for simulating that vehicle as well as other files that may have been created to backup files or to save specifications for the vehicle or a given simulation of that vehicle. Table C.1 gives the list of parameter files necessary to simulate a vehicle.

Table C.1: Files required for a complete dynamics model within VDANL

<table>
<thead>
<tr>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle_name.vcf</td>
</tr>
<tr>
<td>Vehicle_name.vpf</td>
</tr>
<tr>
<td>Vehicle_name_Driver.vpf</td>
</tr>
<tr>
<td>Vehicle_name_DriveTrain.vpf</td>
</tr>
<tr>
<td>Vehicle_name_Nonlin.vpf</td>
</tr>
<tr>
<td>Vehicle_name_Param.vpf</td>
</tr>
<tr>
<td>Vehicle_name_Suspension.vpf</td>
</tr>
<tr>
<td>Vehicle_name_Tire.vpf</td>
</tr>
<tr>
<td>Vehicle_name_Vehicle.vpf</td>
</tr>
</tbody>
</table>

The list above corresponds to vehicle parameter files for the vehicle named “Vehicle_name”. Once the file of vehicle parameters has been duplicated you must rename the file with the name of the new vehicle. The name chosen for the new vehicle should describe that vehicle in some way so that it will be easily recognizable in the
future. The name of the new vehicle file must now be used to rename each of the files listed in Table C.1. Note that the new vehicle name should only replace the text “Vehicle_name” in each of the file names. Each of these files can be opened and viewed using a text editor such as Notepad. After each file has the appropriate name, the heading within each file except the two files appearing in rows one and two of Table C.1 should be updated to reflect these changes. The ‘%’ within these files denotes comments and everything after them and on the same line is not used in simulation. The two files Vehicle_name.vcf and Vehicle_name.vpf need to be treated separately. Within these files are extensions of the other vehicle parameter files and the specifics of how these files function within a simulation is detailed in the VDANL help window that can be accessed by running VDANL on Lyon and selecting the help tab. Vehicle_name.vpf should have the title updated and each file extension within this file must be changed to reflect the location of the new vehicle parameter files. Alterations to Vehicle_name.vcf will reflect the instructions given in the VDANL help window. This file is not used when running the simulator, only when running stand-alone simulations.

Creating Vehicle Graphics Files

Graphics files within Marseille_D:\cats\data\vehicle must be duplicated and renamed to reflect the name of the new vehicle. Within the folder given above, open the ‘tech’ folder and create a copy of the file that corresponds to the original vehicle that was copied and rename this copy with the name of the new vehicle. Open the file Models.db and add a line at the bottom of the file with the next higher number and the name of the
new vehicle. Now go into the Marseille_D:\cats\data\vehicle\graphic folder and repeat the duplicating and renaming process for the new vehicle using the entire folder for the duplicated vehicle. Due to the setup of the data on the visual computers, this entire process must be repeated on each of the visual generation computers. However, making the changes on Marseille and then using network connections to copy and paste the information to the appropriate folder on each of the five computers facilitates the process. The files are located within the ‘cats’ directory on the hard drive of each of the computers.
%Andrew Hoskins
%created - 05.25.02
% file for analyzing lane change data taken from the simulator, the DMU and VDANL, Written for MATLAB v.6.0

clear all

load the text data taken from the Dynamic Measurement Unit (DMU) this data takes the form of 9 columns of data each row corresponds to the time step specified in the first column (t_dmu) given in seconds. Columns 2, 3, and 4 are angular velocities about the x, y, and z axes respectively. Columns 5, 6, and 7 are linear accelerations along each axis of the DMU given in multiples of the acceleration due to gravity or G's. The DMU was oriented in the cab such that the positive x-axis was toward the front of the truck, the y-axis is positive out the passenger side door leaving the z-axis positive downward. Column 8 is the temperature in Celsius degrees and column 9 is a timer of some sort but is not important to the current study.

