The Pennsylvania State University

The Graduate School

Department of Mechanical and Nuclear Engineering

Development of a Rollover-Warning Device for Road Vehicles

A Thesis in

Mechanical Engineering

by

Robert W. Goldman

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2001

We approve the thesis of Robert W. Goldman

Date of Signature

Bohdan T. Kulakowski
Professor of Mechanical Engineering
Thesis Co-Adviser
Chair of Committee

Moustafa El-Gindy
Sr. Research Associate of Mechanical Engineering
Thesis Co-Adviser

Christopher Rahn
Associate Professor of Mechanical Engineering

Ageliki Elefteriadou
Associate Professor of Civil Engineering

Richard C. Benson
Professor of Mechanical Engineering
Head of the Department of Mechanical and
Nuclear Engineering

Abstract

This dissertation describes the research and development of a rollover-warning device (RWD) to be used for road vehicles, particularly heavy vehicles with high center-of-gravity height to track-width ratios. The RWD uses Artificial Neural Networks to essentially 'learn' the dynamics of a road vehicle and make a prediction of the instantaneous roll stability using dynamic states of the vehicle that are relatively easy to measure. The state of the dynamic roll stability, specifically the load-transfer ratio (LTR), is used in conjunction with the rate of change of LTR as inputs to the RWD based upon a Fuzzy Logic rule-base for determination of an output warning level. The device is based upon extensive computer simulation and verified using experimentally acquired data. Due to the fact that the device is passive and therefore depends upon the driver of the vehicle to take corrective action, the success of the device is limited to helping with dangerous maneuvers that develop slowly, such as during an on-ramp maneuver.

Table of Contents

List of Figures

List of Tables

Acknowledgments

The author would like to thank the Department of Mechanical & Nuclear Engineering and particularly Dr. Richard C. Benson, Head of the Department, for providing the majority of the financial support over the course of this research. Furthermore, Dr. Benson allowed the author to gain his first familiarity with teaching and is an experience that will never be forgotten.

The author would also like to thank Chair of the Committee and Thesis Co-Adviser, Dr. Bohdan Kulakowski, for his support over the past 10 years the author has had the pleasure to know him; first as an undergraduate in undergraduate engineering courses, then as his graduate student. Dr. Kulakowski's support for his students and his devotion teaching is unsurpassed. The author always found Dr. Kulakowski's door open to talk and answer questions and his knowledge of engineering systems remarkable.

The author would like to thank the Thesis Co-Adviser, Dr. Moustafa El-Gindy for the tremendous amount of intellectual and emotional support over the course of this research. Dr. El-Gindy was always available to help and spent countless hours with the author over the past few years assisting and guiding his research. Dr. El-Gindy's tireless devotion to his students, teaching and his research is a real credit to the University.

Finally, the author would like to thank his father, Dr. George Goldman for a lifetime of intellectual, emotional and financial support that allowed the author to complete this research.

I. Introduction

Rollover accidents are a significant problem on North American highways, if not the entire civilized world. Using data in the National Accident Sampling System from 1989, Hinch et. al [1] concluded that there were approximately 213,200 rollover accidents involving passenger cars, light trucks, sport utility vehicles and vans. Of these 190,600 (89%) were single-vehicle rollover accidents, i.e., accidents involving only the vehicle that rolled over. Furthermore, of the 15,901 fatalities in single-vehicle accidents reported in the 1990 Fatal Accident Reporting System, 8,088 (51%) were related to rollover. Hinch et. al proceeded to investigate the number of rollover related fatalities per type of vehicle, whether it be passenger cars, light trucks, etc., and concluded that light trucks, sport utility vehicles and vans had a rollover fatality rate 103% greater than that for passenger cars.

Accidents involving heavy trucks (tractor semi-trailers, etc.) have also contributed significantly to the number of injuries and fatalities on North American highways over the past few decades. In addition, monetary expenses, consisting of damage to roadways and environmental cleanup from hazardous spills, are at an all-time high. A 1988 report from the National Highway Traffic Safety Administration (NHTSA) showed rollover occurred in 52% of the heavy vehicle accidents where the driver was killed, where a recent US Department of Transportation report [2] put the number at approximately 60%. Also, an earlier report by Ervin et. al [3] concluded that rollover of heavy vehicles was responsible for 95% of the bulk spillage of hazardous materials. To make matters worse, the number of heavy trucks on the roads in North America increases every year and economic demands continuously put pressure on regulators to increase the size and weight limits on heavy trucks. Of course, the reason companies desire increases in size and weight limits is obvious, but these decisions seem to be made without much regard to the reduction in rollover stability this may cause. Therefore, the frequency of rollover accidents in the years to come will almost certainly grow unless measures are taken to compensate for not only the increase in number of heavy trucks on the road, but also for the reduction in the roll stability due to the changes in size and weight regulations.

II. Objective

The objective of this research will be to develop a rollover-warning device (RWD) to be implemented onto a road vehicle. The final product with a method to develop an electronic apparatus that will measure pertinent dynamic properties of the vehicle, make predictions about other dynamic vehicle properties based upon those measured, and finally make a determination on the instantaneous roll stability of the vehicle, all in real-time. The core of this research will involve determination of the pertinent stability cues and how they can be measured or predicted for use with the RWD. Using the instantaneous stability condition, a warning signal will be produced by the RWD to alert the driver of an impending instability condition. The research and development of this RWD can be dissected into six parts:

1. Define a stability indicator (SI) that provides a measure of the instantaneous roll stability of the vehicle
2. Determine how to physically measure or predict the SI determined in Step 2
3. Determine how the RWD would use the SI
4. Develop the RWD based on Steps 1-3 using computer simulation
5. Obtain experimental data for a vehicle performing specific maneuvers for off-line testing and verification of the RWD
6. Propose a method for the implementation of the RWD via hardware to be on-board a vehicle and performing in real-time

III. Literature Review

List of Symbols

ϕ	Roll Angle
ϕ_T	Wheel Lift-Off (Threshold) Roll Angle
A_y	Lateral Acceleration
ANN	Artificial Neural Network
b	Neural Network Bias
BP	Backpropogation
CG	Center of Gravity
CRT	Constant-Radius Turning
CSV	Critical Sliding Velocity
DOF	Degree of Freedom
FLIS	Fuzzy Logic Inference System
GVW	Gross Vehicle Weight
h	Height to Center of Gravity
LC	Lane Change
LMA	Levenberg-Marquardt Algorithm
LTR	Load-Transfer Ratio
MF	Membership Function
MSE	Mean-Squared Error
p	Neural Network Input
RA	Roll Acceleration
RC	Roll Center
RC_s	Sprung Mass Roll Center
RC_t	Tire Roll Center
RPER	Rollover Potential Energy Reserve
RPM	Rollover Prevention Metric
RSF	Roll Safety Factor
RSM	Roll Stability Margin
RWD	Rollover-Warning Device
SF	Steering Factor
SPR	Side Pull Ratio
SRT	Static Roll Threshold
SSF	Static Stability Factor
ST	Sharp Turning
SVA	Single Vehicle Accident
T	Track Width
TTR	Tilt-Table Ratio
TWR	Track Width Ratio
W	Vehicle Weight
w	Neural Network Weight
WHLR	Widrow-Hoff Learning Rule
y	Neural Network Output
YA	Yaw Acceleration

III.1 Vehicle Roll Stability/Dynamics of Rollover

Although a technically correct definition of rollover would be the state at which the overall center of gravity (CG) of the vehicle has moved laterally passed the vehicle's 'balance point', researchers typically define the rollover point or rollover *threshold* as the state where the load from one side of the vehicle has transferred to the other side. This more conservative definition of rollover is standard in the literature for the main reason of safety.

III.1.1 Simple Rigid Model

The cause of vehicular rollover is apparent in even the simplest analytical models. For instance, in the process of analyzing more complex models, Ervin [4] examines the static roll performance of the completely rigid vehicle shown in Figure 1. The vehicle is modeled as lying on a flat surface and made up of a single axle rigidly attached to the vehicle body (i.e., no suspension) and with rigid tires. The model has a single DOF (the roll angle ϕ) and is general enough to describe any type of rigid multi-axle vehicle, e.g., passenger cars, utility vehicles or heavy articulated or non-articulated vehicles.

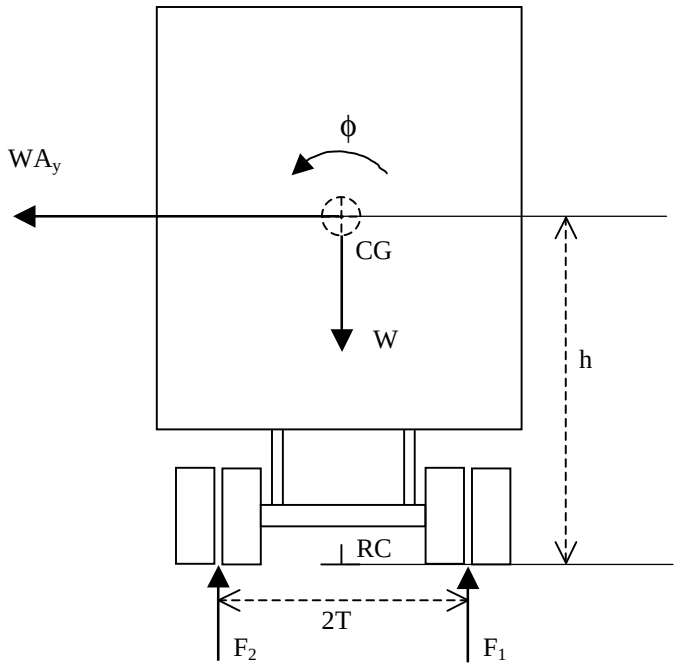


Figure 1: Rigid vehicle model

As the vehicle negotiates a turn (in this case a right-hand turn), a lateral force is developed, WA_y on the vehicle, which may be modeled as being applied to the CG of the vehicle. Note that A_y is measured in g's (unitless) and therefore the units of WA_y are still

those of force. In addition, lateral friction at the tires is not shown in the diagram. Summing the moments on the vehicle about the tire/road interface centered between the tires and invoking equilibrium reveals (for small angles):

$$WA_y h + (F_1 - F_2)T + Wh\phi = 0$$

Ervin refers to the terms from left to right side as the *primary overturning moment*, the *restoring moment* and the *lateral displacement moment*. This equation may be re-arranged into a more convenient form shown in Equation 1.

$$WA_y h = (F_2 - F_1)T - Wh\phi \quad (1)$$

Note how the lateral displacement moment term caused by roll of the vehicle subtracts from restoring moment term available at the tires. Ervin refers to the summation of the two terms as the *net stabilizing moment* and points out that rollover occurs when the left side of Equation 1 is greater than the right side, i.e., the primary overturning moment is greater than the net restoring moment. However, to examine the so-called roll threshold of this vehicle, we note that since this vehicle is rigid, the only way for it to exhibit a non-zero roll angle (non-zero ϕ) is for all the weight to have shifted to one side of the vehicle. Therefore, during a right-hand turn and at the instant the vehicle in Figure 1 starts to roll, $F_1 = 0$, $F_2 = W$ and $\phi = 0$. Since the roll threshold is defined as the value of lateral acceleration at this point, solving Equation 1 for the lateral acceleration yields

$$A_y = T/h.$$

Therefore, the roll threshold of a rigid vehicle is simply the ratio of half-track width to CG height that is commonly referred to as the *Track Width Ratio* (TWR). Although the rigid vehicle model is extremely simplified, the TWR provides a useful starting point for more complex roll models.

A similar rigid vehicle model that includes variable road inclination angle (superelevation) and tire deflections may be found in Bernard et. al [5]. Lateral tire deflections may be found from the slope of the tire overturning moment versus slip angle plot (typically the linear range of tire data is used). Bernard et. al point out that the lateral tire deflections may contribute significantly to the static roll threshold of passenger cars and utility vehicles, but not so much in the analysis of heavy commercial vehicles where the CG heights and track widths are relatively large. However, from their analysis of more complex models, Bernard et. al show that lateral tire deflections have less of an effect on the system during non-static or transient maneuvers.

III.1.2 Compliant Suspension Model

Ervin [4] continues his analysis with a slightly more complex model shown in Figure 2. Incorporating a spring suspension between the axle and the body as well as compliant tires, the vehicle is modeled as rolling in a single roll plane. The major difference

between this and the previous rigid model is that the vehicle is able to roll (non-zero ϕ measured from vertical) without all the weight shifted to one side. Note that ϕ is the roll angle of the vehicle body, or *sprung-mass*. Equation 1 still applies and we see that with a non-zero roll angle, the lateral displacement moment ($Wh\phi$) subtracts and therefore reduces the net restoring moment. This in effect reduces the roll threshold of the vehicle to

$$A_y = T/h\text{-}\phi_T \tag{2}$$

where ϕ_T is the *lift-off* angle, i.e., the maximum angle the vehicle may roll through until one side of the vehicle 'lifts off' the ground. Ervin suggests that this analysis may be extended to vehicles with more than one axle, i.e., multi-axle vehicles as long as the suspension and tire stiffness at each axle is 'uniformly proportional' to the load carried by that axle.

Therefore, though multi-axle, this is a lumped suspension model and the vehicle will still roll in a single roll plane with a maximum lateral acceleration given by Equation 2. Figure 3 is a graphical representation of this multi-axle lumped suspension model with the value of roll moment plotted on the vertical axis, stabilizing or restoring moments plotted to the right and lateral acceleration plotted to the left. The net restoring moment may be considered a combination of the suspension moment (from weight shifting in a turn causing a roll angle ϕ) and lateral displacement moment caused by lateral movement of the vehicle CG. As ϕ increases, the slope of the net restoring moment curve stays positive until it reaches the roll threshold value ϕ_T and defines the maximum lateral acceleration the vehicle may withstand before rolling over, i.e., the roll threshold. This is the point where the restoring moment (due to the suspension) saturates, i.e., is at its maximum. Past this point the net restoring moment takes the negative slope of the lateral displacement moment which is still increasing with increasing roll angle. Notice that the roll threshold $(A_y)_{max}$ is less than the roll threshold found with the rigid model, T/h.

Ervin also examines the effect of a vehicle's *roll center* – specifically the suspension roll center (RC_s) and the tire roll center (RC_t) as depicted in Figure 2. The suspension roll center is an imaginary but convenient point that the suspension may be thought of as rolling about. Another definition is a point on a vertical plane passing through wheel centers where a force may be applied without causing suspension roll [6]. Likewise, the tire roll center is another imaginary point which the axles or unsprung mass may be thought of as rolling about. Therefore, for vehicles with large suspension roll heights (h_s), such as utility vehicles or heavy trucks, the lateral displacement moment ($Wh_s\phi$) is larger per degree of roll angle ϕ. Consequently, these vehicles have reduced roll threshold levels relative to those vehicles with smaller values of suspension roll heights.

III.1.3 Effect of Superelevated Roadway

Typically, curved transitions on roadways, such as on-ramps, exit-ramps, etc., are designed with superelevations to reduce the effect of lateral acceleration on the roll response of the vehicle traversing the curve. As may be seen in Figure 4, the

superelevation has the effect of reducing or possibly even changing the sign of the lateral displacement moment (by reduction of the moment arm) thereby increasing the effective roll threshold of the vehicle [7].

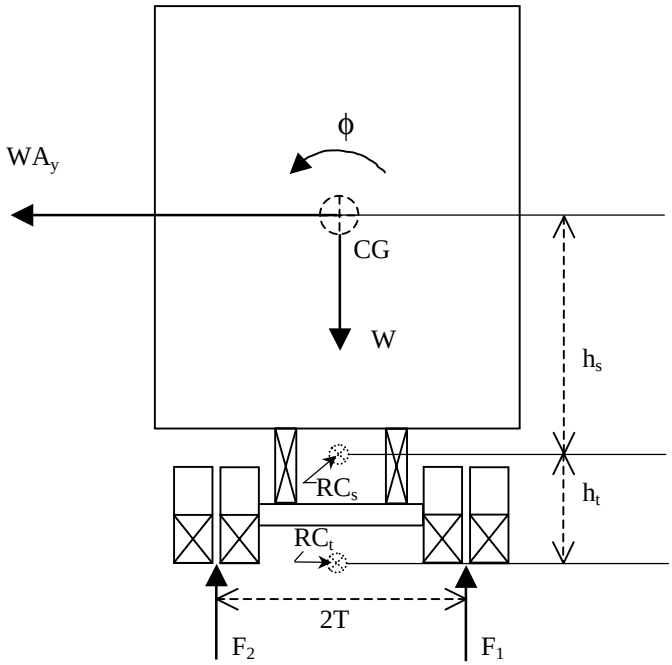


Figure 2: Vehicle model with compliant suspension and tires

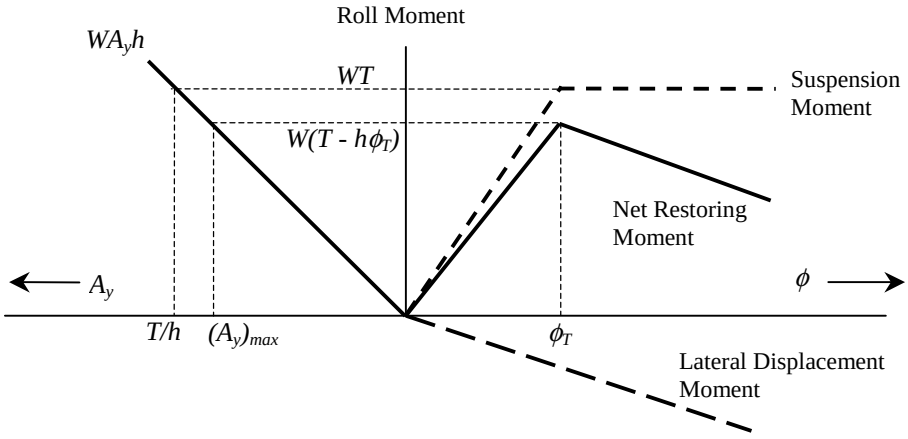


Figure 3: Roll moment diagram of lumped suspension multi-axle vehicle (adapted from [4])

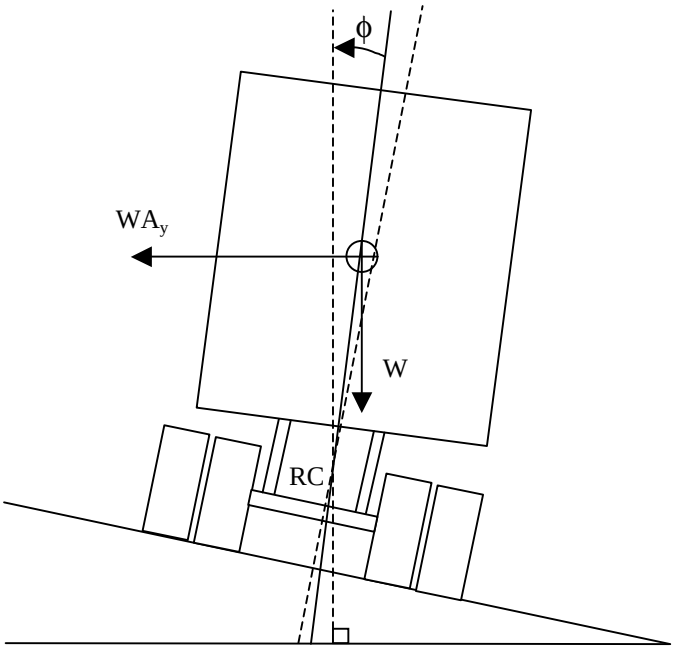


Figure 4: Effect of superelevation on lateral displacement moment (adapted from [7])

III.1.4 Non-Articulated Vehicles: Stability Metrics

Typically, passenger cars have relatively low values of suspension roll center heights and therefore high levels of roll threshold. As a matter of fact, the value of lateral acceleration that a passenger vehicle may support is governed not by the roll threshold, but by the coefficient of friction at the tire/road interface as described by Allen et. al [8]. Therefore, a passenger car will most likely slide laterally before it will roll over (assuming it is not a case of tripped rollover which will be examined later). Targeting the roll stability of passenger cars and utility vehicles and using rollover risk data obtained from an U.S. government report, Allen et. al ran a variety of statistical regressions attempting to correlate various vehicle parameters with rollover propensity. As expected, they found a strong correlation between the rollover rate and TWR. In addition, a strong correlation between the wheel base ratio (wheel base divided by CG height) and TWR was found, which indirectly implies a relation between wheel base ratio and rollover frequency. Finally, the authors conducted field tests with three vehicles: a light utility vehicle, a subcompact and 'intermediate sized' sedan. These vehicles were believed to have characteristics representative of the vehicles selected for the government report. More specifically, their levels of rollover propensity, defined by the stability metrics obtained in the first part of their research (TWR, etc.), spanned the range of the vehicles selected for the government report. The vehicles were exposed to a variety of common steady-state and transient maneuvers, including limit-performance maneuvers designed to determine vehicle characteristics in extreme situations. Computer simulation was performed and the results compared and validated with the field test data. Results of the field tests and computer simulations indicate that the TWR indeed has a severe effect on

a vehicle's roll stability. Specifically, the light-utility vehicle (which has the lowest TWR) was found to roll with two of the limit-performance maneuvers, while the passenger cars were found to be 'quite stable'. In addition, the longitudinal and lateral load-transfer ratio (LTR) was believed to have a severe effect on vehicle directional stability.

In a follow-up paper by Allen et. al [9], the range of test vehicles was increased to twelve for field-testing and simulation (the number was later increased to twenty-nine, but these additional vehicles were only for statistical analyses). The types of vehicles included in the test matrix also increased, including utility vehicles, pickups, vans, and front and rear-wheel drive passenger cars. This time, the effect of the LTR, roll stiffness distribution and throttle settings on vehicle stability during limit maneuvers were examined in more detail. In addition, side-pull tests were conducted on the twelve vehicles calculating the side force 'required for rollover'. This measure was used in conjunction with the TWR to calculate *efficiency factors*, which illustrate a particular vehicle's resistance to rollover. Results of the analysis of roll stiffness distribution indicated that percentage roll stiffness is typically shifted toward the front axle of the vehicles examined, thereby increasing understeer capabilities. This was particularly true for rear-wheel drive vehicles, which require more inherent understeer characteristics in the suspension to make up for their lack of drive traction at the front wheels. Conversely, due to the drive traction available at the front wheels and more weight (typically) on the front axles, the vehicles investigated had less of their percentage roll stiffness towards the front than the rear-wheel drive vehicles. This is because front-wheel drive vehicles require less suspension understeer characteristics.

In the same fashion, Garrott and Heydinger [10] attempted to correlate single-vehicle accident data from the state of Michigan with a wide range of parameters, both vehicular and environmental. The vehicle information included Tilt-Table Ratios (TTR) and parameters such as wheelbase length and whether or not the vehicle was front-wheel drive or rear-wheel drive. In addition, various dynamic measures such as understeer at various levels of lateral acceleration and yaw-rate steady-state gain, were included and calculated solely by computer simulation, using a wide array of steering maneuvers. The dynamic measures were what they had hoped to correlate with the vehicle accident data involving rollovers. However, results showed that six of the most important variables were environmental and static vehicle parameters and not any of the dynamic metrics they had hoped to correlate. It was concluded that none of the simulated vehicle response metrics were good predictors of rollover propensity for a particular vehicle.

Likewise, Hinch et. al [1] attempted to correlate accident data with both vehicular and environmental metrics. However, they extended the range of both the accident data (taken from five states instead of just one) and the static vehicle metrics, including the TTR, TWR (they refer to the TWR as the Static Stability Factor or SSF) and the Side Pull Ratio (SPR). Some of the environmental variables were whether the accident happened in a rural or non-rural area, whether the driver was under the influence of alcohol or drugs and whether the accident occurred on a curved roadway, among others. In addition, they attempted to correlate two other dynamic metrics designed for tripped rollover situations. These are the *Rollover Prevention Metric* (RPM) and the *Critical Sliding Velocity* (CSV). The RPM is defined as the difference between the initial

translational energy of the vehicle and the rotational energy after tripping divided by the initial translational energy (then multiplied by 100 to get a percentage). Mathematically this is defined as

$$\% \text{ RPM} = \frac{E_i - E_r}{E_i} \times 100 \tag{3}$$

where E_i is the initial kinetic energy and E_r is the rotational energy after the vehicle has been tripped (refer to the paper for complete definitions of the energy terms). An interesting result is obtained when Equation 3 is simplified. It turns out that the RPM is independent of velocity and only a function of the vehicle's mass, moment of inertia and height to the CG.