[t_dmu_1, x_ang_vel_1, y_ang_vel_1, z_ang_vel_1, long_acc_1, dmu_lat_acc_1, vert_acc_1, temp_1, timer_1]=textread('dmu_50_001_1.txt', '%f %f %f %f %f %f %f %f', -1);
[t_dmu_2, x_ang_vel_2, y_ang_vel_2, z_ang_vel_2, long_acc_2, dmu_lat_acc_2, vert_acc_2, temp_2, timer_2]=textread('dmu_50_005_1.txt', '%f %f %f %f %f %f %f %f', -1);
[t_dmu_3, x_ang_vel_3, y_ang_vel_3, z_ang_vel_3, long_acc_3, dmu_lat_acc_3, vert_acc_3, temp_3, timer_3]=textread('dmu_50_015_1.txt', '%f %f %f %f %f %f %f %f', -1);
[t_dmu_4, x_ang_vel_4, y_ang_vel_4, z_ang_vel_4, long_acc_4, dmu_lat_acc_4, vert_acc_4, temp_4, timer_4]=textread('dmu_50_03_2.txt', '%f %f %f %f %f %f %f %f', -1);
[t_dmu_5, x_ang_vel_5, y_ang_vel_5, z_ang_vel_5, long_acc_5, dmu_lat_acc_5, vert_acc_5, temp_5, timer_5]=textread('dmu_50_05_1.txt', '%f %f %f %f %f %f %f %f', -1);

Load the data recorded by the simulator. This data takes the form of eight columns, each row corresponds to a record number given in column 1. The remaining columns are configurable through programming in the simulator program for scenarios "Mice". For this study the simulator was programmed so column 2 is time (s), column 3 is vehicle
speed (kph), column 4 is steering wheel input on a [-1, 1] scale, column 5 is distance from a parked vehicle, column 6 is the lateral acceleration (m/s^2), column 7 is road slope change in height (m) for a traversing 100m and column eight is the distance of the right side of the vehicle from the right side of the road (m).

[number_1, t_sim_1, speed_1, steer_1, dist_1, lat_acc_1, slope_1, pos_1] = textread('sim_50_001_1.txt', '%f %f %f %f %f %f %f %f', -1);
[number_2, t_sim_2, speed_2, steer_2, dist_2, lat_acc_2, slope_2, pos_2] = textread('sim_50_005_1.txt', '%f %f %f %f %f %f %f %f', -1);
[number_3, t_sim_3, speed_3, steer_3, dist_3, lat_acc_3, slope_3, pos_3] = textread('sim_50_015_1.txt', '%f %f %f %f %f %f %f %f', -1);
[number_4, t_sim_4, speed_4, steer_4, dist_4, lat_acc_4, slope_4, pos_4] = textread('sim_50_03_2.txt', '%f %f %f %f %f %f %f %f', -1);
[number_5, t_sim_5, speed_5, steer_5, dist_5, lat_acc_5, slope_5, pos_5] = textread('sim_50_05_1.txt', '%f %f %f %f %f %f %f %f', -1);

% convert speed to m/s
speed_1_mps = speed_1 / 3.6;
speed_2_mps = speed_2 / 3.6;
speed_3_mps = speed_3 / 3.6;
speed_4_mps = speed_4 / 3.6;
speed_5_mps = speed_5 / 3.6;

% find the total amount of time elapsed over the entire trial for both
% dmu and simulator timers

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % first set % %
[p_1, q_1] = size(t_sim_1);
max_sim_t_1 = t_sim_1(p_1);
[r_1, s_1] = size(t_dmu_1);
max_dmu_t_1 = t_dmu_1(r_1);

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % 2ND set % %
[p_2, q_2] = size(t_sim_2);
max_sim_t_2 = t_sim_2(p_2);
[r_2, s_2] = size(t_dmu_2);
max_dmu_t_2 = t_dmu_2(r_2);