The CSV is defined as the 'minimum lateral velocity required to initiate rollover when the vehicle is in a tripping orientation'. The equation for the CSV is given in the paper and is only a function of track width, moment of inertia, CG height and the coefficient of restitution. Initially, Hinch et. al wished to correlate the CSV and RPM to the incidents of tripped rollover from the accident data of the five states. Unfortunately, the accident data was not specific enough to separate tripped rollover from the non-tripped rollover cases and the two metrics were therefore used to examine their effect (if any) on *all* the rollover accidents. Other stability metrics included wheelbase length, percentage of vehicle weight on the rear axle (typically effecting oversteer and yaw stability) and a braking stability metric which measures the deceleration of a vehicle required to lift the rear axle off the road. Using the TWR as the reference, a strong correlation was obtained between the TWR and not only the two other static stability measures (TTR and SPR), but the RPM as well. In the second phase of the research, only the Michigan data files were used for statistical correlation since it had the largest sample size and reporting methods seemed to be more consistent and detailed than other states. Hinch et. al was interested in *predicting* rollover occurrences between like vehicles using the various environmental and stability metrics. The best model, correlating with an R^2 value of 0.80, was found with the TTR, vehicle class/drive configuration, accidents per registered vehicle, whether the accident occurred in a rural area, whether the roadway was curved, whether the accident occurred on or off the road, the age of the driver, whether the driver was under the influence of drugs or alcohol and the presence or absence of anti-lock brakes.

III.1.5 Non-Articulated Vehicles: Predicting Rollover

In an attempt to predict maneuver-induced rollover, Nalecz [11] used a relation based upon kinetic and potential energies of a vehicle known as the Rollover Potential Energy Reserve (RPER). The RPER was originally developed to investigate cases of tripped rollover (see [12]) and provides an instantaneous measure of roll stability for a particular vehicle. It is defined as the difference between the potential energy of the vehicle at the

unstable or 'tip-over' position ($V_{critical}$) and the instantaneous rolling kinetic energy of the vehicle ($E_{rollingKE}$). Mathematically, this is simply

$$\text{RPER} = V_{critical} - E_{rolling\ KE}\ .$$

The RPER should be positive for non-rollover situations and become negative for rollover cases. Using his Intermediate Maneuver Induced Rollover Simulation (IMIRS) software, Nalecz investigated how changes in the various vehicle design parameters affected the RPER in both J-turn and S-turn (sinusoidal steer) maneuvers, i.e., a sensitivity analysis. Both rollover and non-rollover situations were generated for each type of maneuver resulting in a total of four tests. The tests were run at a constant vehicle speed of 40 mph and only the road friction was altered to produce the desired rollover or non-rollover conditions. The vehicle parameters used in the sensitivity analysis were segregated into seven different categories: two Geometrical sets, a Mass/Inertia set, a Stiffness set, a Damping set, an Aerodynamic set and a Tire Friction set. A summary of the sensitivity analysis and the effect on the RPER function may be found in [13]. Overall, the results showed that the RPER function was able to consistently predict maneuver induced rollover of a vehicle as well as predicting cases of tripped rollover that the function was originally derived for.

Although a following paper by Nalecz et. al [14] investigates the use of the RPER with *tripped* rollover situations, it is included here for the fact that the RPER function has been validated with full-scale testing. Two passenger cars and six light trucks, vans and utility vehicles were tested for their rollover propensity as they were slid into a curb and at other times into sand (simulating soft soil on the side of a roadway). A modified version of the RPER was introduced that neglected the translational energy term (due to CG velocity perpendicular to the roll axis) of the overall kinetic energy. Both the modified and the original RPER values were calculated for all vehicle maneuvers. Results indicated that the modified RPER was less sensitive to the severity of impact and was able to filter out some unwanted dynamic effects after impact, such as oscillations in the roll rate. These unwanted dynamics were felt to be responsible for the original RPER to falsely predict rollover when no rollover occurred; where the modified RPER always correctly predicted rollover. One benefit of the original RPER was that it was able to predict rollover *before* the modified RPER in all cases. This would be helpful in designing a rollover-warning device to alert the driver or in the design of a control system that can help stabilize the vehicle to avoid rollover before it occurs. However, Nalecz et. al recommend using the modified RPER for all rollover situations due to the consistent results obtained.

III.1.6 Articulated Vehicles: Roll Dynamics

The handling dynamics of an articulated vehicle, such as a tractor semi-trailer, differ from that of a non-articulated vehicle significantly. Among other reasons, the ability for the trailer to articulate or pivot relative to the tractor contributes an additional mass that the driver must be concerned with. In addition, the response of the trailer to inputs from the tractor, such as steering maneuvers, is typically amplified and lags behind the response of the tractor making it difficult to control. Many times this causes stability problems and a trailer may start to roll excessively or begin to jackknife (yaw instability)

before the driver is aware of the problem and therefore may not have time to take corrective action.

As described earlier, Ervin [4] derived the steady-state equations of motion for a multi-axle vehicle rolling in a single roll plane with the maximum lateral acceleration given by Equation 2. It was also noted that the analysis was only valid for vehicles with roll stiffness at each axle that are 'uniformly proportional' to the load carried by that axle. However, Ervin observes that in practice, roll stiffness may vary significantly at each axle. This implies Equation 2 is not valid for these vehicles. Therefore, he examined the effect on the vehicle of individual axles with non-uniform suspension stiffness relative to the load carried. Referring to the individual roll stiffness as R_{axle1}, R_{axle2} and R_{axle3} with $R_{axle1} < R_{axle2} < R_{axle3}$ (typically this is the case), the results are shown graphically in Figure 5 for a three-axle tractor semi-trailer. As the truck undergoes a lateral acceleration, A_y, the first wheel to lift off is the inside wheel of the trailer axle (axle 3) at a roll angle ϕ_3 of the trailer. This is the point where the rear axle produces its maximum restoring moment W_3T_3 where W_3 and T_3 are the weight and track width of axle 3, respectively. The saturation of the axle 3 restoring moment effectively reduces the slope of the net restoring moment. As A_y increases further, the inside wheel of axle 2 begins to lift off at a roll angle ϕ_2 due to its intermediate level of roll stiffness R_{axle2}. This again has the effect of reducing the net restoring moment even further and this time allows the slope to become negative. The roll angle ϕ_2 is the point of instability of the entire vehicle since the net restoring moment is unable to react to the increasing lateral acceleration and impending rollover ensues. The roll angle ϕ_2 therefore defines the roll threshold of the vehicle $(A_y)_{max}$. Finally, the suspension at the tractor axle, axle 1, saturates at a roll angle ϕ_1, reducing the slope of the net restoring moment even further. It is also clear from the analysis that treating the individual axles with separate roll stiffness reveals a roll threshold $(A_y)_{max}$ less than that when the vehicle is treated with the lumped suspension model as was done previously. Furthermore, both of the roll thresholds are less than that of the rigid model.

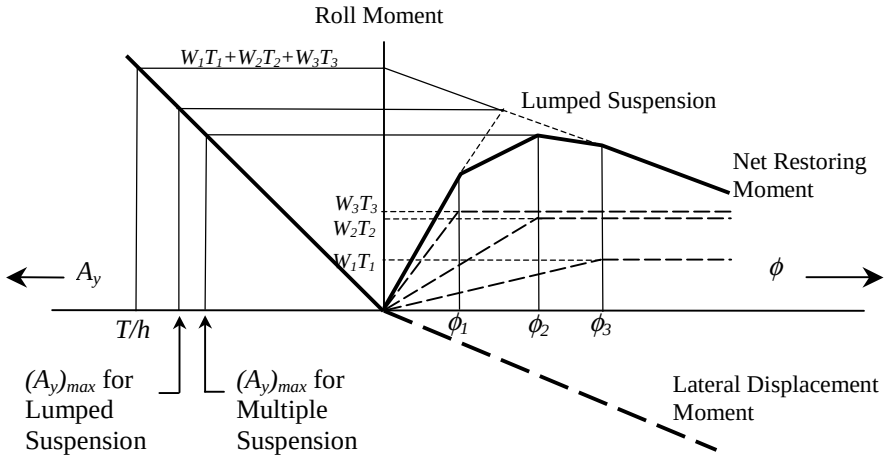


Figure 5: Roll moment diagram of multiple suspension multi-axle vehicle (adapted from [4])

The longitudinal and lateral distribution of cargo also has a profound effect on the roll stability of a tractor semi-trailer. Ervin shows that a forward shift of goods actually decreases the roll threshold of the vehicle. However, he points out that this may increase the trucks yaw stability as previous research suggests [15]. In addition, if the cargo is laterally offset relative to the centerline of the vehicle, a static moment is produced which will increase or decrease the trucks roll threshold depending on which side the load is offset on and which direction the truck is turning. For example, a truck turning to the right with a cargo offset to the left (thereby causing a static moment to the left), will increase the lateral displacement moment ($Wh\phi$), decrease the net restoring moment and therefore decrease the roll threshold of the vehicle. Conversely, if the truck makes a left-hand turn with the cargo still to the left, the static moment acts to increase the net restoring moment and therefore providing an increase in the roll threshold of the vehicle.

III.1.7 Articulated Vehicles: Stability Metrics

Static Roll Threshold

Trying to correlate rollover accident data with various vehicle parameters or stability metrics for heavy trucks is more difficult a task than with automobiles, light trucks and sport utility vehicles. This is for no other reason than the inconsistent reporting of the details in single-vehicle heavy truck rollover accidents. Specifics about the type of truck, load carried, suspension properties, etc., are not always recorded well and therefore make it difficult to correlate heavy truck parameters with accident data. However, one may expect that the TWR of an articulated heavy vehicle has the same first-order effect on the roll stability as it does with non-articulated vehicles. One may also assume a strong correlation between the TWR and the static roll threshold (SRT) as found with non-articulated vehicles. The SRT may be measured in a variety of ways including static tests such as the Tilt-Table or Side-Pull, a quasi-steady-state turning test or finally mathematically using simulation software such as the Static Roll Model ([16]; [17]) or the Yaw/Roll Model [18]. Indeed, the SRT is considered the single best measure of a particular vehicles roll stability.

The Static Roll Model, developed at the University of Michigan's Transportation Research Institute (UMTRI), is typically preferred for calculation of SRT's over other models simply because of its ease of use. Although the results are not as accurate as those that may be obtained with more complex software, such as the Yaw/Roll Model, it requires fewer input parameters and still gives a first-order approximation of the SRT. Typically, accurate input parameters are difficult and/or expensive to collect. Therefore, El-Gindy and Hosamel-deen [19] conducted a sensitivity analysis of the Static Roll Model to separate those vehicle parameters that have a significant effect on the results from those that do not. The less-significant parameters may therefore be approximated without affecting the results and eliminating many of the costs. The sensitivity analysis

consisted of analyzing a 'baseline' vehicle, a five-axle tractor semi-trailer, and adjusting the various parameters individually by a 'reasonable percentage' while keeping the others constant at their baseline value. The parameters examined are as follows:

Tractor Frame: Torsional stiffness and coulomb friction

Trailer: Combined structural and fifth-wheel stiffness

Fifth-Wheel: Lash

Tires: Vertical stiffness, lateral stiffness, and overturning stiffness

Suspension:

Tractor: Spring rates (front and rear), and lash (front and rear)

Trailer: Spring rates, lash, auxiliary roll stiffness and roll center heights

Dimensions: CG heights of sprung masses and track widths

El-Gindy and Hosamel-deen found it useful to separate the results on the sensitivity analysis into three different groups (A, B and C) of parameters. Group A consists of parameters that had virtually no effects on the results and may be eliminated from the actual model of the system. Group B contains parameters that are required in the model, but accuracy of the parameters is not very important. Finally, Group C consists of those parameters, which are not only vital to the model, but the accuracy of measured values for these parameters is very important. The results are as follows:

Group A (may be ignored in the model)

- tractor frame torsional stiffness
- tractor frame coulomb friction
- overturning stiffness of the tires
- lash in the fifth-wheel plates

Group B (important to model/accuracy of measurement *not too* important)

- tractor front spring rates
- trailer combined structural and fifth-wheel stiffness
- lateral stiffness of the tires

Group C (important to model/accuracy of measurement *very* important)

- tire vertical stiffness
- tractor rear spring rates
- trailer spring rates
- lash in tractor front and rear suspension and trailer suspension
- auxiliary roll stiffness
- roll center height
- sprung mass center of gravity height
- wheel track widths

One of the most frequently cited relationships between heavy truck rollover and SRT was developed by Ervin [20] and is shown in Figure 6. The accident data was obtained from the Bureau of Motor Carrier Safety and describes the rollover occurrence of heavy trucks in single vehicle accidents (SVA's). Using software developed at UMTRI that was a precursor to the Static Roll Model, Ervin was able to estimate a SRT for each heavy truck using the reported gross vehicle weight (GVW) and using typical values of tire, suspension and geometric parameters. It is clear from the graph that there exists a strong relationship between a heavy truck's roll threshold and rollover occurrence in an SVA. One of the most notable observations is that a fully loaded truck with a roll threshold between 0.40 and 0.45 g is at five to seven times as likely to roll over as an empty vehicle with a roll threshold of 0.65 g.

Preston-Thomas and El-Gindy [21] examined the feasibility of requiring a minimum SRT value of heavy trucks for operation in Canada and possibly all of North America. Data collected from tilt-table tests by Preston-Thomas [22] and Woodrooffe [23] is summarized for a variety of vehicles, including dump trucks, cement mixers, tractor semi-trailers and a B-train tank truck. The results support the previous suggestions for a minimum SRT ranging from 0.38 to 0.42 g, depending on the type of heavy truck. However, the authors point out the difficulty that may arise in enforcing a minimum SRT due the wide range of heavy trucks and loads and the 'power of special-interest groups'. Preston-Thomas and El-Gindy suggest that the policy of requiring a minimum SRT may be successful if started by enforcing the policy on a single type of heavy trucks, such as petroleum-hauling trucks. It is believed petroleum-hauling trucks are a practical place to start since they are all relatively similar in their tank design, typically have lower roll thresholds than that of heavy trucks and rollover of these trucks is particularly dangerous due to their hazardous cargo. In addition, it is believed that the manufacturers and operators are willing to spend money to improve safety for these vehicles.

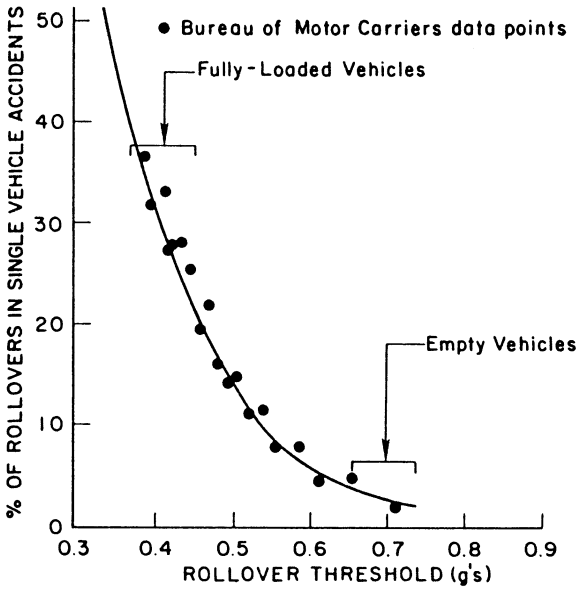


Figure 6: Rollover occurrence vs. estimated static roll threshold [20]

Roll Stability Margin

In a paper by Blue and Kulakowski [7], a dynamic rollover stability measure called the *Roll Stability Margin* (RSM) was developed to aid in the design of curve transitions on roadways and to minimize the effect of the curve transitions on the roll stability of a vehicle. The RSM takes into account the level of superelevation of the curve transition and its first-order effect on a vehicle's roll stability. As described in a previous section, superelevation of a roadway has the effect of reducing or even changing the sign of the lateral displacement moment thereby increasing the roll threshold of the vehicle. The RSM is a unitless measure and is defined as the difference between the roll threshold of the vehicle on a flat surface, ρ_0, and the *effective side friction factor*, F_{eff}, which is defined as the difference between the maximum lateral acceleration of the vehicle, A_{max}, and the superelevation at the point of maximum lateral acceleration, $E(x)_{Amax}$ (unitless), or

$$\text{RSM} = \rho_0 - F_{eff} = \rho_0 - (A_{max} - E(x)_{Amax}).$$

Therefore, the equation above may be rearranged into perhaps a more intuitive form as

$$\text{RSM} = (\rho_0 + E(x)_{Amax}) - A_{max} \qquad (4)$$

which clearly shows the positive effect on roll stability due to the superelevation of the curve. The RSM needs to be greater than zero for the vehicle to be roll stable. Note that it was assumed that the roll threshold of a vehicle increases on a 1:1 ratio with

superelevation where Blue and Kulakowski later justifies this assumption with testing and find it is actually a conservative estimate.

Blue and Kulakowski continue their investigation and define an *Acceleration Overshoot*, M_p, which indicates the smoothness of the curve transition. Curve transitions are typically not designed to minimize the effect of the sudden change in superelevation on the vehicle as it first enters the curve. This acts as a impulse to the system and can potentially lead to a rollover situation. The acceleration overshoot is a convenient measure of this effect and is defined as the ratio of the difference between the maximum and steady-state lateral acceleration to the steady-state lateral acceleration, or

$$M_p \equiv \frac{A_{diff}}{A_{ss}} = \frac{A_{diff} R}{V^2} = \frac{\left(A_{\max} R - V^2\right)}{V^2} \tag{5}$$

where R is the radius of curvature and V is the vehicle speed. They then derive a *Critical Speed*, V_{crit}, which is the maximum speed a vehicle of a given configuration (and therefore roll threshold on a flat surface, ρ_0) may traverse the curve of a particular superelevation and radius of curvature design. The critical speed is found at the point where the RSM is zero. Using Equation 5 and setting the RSM in Equation 4 to zero, the critical velocity is found to be

$$V_{crit} = \sqrt{\frac{\left[\rho_0 + E(x)_{A\max}\right]R}{1 + M_p}} .$$

At this speed the vehicle has reached its roll threshold in the turn.

A five-axle tractor semi-trailer was simulated using a 'modified version' of the Phase 4 software originally developed at UMTRI [24]. The truck was believed to represent a typical commercial vehicle with a weight of 80,000 lbs., a CG height of 90 in. and a rollover threshold on a flat curve of 0.36 g. A total of 72 simulations were performed at various speeds, types of curve transitions and design speeds, and levels of superelevation. It was concluded that the *spiral curve* [25], in which the superelevation is completely developed within the curve, allowed the smoothest transition and required the least amount of side-friction. In addition, the spiral curve had the largest critical speeds and therefore was the safest transition in terms of roll stability. In all cases, the critical speeds found using the RSM exceeded the design speeds as defined by the American Association of State Highway and Transportation Officials (AASHTO).

III.1.8 Articulated Vehicles: Factors Affecting Roll Stability

Ervin [4] examined, via simulation, the sensitivity of a heavy truck's roll threshold from variations in size and weight parameters. The goal was to examine the effect on a truck's roll threshold due to legislative changes in requirements of the size and weight variables. Specifically, the effect of changes in axle loading, gross vehicle weight, width, payload

CG height and lateral offset of the payload were examined quantitatively. A summary of the key results is as follows:

Axle Loading -- Increases in axle load limit decreases roll threshold and the percentage decrease in roll threshold is approximately in proportion to the percentage increase in axle load limit. In addition, the roll threshold is reduced as the tractor's load is shifted towards the front axle (for a fixed value of gross vehicle weight) thereby increasing the load on the tractor's steering axle. However, as the loading increases, the effect of the load distribution on the roll threshold is not as great as the payload weight or CG height.

Gross Vehicle Weight (GVW) -- In all simulations, it was clear that increases in GVW decreases roll threshold.

Vehicle Width – The width of the vehicle may be observed three different ways: the width of the trailer bed, suspension width and track width of the tires. All three were examined for their effect on the roll threshold and it was concluded that in general, increases in the widths have a very beneficial effect on the roll threshold of the vehicle.

Payload CG Height – Increases in payload CG height greatly reduces the roll threshold of the vehicle. In addition, the payload CG height does not seem to create differences of roll threshold between the front and rear trailer units of a doubles combination.

Lateral Offset of Payload CG – It was clear that increases in payload CG lateral offset significantly reduce the roll threshold of the vehicle.

Note that these results were also concluded by Harwood et. al (1989).

In a study by El-Gindy [26], various performance measures, which are used to evaluate the dynamic stability of commercial vehicles, are reviewed. Specifically, the paper reviews performance measures used originally in the Canadian Vehicle Weights and Dimension Study of 1984 (sponsored by the Roads and Transportation Association of Canada, or RTAC). The study was an attempt at reform of the various regulatory principles governing commercial vehicle transport and safety. In addition, improvements to these original RTAC measures that were subsequently suggested and some new performance measures are reviewed as well.

El-Gindy notes that some heavy vehicles can experience yaw instability during a maneuver at lateral accelerations lower than their static rollover threshold. Therefore, the typical definition of the static roll threshold, which states that the SRT is the lateral acceleration at which rollover occurs in a steady turn, can not apply to these cases of yaw divergence. He therefore proposed a new definition of the SRT as 'the maximum lateral acceleration level in g's beyond which static rollover of a vehicle occurs'. In addition, he suggests use of a 'validated static roll model' or experimental testing (e.g., tilt-table test) to calculate an SRT and avoid use of dynamic models such as the Yaw/Roll Model since it does not accurately assess the SRT of vehicles that demonstrate yaw divergence at levels below their rollover threshold.

El-Gindy also reviews two dynamic roll stability measures, the *Load-Transfer Ratio* (LTR) and *Rearward Amplification* (RWA), and a damping measure that was not included in the original set RTAC measures known as the *Yaw Damping Ratio* (YDR).

Though the YDR is not a direct measure of roll stability, it is included because it may have a substantial indirect affect on the roll response of an articulated vehicle.

Load-Transfer Ratio (LTR) – The LTR is defined as 'the ratio of the absolute value of the difference between the sum of the left wheel loads and the sum of and right wheel loads, to the sum of all the wheel loads' (Woodrooffe and El-Gindy, 1992), or

$$\text{LTR} = \frac{\left|\sum_{i=1}^{n}(Fr_i - Fl_i)\right|}{\sum_{i=1}^{n}(Fr_i + Fl_i)}$$

where Fr_i and Fl_i are the right and left vertical loads, respectively, of the *i*th axle and where *n* is the number of axles. The LTR is a useful measure of how close the vehicle is to rollover. It has values ranging from 0 to 1 with an LTR = 0 signifying the most stable state with all loads balanced and LTR =1 at rollover with all the weight shifted to one side of the vehicle. Typically, the tractor front axle is neglected in the LTR calculation due to its high roll compliance and therefore negligible effect on the roll response of the vehicle [27]. In addition, the LTR is calculated for individual units that are coupled in roll. For example, a tractor and its semi-trailer are coupled in roll and therefore considered a single unit. Likewise, a B or C-train combination is considered a single unit since the tractor, semi-trailer and full-trailer are all coupled in roll. However, a combination such as an A-train is decoupled in roll at the full-trailer and the LTR must therefore be calculated for both the tractor and semi-trailer combination as a unit and then for the full-trailer separately. For safety, El-Gindy recommends that the LTR not exceed 0.6.

Rearward Amplification (RWA) – RWA is a measure of the severity of the rearmost trailer's 'reaction' to inputs from the tractor. It is a frequency dependent measure and is defined as the ratio of the peak lateral acceleration (positive or negative) of the CG of the rearmost trailer to the amplitude of controlled lateral acceleration of 0.15 g measured at the center of the front axle of the tractor [26]. The RWA measure may be applied to single or multiple trailer configurations and it is recommended that it not exceed 2.2.

Yaw Damping Ratio (YDR) – The YDR is a measure of the rear most trailer's ability to dampen the lateral acceleration oscillations. Therefore, a trailer with a relatively small YDR may have exceptionally large oscillations resulting in peak lateral accelerations that may exceed the roll threshold of the vehicle. Though this measure was not one of the original RTAC measures, El-Gindy and others believe it is important since excessive oscillations may lead to an accident. The YDR is evaluated from a 80° pulse input at the steering wheel over a time period of 0.1 seconds and at a vehicle speed of 100 km/hr.