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % 3RD set % %
[p_3, q_3] = size(t_sim_3);
max_sim_t_3 = t_sim_3(p_3);
[r_3, s_3] = size(t_dmu_3);
max_dmu_t_3 = t_dmu_3(r_3);

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % 4TH set % %
[p_4, q_4] = size(t_sim_4);
max_sim_t_4 = t_sim_4(p_4);
[r_4, s_4] = size(t_dmu_4);
max_dmu_t_4 = t_dmu_4(r_4);

%%%%%%%5TH set
[p_5, q_5] = size(t_sim_5);
max_sim_t_5 = t_sim_5(p_5);
[r_5, s_5] = size(t_dmu_5);
max_dmu_t_5 = t_dmu_5(r_5);

%Find difference in the two clocks

  t_diff_ = [(max_dmu_t_1 - max_sim_t_1)
             (max_dmu_t_2 - max_sim_t_2)
             (max_dmu_t_3 - max_sim_t_3)
             (max_dmu_t_4 - max_sim_t_4)
             (max_dmu_t_5 - max_sim_t_5)];

%%%%%%%find the Roll angle measured by the dmu by integrating the Roll velocity data
% and give it in deg. Also output the maximum of this value over the entire trial.

% first set

  dmu_roll_ang_1(1)=0;
  for i = 1:r_1-1
    dmu_roll_ang_1(i+1) = ((x_ang_vel_1(i))*(t_dmu_1(i+1)-t_dmu_1(i))+ 0.5 *
                             (x_ang_vel_1(i+1)-x_ang_vel_1(i))*(t_dmu_1(i+1)-t_dmu_1(i)))+dmu_roll_ang_1(i);
  end

  dmu_roll_ang_1= dmu_roll_ang_1'*(-1)-1.359;
  max_roll_1=max(dmu_roll_ang_1);
  min_roll_1=min(dmu_roll_ang_1);

% 2ND set

  dmu_roll_ang_2(1)=0;
  for i = 1:r_2-1
    dmu_roll_ang_2(i+1) = ((x_ang_vel_2(i))*(t_dmu_2(i+1)-t_dmu_2(i))+ 0.5 *
                             (x_ang_vel_2(i+1)-x_ang_vel_2(i))*(t_dmu_2(i+1)-t_dmu_2(i)))+dmu_roll_ang_2(i);
  end

  dmu_roll_ang_2= dmu_roll_ang_2'*(-1)-0.6175;
  max_roll_2=max(dmu_roll_ang_2);
  min_roll_2=min(dmu_roll_ang_2);
% 3RD set
dmu_roll_ang_3(1)=0;
for i = 1:r_3-1
    dmu_roll_ang_3(i+1) = ((x_ang_vel_3(i))*(t_dmu_3(i+1)-t_dmu_3(i))+ 0.5 *
        (x_ang_vel_3(i+1)-x_ang_vel_3(i))*(t_dmu_3(i+1)-t_dmu_3(i)))+dmu_roll_ang_3(i);
end
dmu_roll_ang_3= dmu_roll_ang_3*(-1)-0.77;
max_roll_3=max(dmu_roll_ang_3);
min_roll_3=min(dmu_roll_ang_3);

% 4TH set
dmu_roll_ang_4(1)=0;
for i = 1:r_4-1
    dmu_roll_ang_4(i+1) = ((x_ang_vel_4(i))*(t_dmu_4(i+1)-t_dmu_4(i))+ 0.5 *
        (x_ang_vel_4(i+1)-x_ang_vel_4(i))*(t_dmu_4(i+1)-t_dmu_4(i)))+dmu_roll_ang_4(i);
end
dmu_roll_ang_4= dmu_roll_ang_4*(-1)-0.905;
max_roll_4=max(dmu_roll_ang_4);
min_roll_4=min(dmu_roll_ang_4);

% 5TH set
dmu_roll_ang_5(1)=0;
for i = 1:r_5-1
    dmu_roll_ang_5(i+1) = ((x_ang_vel_5(i))*(t_dmu_5(i+1)-t_dmu_5(i))+ 0.5 *
        (x_ang_vel_5(i+1)-x_ang_vel_5(i))*(t_dmu_5(i+1)-t_dmu_5(i)))+dmu_roll_ang_5(i);
end
dmu_roll_ang_5= dmu_roll_ang_5*(-1)-0.73;
max_roll_5=max(dmu_roll_ang_5);
min_roll_5=min(dmu_roll_ang_5);

%Convert the steering data into radians
%1ST set
steer_rad_1 = steer_1 * (2 * pi);
%2ND set
steer_rad_2 = steer_2 * (2 * pi);
%3RD set
steer_rad_3 = steer_3 * (2 * pi);
%4TH set
steer_rad_4 = steer_4 * (2 * pi);
%5TH set
steer_rad_5 = steer_5 * (2 * pi);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%subtract the time duration to the point the lane change event occurred
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
new_simtime_1 = t_sim_1 - 54.9;
new_simtime_2 = t_sim_2 - 34.9;
new_simtime_3 = t_sim_3 - 39.9;
new_simtime_4 = t_sim_4 - 34.9;
new_simtime_5 = t_sim_5 - 29.9;

new_dmutime_1 = t_dmu_1 - 54.9;
new_dmutime_2 = t_dmu_2 - 34.9;
new_dmutime_3 = t_dmu_3 - 39.9;
new_dmutime_4 = t_dmu_4 - 34.9;
new_dmutime_5 = t_dmu_5 - 29.9;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%find the dmu roll velocity maximum value during the onset of breaking in deg/s
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%first set
pos_1 = sign(new_dmutime_1)+1;
[c_1, d_1] = find(pos_1);
long_1 = size(c_1);
for i=1:long_1(1)
    new_roll_vel_1(i) = x_ang_vel_1(c_1(i));
end
max_roll_vel_1 = max(new_roll_vel_1);
min_roll_vel_1 = min(new_roll_vel_1);

%2ND set
pos_2 = sign(new_dmutime_2)+1;
[c_2, d_2] = find(pos_2);
long_2 = size(c_2);
for i=1:long_2(1)
    new_roll_vel_2(i) = x_ang_vel_2(c_2(i));
end
max_roll_vel_2 = max(new_roll_vel_2);
min_roll_vel_2 = min(new_roll_vel_2);

%3RD set
pos_3 = sign(new_dmutime_3)+1;
[c_3, d_3] = find(pos_3);
long_3 = size(c_3);
for i=1:long_3(1)
    new_roll_vel_3(i) = x_ang_vel_3(c_3(i));
end
max_roll_vel_3 = max(new_roll.vel_3);
min_roll_vel_3 = min(new_roll_vel_3);

%4TH set
pos_4 = sign(new_dmutime_4)+1;
[c_4, d_4] = find(pos_4);
long_4 = size(c_4);
for i=1:long_4(1)
    new_roll_vel_4(i) = x_ang_vel_4(c_4(i));
end
max_roll_vel_4 = max(new_roll_vel_4);
min_roll_vel_4 = min(new_roll_vel_4);

%5TH set
pos_5 = sign(new_dmutime_5)+1;
[c_5, d_5] = find(pos_5);
long_5 = size(c_5);
for i=1:long_5(1)
    new_roll_vel_5(i) = x_ang_vel_5(c_5(i));
end
max_roll_vel_5 = max(new_roll_vel_5);
min_roll_vel_5 = min(new_roll_vel_5);

max_roll_vel = [max_roll_vel_1
                 max_roll_vel_2
                 max_roll_vel_3
                 max_roll_vel_4
                 max_roll_vel_5];

min_roll_vel = [min_roll_vel_1
                min_roll_vel_2
                min_roll_vel_3
                min_roll_vel_4
                min_roll_vel_5];

max_roll = [max_roll_1
            max_roll_2
            max_roll_3
            max_roll_4
            max_roll_5];
min_roll = [min_roll_1
            min_roll_2
            min_roll_3
            min_roll_4
            min_roll_5];

gains = [0.001
          0.005
          0.015
          0.03
          0.05];