The response plot will be a sinusoidal decay (assuming the system is underdamped) and the YDR is simply found using the logarithmic decrement technique. Therefore,

$$\text{YDR} = \frac{\delta}{\sqrt{(2\pi)^2 + \delta^2}}$$

where

$$\delta = \ln\left(\frac{x_1}{x_2}\right)$$

and x_1, x_2 are two successive amplitudes on the response plot. El-Gindy recommends a target value of 0.15 at a vehicle speed of 100 km/h.

III.1.9 Articulated Vehicles: Anti-Roll Suspensions

As described in a previous section, rollover of automobiles from severe handling maneuvers is unlikely to occur due to the limited friction available at the tire/road interface. The vehicles are therefore more likely to slide laterally than rollover. Research into anti-roll suspensions for automobiles is extensive, but typically the reasoning behind it is less for safety than that for ride comfort or handling performance. Therefore only anti-roll suspensions for heavy vehicles will be addressed here. Ironically, little research has been conducted into anti-roll suspensions for heavy trucks though they have the greatest rollover risk compared with that of non-heavy vehicles.

The most basic anti-roll suspension typically just includes a *passive* anti-roll bar (also referred to as a stabilizer bar) like that shown in Figure 7. Points A and B are attached to the axle while the horizontal section of the anti-roll bar is attached to the body of the vehicle. As the body of the vehicle rolls through some angle ϕ, and since points A and B can not move, the anti-roll bar twists thereby creating a reaction torque M_r to help stabilize the vehicle. An *active* anti-roll suspension includes some kind of torsional actuator that can dynamically alter the overall stiffness of the system depending on control inputs from an onboard computer.

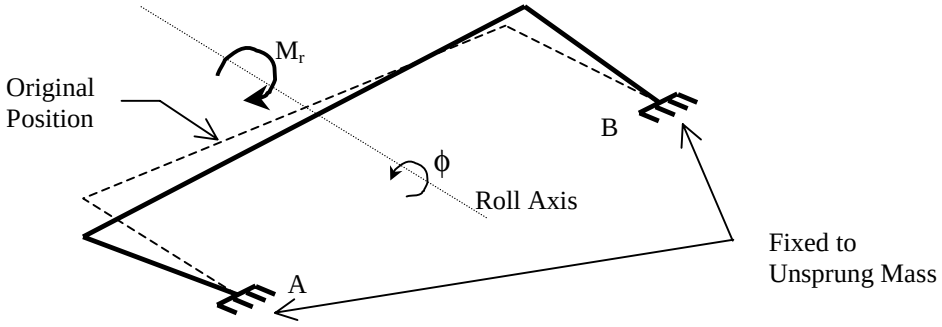


Figure 7: Typical anti-roll (stabilizer) bar

Lin et. al [28] investigate the performance of an active anti-roll suspension for heavy trucks using the Yaw/Roll Model. Initially, a simple linear model of an articulated vehicle, based on work from Segel [29], is used to develop baseline controller gains using feedback during random and step-steer inputs. The controller is then implemented into the more complex Yaw/Roll model for simulation. The results are then compared to that of two other vehicles with the same properties; one with passive suspension and one with a passive suspension and a passive anti-roll bar. The response of the trailer to a step-steer input is shown in Figure 8 for all three types of suspensions. It is evident that the active suspension is able to roll the trailer into the turn thereby increasing its roll threshold. It was concluded that an actuator with a bandwidth between 1 and 5 Hz is required and the hydraulic power supply must be able to produce 10 kW for active roll control of the trailer.

Lin et. al [30] continue their investigation of active anti-roll suspensions with optimal roll control. Specifically, they examine two control laws: lateral acceleration feedback (LAFB) using proportional and derivative gains and a controller optimized by the standard linear quadratic regulator (LQR) technique. As done in their previous work, a simple linear model [29] of a single-unit articulated vehicle was used. The gains for both the LAFB and LQR controller were tuned using stochastic input data (power spectrums) that was believed to cover the range of typical steering angle input by a driver. Results indicate that the roll threshold of the vehicle was increased by as much as 66% and lateral load-transfer was decreased 34% (for a random steering input) by use of the optimized active anti-roll bar. Figure 9 shows the vehicle response to a step steer input for the passive suspension and the LAFB and LQR active suspensions. It is evident from the plot that both active suspensions perform similarly though the LAFB controller responds slightly faster than LQR, but with a less damped response. In addition, the LQR controller is able to roll the vehicle somewhat further into the turn (reducing the roll threshold). However, the LAFB controller is preferred by Lin et. al over the LQR controller for active roll control of an articulated vehicle due to its simple transducer requirements.

In research conducted by Palkovics and El-Gindy [31], a method called Active Unilateral Braking Control (AUBC) at the tractor's rear axles was used to stabilize yaw divergence

of a five-axle tractor semi-trailer during simulations. Use of the AUBC system, which produced a yaw torque at the rear tandem axle, was able to not only minimize yaw divergence of the tractor (and reduce the risk of jackknifing), but to also help reduce roll divergence of the trailer. A 3-DOF linear model of the tractor semi-trailer [32] was initially used for optimization of the AUBC system with a standard LQR technique. The controller was then implemented onto a 34-DOF non-linear model using inputs such as force measured at the kingpin and articulation angle. It is believed that an advantage of the AUBC system is that the inputs (kingpin force and articulation angle) are readily available from current anti-lock brake systems (ABS) and the output of the controller (brake force) may be adjusted by valves controlling mean brake pressure. In other words, the system requirements are relatively simple and available as long as the truck has been implemented with ABS. Figure 10 and Figure 11 are vertical wheel load plots for a passive and an LQR optimized AUBC controlled vehicle, respectively, for a path-follow evasive maneuver conducted at 110 km/hr. The right wheels of the passive vehicle in Figure 10 clearly lose contact with the road initiating total rollover of the vehicle while the vehicle controlled by the AUBC system in Figure 11 only loses about half of its nominal load. Therefore, the active braking has a strong beneficial effect on the load-transfer variations.

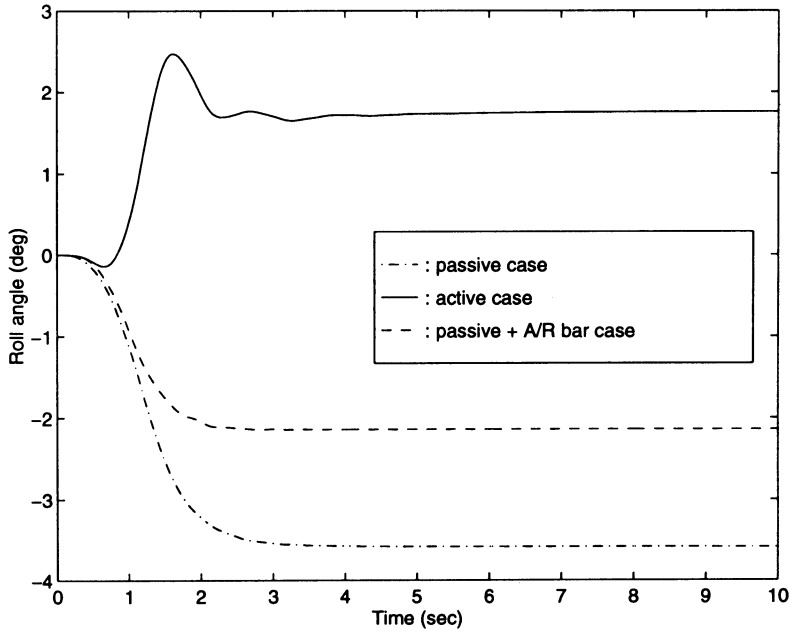


Figure 8: Roll angle responses of trailer with 3 kinds of suspension for a step input of steering angle [28]

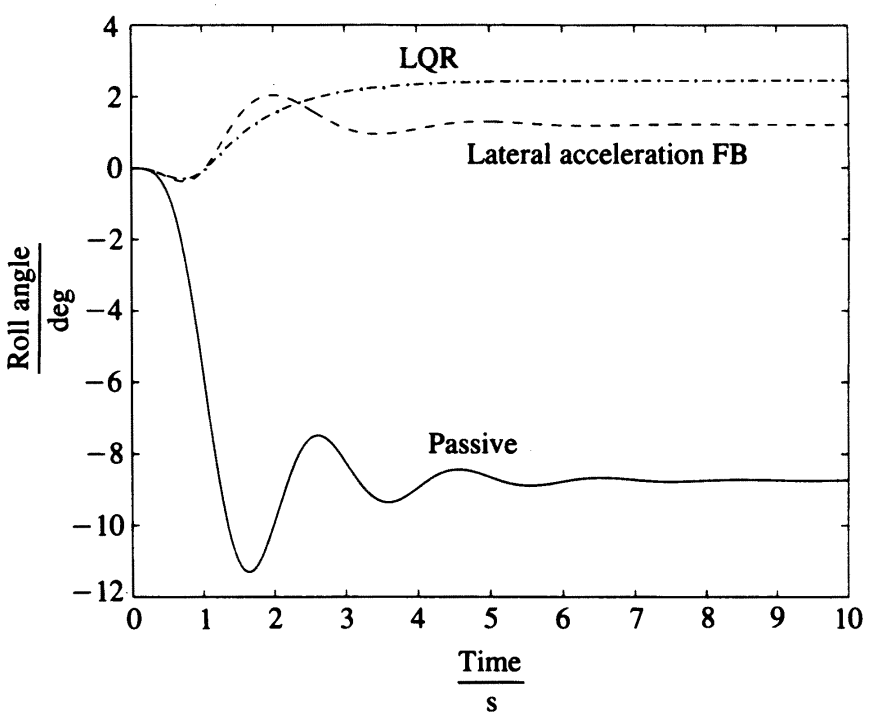


Figure 9: Roll angle responses of trailer with optimized suspensions for a step input of steering angle [30]

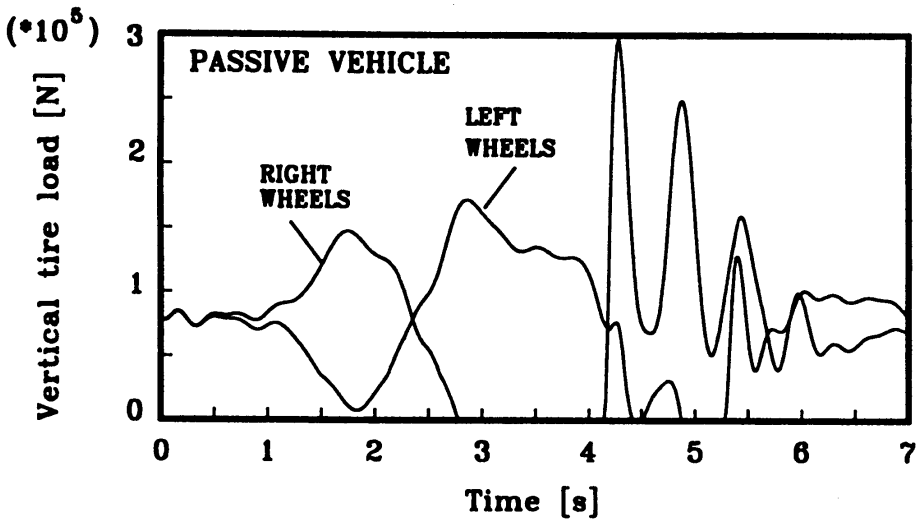


Figure 10: Vertical tire loads for a passive vehicle during an evasive maneuver at 110 km/hr [31]

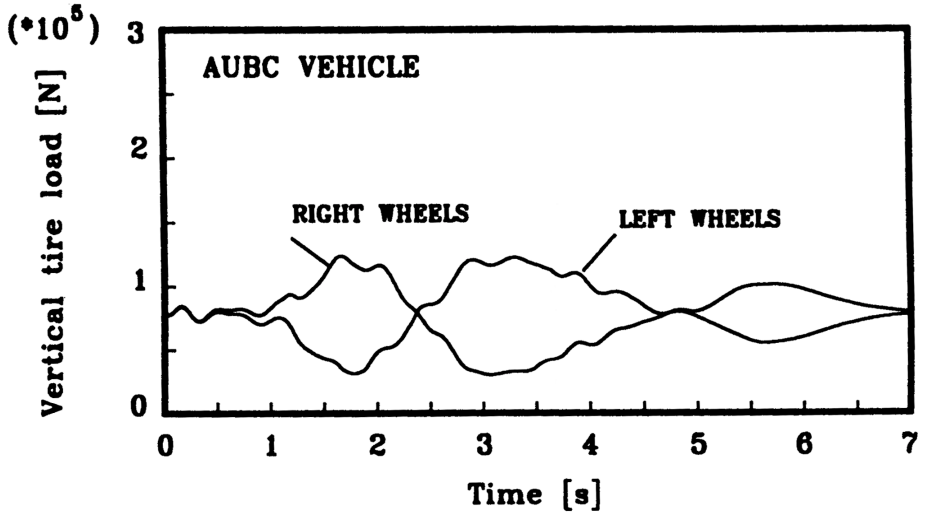


Figure 11: Vertical tire loads for an AUBC controlled vehicle during an evasive maneuver at 110 km/hr [31]

III.1.10 Articulated Vehicles: Liquid versus Rigid Cargo

Up to this point, the roll stability of vehicles with only a fixed CG position, i.e., rigid cargo has been examined. However, it should be intuitive that a moving CG position, either laterally, longitudinally, vertically or a combination of all three, can have a significant effect on the roll threshold of a vehicle. Therefore, a review of some of the research into this effect is warranted.

Ranganathan et. al [33,34] completed an extensive investigation into the effect of liquid cargo on the roll stability of the vehicle. Three different tank designs, circular, modified oval and modified square shown in Figure 12, were examined and the corresponding equations of describing CG position solved for various inclination angles of the tank. These quasi-steady-state equations for movement of liquid within the tank were then integrated into a validated steady-state roll model of the vehicle originally designed for rigid cargo [18]. The solution of the algebraic equations involved incrementally updating specific matrices to account for movement of the liquid at specific intervals. Finally, the effect of compartmentation within the tank on roll stability, which is sometimes used to minimize the lateral and longitudinal slosh [35], was investigated.

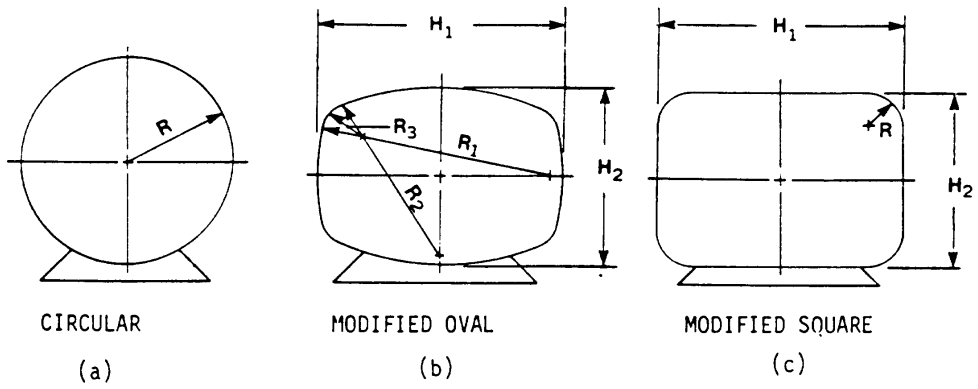


Figure 12: Tank cross-sections in general purpose transportation [33,34]

Results indicate that circular tanks are superior to the modified oval and square tanks at minimizing load-transfer due to shift of cargo CG position from changes in lateral acceleration. This is evident from Figure 13 where lateral CG translation versus lateral acceleration for a fixed tank tilt angle of 5° is plotted for various fill levels of an inviscid fluid. Likewise, the vertical translation of cargo CG was significantly higher for the modified oval and square tanks than for the circular tank.

Figure 14 is a plot of roll threshold versus percent fill level of a circular tank for liquid and equivalent rigid load. It is evident that the liquid cargo has a significant effect on the roll threshold of the vehicle due to the severe lateral translation of the CG over that for the equivalent rigid load. It was estimated that liquid cargo at a 50% fill condition could reduce the roll threshold by 0.10 g compared with that of an equivalent rigid load. Similarly, results for the modified oval and square tanks indicate a 0.15 and 0.20 g reduction in roll threshold over that for an equivalent rigid load, respectively.

Figures 15 and 16 are plots of roll threshold versus percent fill of the cargo for liquid and an equivalent rigid cargo for circular and modified oval tanks, respectively. Note that the total payload weight is fixed and loads on the composite axles are kept constant for these plots. The density of the cargo was altered to keep the weight constant though the percent fill changed. The circular tank data in Figure 15 shows that the roll threshold of the liquid filled tank vehicle is virtually unaffected by the change in percent fill. Therefore, even though the CG height of the liquid cargo is being reduced with the reduction in percent fill, the excessive lateral offset of the CG has the effect of canceling any gain in roll stability from this. As expected, the roll threshold of the rigid cargo vehicle increases with a reduction in percent fill due to a decrease CG height and the less significant effect of lateral CG translation. However, Figure 16 shows that for the modified square tank, the lateral translation of the CG outweighs the roll stability gains caused from a reduction in CG height. Similar results were obtained for the modified oval tank.

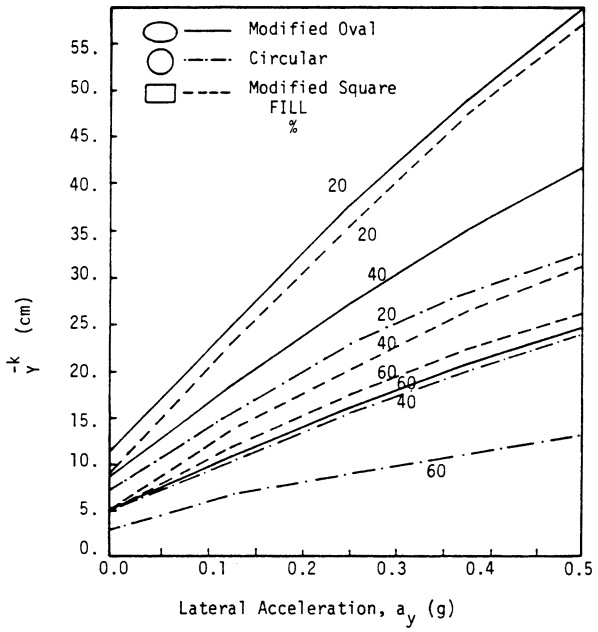


Figure 13: Lateral translation of center of mass of liquid in a compartment due to change in vehicle lateral acceleration for fixed tank tilt angle of 5° [33,34]

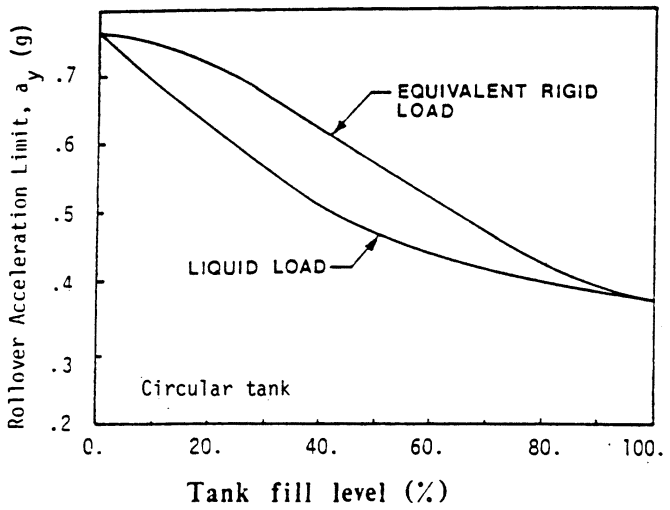


Figure 14: Comparison of rollover acceleration limits for partially filled cleanbore circular tank vehicle and equivalent rigid cargo vehicle [33,34].

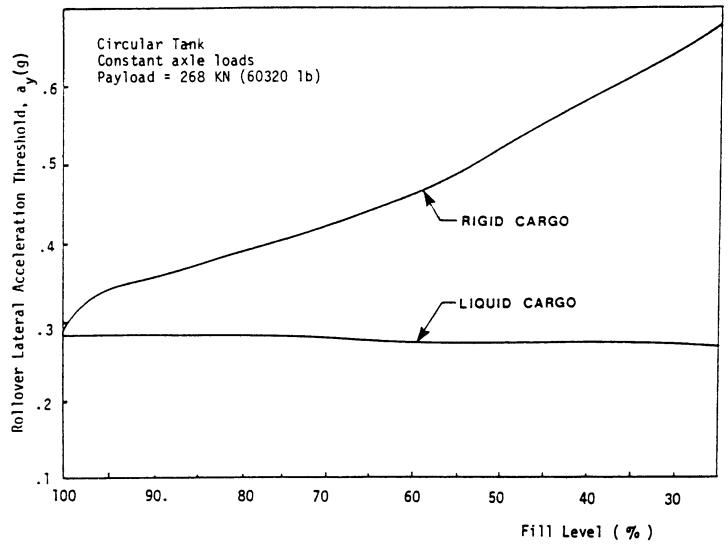


Figure 15: Comparison of rollover lateral acceleration limits of partially filled circular tank vehicle and equivalent rigid cargo vehicle when composite axle loads are held constant [33,34].

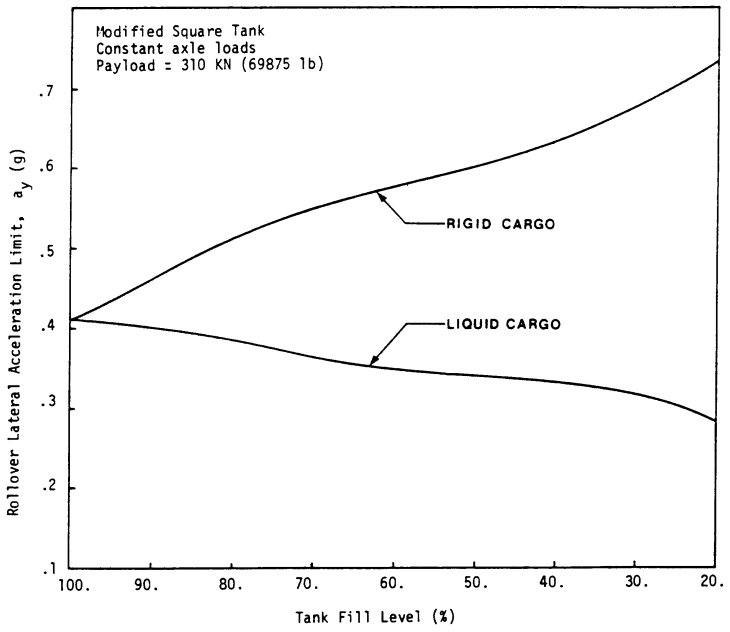


Figure 16: Comparison of rollover lateral acceleration limits of partially filled modified square tank vehicle and equivalent rigid cargo vehicle when composite axle loads are held constant [33,34].

Compartmentation of the tank, like the 4-compartment tank shown in Figure 17, was shown to have a significant effect on the roll threshold of the vehicle. Using a modified oval compartmented tank for simulation purposes, it was concluded that for a 50% fill condition, the highest rollover threshold is obtained when compartment II is partially filled. However, the exact amount of fill is not mentioned in the paper. For fill levels of about 50%, it is suggested that compartments II or III may be partially filled to generate the highest roll threshold value. Again, the amount of fill for this case is not mentioned in the paper.

Although this research involved simulation with an articulated vehicle, many of the results may possibly be extended to non-articulated tank vehicles, though further research would have to verify that proclamation. Furthermore, future research into the effect of dynamic load-transfers from movement of liquid cargo would be of interest. It is plausible that the dynamic load shifts may have even a more severe effect on the roll stability of the vehicle and the driver may have a difficult time compensating for the momentum and oscillations of the liquid cargo as it moves from side to side in the tank.