Load VDANL data
%Load VDANL data
load('lane_001.mat', '-mat');
time_001 = Time - 55;
steer_001 = STEER;
%Convert units
%linear accelerations in g's
RoAx_lat_acc_001 = RALT_ACC/32.1846;
SM_long_acc_001 = SM_LGACC/32.1846;
SM_lat_acc_001 = SM_LTACC/32.1846;

%linear velocities in m/s
CG_lat_vel_001 = CGLT_VEL/3.2808;
CG_long_vel_001 = CGLG_VEL/3.2808;

%Angular measures in degrees
roll_deg_001 = ROLLANG * (180/3.14159);
roll_vel_deg_001 = ROLLVEL * (180/3.14159);
roll_acc_deg_001 = ROLLACC * (180/3.14159);

load('lane_005.mat', '-mat');
time_005 = Time - 35;
steer_005 = STEER;
%Convert units
%linear accelerations in g's
RoAx_lat_acc_005 = RALT_ACC/32.1846;
SM_long_acc_005 = SM_LGACC/32.1846;
SM_lat_acc_005 = SM_LTACC/32.1846;

%linear velocities in m/s
CG_lat_vel_005 = CGLT_VEL/3.2808;
CG_long_vel_005 = CGLG_VEL/3.2808;

%Angular measures in degrees
roll_deg_005 = ROLLANG * (180/3.14159);
roll_vel_deg_005 = ROLLVEL * (180/3.14159);
roll_acc_deg_005 = ROLLACC * (180/3.14159);

load('lane_015.mat', '-mat');
time_015 = Time - 40;
steer_015 = STEER;
%Convert units
%linear accelerations in g's
RoAx_lat_acc_015 = RALT_ACC/32.1846;
SM_long_acc_015 = SM_LGACC/32.1846;
SM_lat_acc_015 = SM_LTACC/32.1846;

%linear velocities in m/s
CG_lat_vel_015 = CGLT_VEL/3.2808;
CG_long_vel_015 = CGLG_VEL/3.2808;

%Angular measures in degrees
roll_deg_015 = ROLLANG * (180/3.14159);
roll_vel_deg_015 = ROLLVEL * (180/3.14159);
roll_acc_deg_015 = ROLLACC * (180/3.14159);

load('lane_03_2.mat', '-mat');
time_03 = Time - 35;
steer_03 = STEER;
%Convert units
%linear accelerations in g's
RoAx_lat_acc_03 = RALT_ACC/32.1846;
SM_long_acc_03 = SM_LGACC/32.1846;
SM_lat_acc_03 = SM_LTACC/32.1846;

%linear velocities in m/s
CG_lat_vel_03 = CGLT_VEL/3.2808;
CG_long_vel_03 = CGLG_VEL/3.2808;

%Angular measures in degrees
roll_deg_03 = ROLLANG * (180/3.14159);
roll_vel_deg_03 = ROLLVEL * (180/3.14159);
roll_acc_deg_03 = ROLLACC * (180/3.14159);

load('lane_05.mat', '-mat');
time_05 = Time - 30;
steer_05 = STEER;

%Convert units
%linear accelerations in g's
RoAx_lat_acc_05 = RALT_ACC/32.1846;
SM_long_acc_05 = SM_LGACC/32.1846;
SM_lat_acc_05 = SM_LTACC/32.1846;

%linear velocities in m/s
CG_lat_vel_05 = CGLT_VEL/3.2808;
CG_long_vel_05 = CGLG_VEL/3.2808;

%Angular measures in degrees
roll_deg_05 = ROLLANG * (180/3.14159);
roll_vel_deg_05 = ROLLVEL * (180/3.14159);
roll_acc_deg_05 = ROLLACC * (180/3.14159);