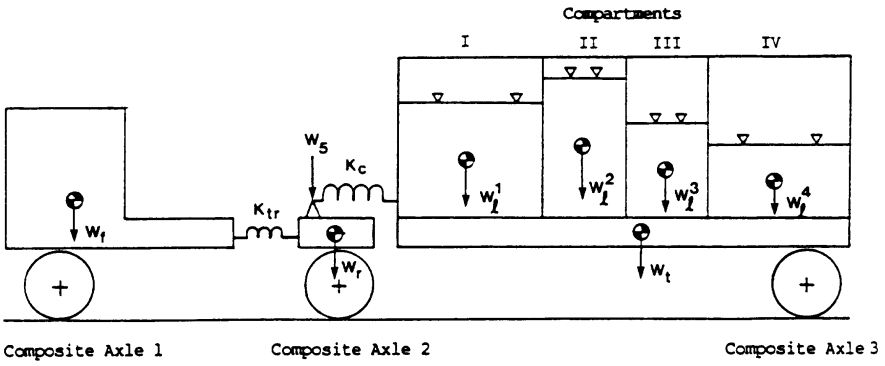


Figure 17: Representation of the tractor semi-trailer axles and sprung weights with tank compartmentation [33,34]

III.2 Warning Systems/Predicting Rollover

Obviously, without any control devices to help stabilize a vehicle in an unstable situation, it is up to the driver to take the corrective action. To do this, it is necessary that the driver receive some kind of warning of an impending rollover and in enough time for the driver to react. Therefore, the warning device must be able to dynamically predict a rollover situation quickly and correctly. The correctness of the prediction may be examined a couple of different ways. If the warning system fails to signal the driver in a rollover situation, i.e., a false negative, catastrophic results may occur. On the other hand, if the warning system alerts the driver to an imminent rollover though the vehicle is in no real

danger of rollover, i.e., a false positive, then the driver may feel that the warning system is too liberal at giving a warning and may opt to ignore one in the future where the danger of rollover is real. Therefore, it is clear that a rollover-warning system must be able to predict *all* situations of imminent rollover and give few, if any, false warnings to ensure that the warnings are taken seriously at all times. In addition, a common observation among researchers is that for a warning device to be feasible for heavy articulated vehicles, it would need to be as insensitive as possible to the many different parameters inherent with commercial trucking, such as payload CG heights and suspension properties, among others.

Rakheja and Piché [36] attempted to establish a set of rollover stability criteria for an articulated vehicle for use as input parameters to a rollover-warning device. In addition, they investigated the phenomenon of jackknife instability and aimed at developing a similar set of stability criteria for use in a jackknife-warning device. However, although jackknifing is an unstable condition of an articulated vehicle that sometimes leads to rollover, this review is limited to pure cases of rollover and will therefore not address instability due to jackknifing further. The authors note that previous research has pinned the onset of rollover instability on a variety of dynamic factors including high levels of lateral acceleration, rapid increases in roll angle, roll frequency of the trailer approaching its resonant roll frequency and excessive spring deflection in the suspension. However, they also point out that many of these parameters are 'extremely sensitive to vehicle design and operational factors' and require significant instrumentation to measure them. The procedure for developing a warning device is therefore two-fold:

1) Identification of vital motion cues or dynamic response parameters related to onset of vehicle instabilities.

2) System development including on-line acquisition of motion cues and generation of warning to the driver.

In addition, the dynamic response parameters from step 1 must be 'relatively insensitive to variations in design and operating factors' and 'directly measurable'. Note, Rakheja and Piché attempt only to complete step 1 for this paper.

To identify the key response variables and using common values of vehicle parameters such as weights, dimensions, tires, suspension, etc. for a five-axle tractor semi-trailer, simulations were run using various computer models. Furthermore, they note that rollover may occur during simple steady-state (static) cornering or from high-speed dynamic maneuvers such as a lane-change or obstacle avoidance. Therefore, both types of maneuvers (static and dynamic) must be examined to establish the key response variables.

III.2.1 Steady-State Cornering

Since the static roll threshold (SRT) of a vehicle is a good measure of a vehicle's roll threshold during steady-state cornering, the Static Roll Model was used. The SRT's of 72 different vehicle configurations were analyzed to decide which parameters most affected the SRT. Using the TWR of a rigid vehicle (T/h) and the calculated SRT's for the 72 configurations, Rakheja and Piché developed a list of 'compliance factors'. These

compliance factors are a convenient measure of the reduction in the TWR for a rigid vehicle due to the inclusion of compliance in the vehicle model. They are calculated simply by the ratio of the SRT to the TWR as shown in Equation 6.

$$C = \frac{SRT}{TWR} = \frac{SRT}{T/h} \tag{6}$$

Values for the compliance factors ranged from 0.64 to 0.74 and 0.68 to 0.75 for the 2.44 and 2.59-m wide vehicles, respectively. In addition, they concluded that the SRT calculated with the Static Roll Model is solely dependent on CG height for a given track width and was relatively insensitive to various vehicle parameters such as suspension properties, tractor frame torsional stiffness, articulation mechanisms and trailer structure. Although most of these results agree with previous conclusions of the sensitivity analysis performed on the Static Roll Model by El-Gindy and Hosamel-deen [19], their opinion on the sensitivity to suspension properties differs strongly. Specifically, they found the suspension properties of the trailer unit to have a first-order effect on the roll stability of the vehicle. Nevertheless, Rakheja and Piché suggest two plans that can be used to detect the onset of roll instability during steady-state cornering; one when the CG height is known and one when it is not.

Known Center of Gravity Height

Rakheja and Piché suggest that the impending rollover is quite easy to detect when the CG height is known. From their results it was concluded that the compliance factors were relatively unaffected by suspension type and CG height. Therefore the mean and standard deviation of the compliance factors was calculated with values of 0.72 and 0.03, respectively. A *threshold compliance factor*, C_t, was subsequently developed and calculated from

$$C_t = \frac{C_m - 3\sigma}{\alpha}$$

where C_m is the mean compliance factor, σ is the standard deviation and α is a safety factor. As an example, a threshold compliance factor of 0.57 is realized using a safety factor of 1.1. Finally, using the known values of track width and CG height, a ‘safe value’ of lateral acceleration may be calculated from Equation 6 with $C=C_t$.

Unknown Center of Gravity Height

Rakheja and Piché point out that from previous studies [37], the CG height of a fully laden trailer lies somewhere around 2 meters. Therefore, using the compliance factor developed for a trailer with a CG height of 2.03 m as in their study, a ‘safe limit’ of lateral acceleration may be calculated from

$$A_{safe} = \frac{C_{2.03} \times TWR}{\alpha} = \frac{C_{2.03} \times T/h}{\alpha}$$

where α is the safety factor and $C_{2.03}$ is the compliance factor for a vehicle with a CG height of 2.03 m (possibly the average of the 2.03 m CG height compliance factors for all types of suspensions computed in the paper). Consequently, monitoring of the lateral acceleration of the trailer is considered a good indicator of impending rollover during steady-state cornering.

III.2.2 High Speed Directional Maneuver

As noted previously, rollover of a heavy vehicle in a dynamic maneuver may occur at lateral acceleration levels significantly below that of the SRT. Therefore, a separate set of *dynamic* response parameters must be found for the warning device to accurately predict roll instability. The Yaw/Roll Model was used to examine the response of the various vehicle configurations described before at 100 km/hr in both a severe lane-change maneuver and in an obstacle avoidance maneuver. As with steady-state cornering (and the definition of the SRT), wheel lift-off defines the onset of rollover for dynamic maneuvers as well. Therefore, Rakheja and Piché again investigate the wheel loads as a possible indicator. However, wheel loads are difficult to measure directly and as an alternative, suspension loads are examined. Using the 2.03-m CG height, simulations are performed for different types of suspensions at the semi-trailer axles. Results indicate that the suspension load measurement is a good indicator of impending rollover for mechanical spring suspensions, but not for air spring suspensions. Therefore, since the response measure needs to be common for all types of suspensions, it is concluded that the measurement of suspension loads 'does not provide a reliable indication of the tire lift-off and thus the onset of vehicle rollover'. Another indirect measure of tire loads is the axle roll angle which may be measured relatively easily and in real-time. Simulations are performed for various CG heights and trailer widths in both lane-change and obstacle avoidance maneuvers. It is concluded that the axle roll angle may be used as an indicator of imminent rollover. In addition, and probably most importantly, additional simulations suggest that it is possible to stabilize the vehicle with a corrective steering maneuver once the axle roll angle indicates a state of rollover. Specific results indicate safe values (using a safety factor of 1.1) for axle roll angles for 2.44 and 2.59-m wide trailers as approximately 1.4 and 1.3 degrees, respectively.

In an investigation into the frequency a heavy vehicle approaches its roll threshold in actual service, George [38] performed extensive research with five different tractor semi-trailers on loan from various transport companies in Australia. All important weights and dimensions of the vehicles were measured. In addition, a tilt-table was used to measure separate SRT's of the tractor and the trailer and to also estimate the CG height of each trailer. The five tractor semi-trailers were then rigged with data collection equipment to measure in real-time the lateral acceleration and yaw rate of the tractor (measured in the cabin) and lateral acceleration at the rear of the trailer (at the chassis level). Additionally, all five vehicles were equipped with strain link transducers to measure lateral load-transfer. All vehicles were then put back into their normal operations, though the drivers

were instructed to stay off the highways as much as possible to increase the amount of data generated from use of back roads. The software TruckDas, an 'event triggered' package developed by the Australian Road Research Board Ltd. (ARRB), was used to collect and process all the data. The event trigger to start logging data was the lateral acceleration of the tractor and the level was set to 40% of the vehicle's measured SRT. Once triggered (an 'event'), TruckDas recorded the time and magnitude of lateral acceleration separately for the tractor and the trailer from a zero or steady-state level to their peak lateral acceleration. In addition, TruckDas recorded the mean steer path information, estimated road slope, vehicle speed and the time and distance measured from when the vehicle left the terminal.

Results from the 268 events indicated that the vehicles were operated 'most of the time' at a level of 52% of the individual truck's roll threshold. Figure 18 is a histogram of the normalized roll limit (ratio of lateral acceleration measured to the roll threshold of the vehicle) versus number of occurrences. George notes, however, that the results are skewed due to the fact that data for lateral acceleration levels less than 40% of the SRT are not recorded (because of the event trigger set at 40% or the SRT). However, it is not clear to this author why there appears data below 40%-normalized roll-limit in Figure 18. Nevertheless, he proceeds to calculate the 'preferred' level of lateral acceleration from knowledge of the individual lateral acceleration levels and corresponding vehicle speeds with the relation

$$A_c = \frac{V^2}{R}$$

where A_c is the centripetal acceleration, V is the speed and R is the radius of curvature of the turn. The radius of curvature was calculated for each level of lateral acceleration and vehicle speed, and then plotted versus the square of the speed as shown in Figure 19. Analyzing the data as a whole, it reveals the vehicle speed and steer maneuver producing a lateral acceleration that the drivers feel most comfortable with. A linear regression was applied to the data and resulted in a 'preferred' lateral acceleration level of 1.42 m/s^2, or 0.145 g at the tractor. Furthermore, in order to determine the amount of time after a warning of impending rollover that a driver had to take corrective action, the maximum lateral acceleration attained and the time for each of the events to occur was plotted and is shown in Figure 20. Again, it is not clear why there appears data below the 40% roll threshold trigger level. As may be seen in the plot, the data above 75% of the roll-limit has been separated into three distinct categories: those peak accelerations occurring within 2 seconds from a zero or steady-state value ('A'), between 2 and 6 seconds ('B'), and those peaks occurring in greater than 6 seconds ('C'). It is clear from Figure 20 that most events over the 75% roll-limit occurred within 2 to 6 seconds (section 'B') of a zero or steady-state level of lateral acceleration. In addition, eleven events out of the 268 recorded exceeded the 75% roll-limit, where five events exceeded 80%. George notes that although this may seem fairly infrequent, this means that on average a heavy truck comes to within 20% of its roll threshold once every 715 km the vehicle travels. Furthermore, if these numbers are an average representation of the Australian trucking

fleet as a whole (using data from the Australian Bureau of Statistics, 1988), each heavy truck will approach its roll-limit approximately 100 times a year.

Analysis was further carried out to examine the possibility of using the 75% of the roll-limit as the trigger to warn the driver, i.e., when the lateral acceleration reaches 75% of the roll-limit during a particular maneuver, a signal is sent to the driver to warn of impending rollover. Therefore, for each of the eleven events recorded that exceeded the 75% roll-limit, the individual slopes of the acceleration/time curves like that in Figure 21 was calculated. The curve was then extended until it reached the roll threshold for that particular vehicle. The time between the occurrence of peak lateral acceleration and estimated time that the roll threshold was met was recorded. In other words, this is the time from when the lateral acceleration peaked during the experiment to when it would cross the roll threshold for that particular vehicle, assuming the lateral acceleration keeps increasing at the same rate. However, it is not clear exactly how George calculated the slope from Figure 21, though it appears he used the peak acceleration value and the one preceding it. Finally, a relationship between the time to warn the driver and the level of lateral acceleration was obtained as shown in Figure 22. It is clear from the plot that according to this research, the more severe the maneuver that creates high levels of lateral acceleration, the less time there is to warn the driver. A hypothesis that may be drawn

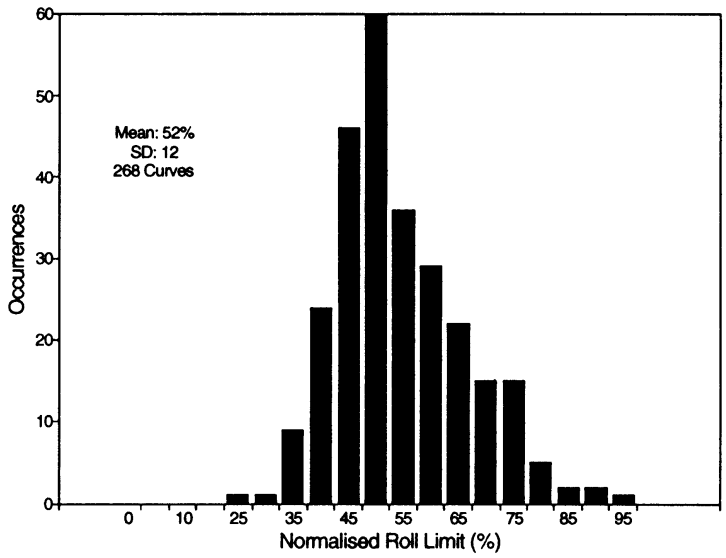


Figure 18: Test vehicles distribution to the roll-limit [38]

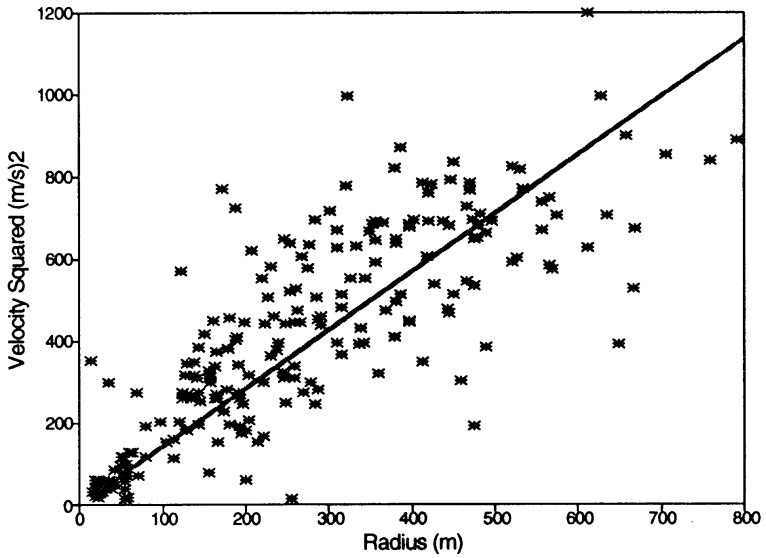


Figure 19: Driver's preferred choice of vehicle speed and steer path [38]

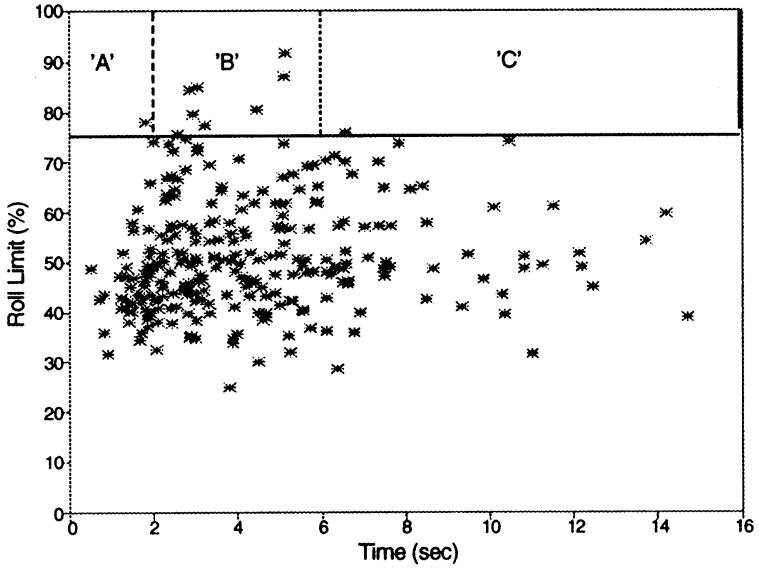


Figure 20: Critical recorded times to reach maximum lateral acceleration from zero to steady-state acceleration [38]

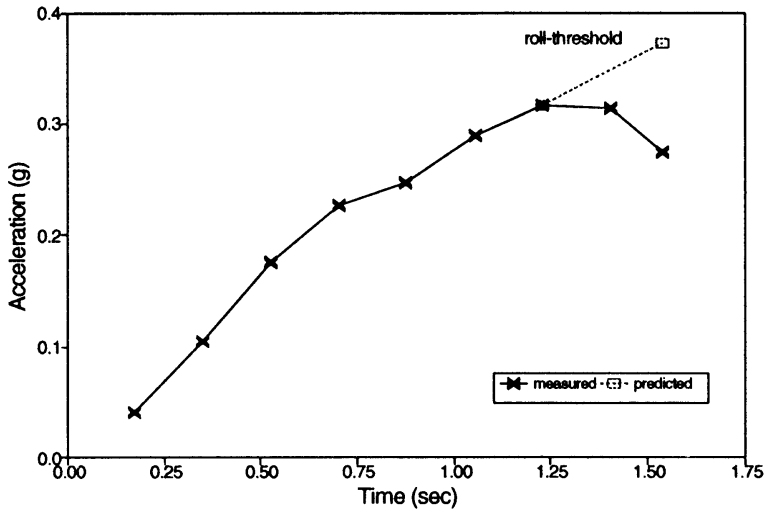


Figure 21: Method of predicting the time to the roll-threshold [38]

from this is that drivers do not intentionally approach the roll threshold of their vehicles since the occurrences happen so quickly and may be caused by unexpected maneuvers such as accident avoidance. For example, Figure 22 shows that the event that reached 92% of the roll-limit had approximately a 0.25 second window before the vehicle reached 100% (rollover). This short amount of time is due to the high slope (rate of change of lateral acceleration) found from Figure 21 for this particular event.

Consequently, the driver approached the 92% roll-limit point very quickly and therefore probably unintentionally. Conversely, if we examine one of the events that peaked at about 75% of the roll-limit, the slope from Figure 21 would be much less and therefore the time window to complete rollover has a value of approximately 3 seconds. In this case, the driver slowly approached the 75% roll-limit of the vehicle, and then made a corrective action to reduce the lateral acceleration, probably all the while feeling in complete control. It should be stated, however, that George makes some general conclusions from Figure 22 that this author believes are somewhat questionable. Specifically, George states that 'providing a warning signal when a vehicle is at 75% of its roll-limit would allow a time period greater than 3 seconds for the driver to act'. This cannot be deduced merely from Figure 22. This plot simply shows that for the vehicles that reached 75% of the roll-limit, *but increased no further*, there is approximately 3 seconds to warn the driver. However, for the vehicle that reached 92% of its roll-limit, nothing can be concluded from Figure 22 about the time between when the vehicle reached the 75% roll-limit point and the 92% roll-limit point. The slope of the curve like that in Figure 21 would need to be determined for the event that reached the 92% roll-limit to estimate the amount of warning time available. Therefore, to state that there is 3 seconds available to warn the driver when the vehicles reach 75% of their roll-limit does not seem justifiable based solely on the data presented in Figure 22. Perhaps a plot of the time between when the vehicle reaches the 75% roll-limit point and the peak lateral

acceleration for the individual vehicles versus the percent roll-limit would be useful in determining the time available to warn the driver. Though this author believes some of the conclusions made are not entirely accurate, George has clearly shown the importance of the *rate* that the lateral acceleration changes on predicting impending rollover.

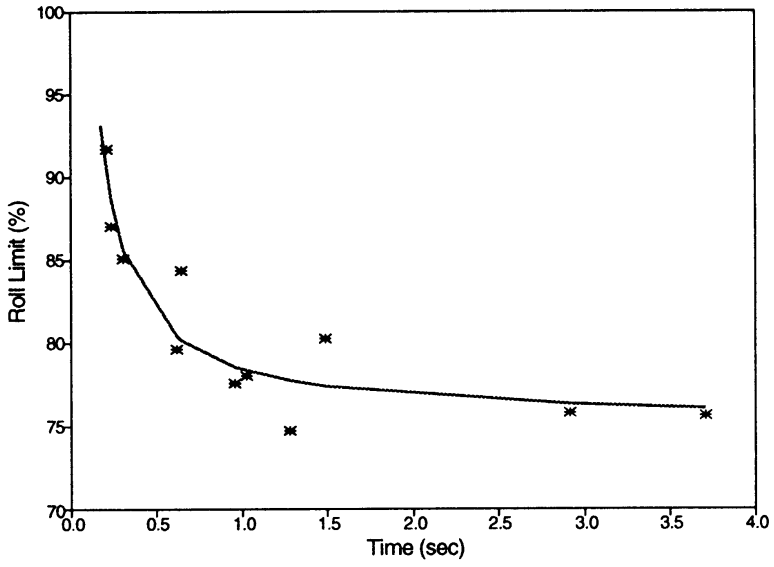


Figure 22: Relationship of the predicted time from the percentage of the roll-limit to the roll-threshold [38]

In a study by Liu et. al [39] an alternate version of the Load-Transfer Ratio, called the Roll Safety Factor (RSF), is presented as a possible stability measure for predicting rollover. However, the RSF only differs from the LTR version in Ervin and Guy [40] in that no absolute value is taken, i.e.,

$$\text{RSF} = \frac{\sum_{i=1}^{n}(Fr_i - Fl_i)}{\sum_{i=1}^{n}(Fr_i + Fl_i)}$$

where i is the axle number and n is the number of axles. As stated previously, the LTR, or RSF in this case, typically neglects the contribution from the front tractor axle.

Liu et. al specify the RSF for this case as 'RSF_s' where, defining the front tractor axle as *axle 1*,

$$RSF_s = \frac{\sum_{i=2}^{n}(Fr_i - Fl_i)}{\sum_{i=2}^{n}(Fr_i + Fl_i)}$$

Note that the RSF (and RSF_s) may take on values between –1 and 1, giving not only the magnitude of relative instability, but also the direction the vehicle is rolling.

Using a single-axle analytical model, they were able to identify which dynamic variables were directly related to the RSF. Specifically, the RSF was related to the axle roll angle (ϕ_u), the sprung mass roll angle (ϕ_s), the lateral acceleration of the sprung mass (A_y), and a parameter frequently referred to as the *steering factor* which is the product of the front wheel steering angle (δ_f), and the square of the vehicle forward velocity (V). Note the steering factor derived from the steady-state cornering equation for a two-axle vehicle is

$$\delta_f = \frac{L}{R} + \frac{K_u A_y}{g} \tag{7}$$

where L is the wheelbase of the vehicle, R is the radius of curvature of the turn, K_u is the understeer coefficient, A_y is the lateral acceleration at the CG of the vehicle and g is the acceleration due to gravity. From the equation for centripetal acceleration, $A_y = V^2/R$, we may substitute for R in Equation 7 and rearrange,

$$\delta_f V^2 = A_y\left(L + \frac{K_u V^2}{g}\right)$$

giving us the steering factor on the left-hand side. Relating these variables to the RSF through the simple vehicle model enabled them to investigate how dependent the variables are on the various vehicle parameters and their level of measurability. Liu et. al also assesses reliability by how dependent or independent that indicator is on the various vehicle design parameters, where independence is desired. Table 1 lists the results of their analysis in order of their reliability from best to worst. It is clear that the RSF is independent of all listed design parameters and therefore the most reliable of the indicators. However, load-transfer (and therefore the RSF) is typically difficult to measure resulting in a 'poor' measurability rating. On the other hand, the axle roll angle is typically easy to measure but depends on vehicle parameters such as weight, track width and tire properties. Likewise, the lateral acceleration and sprung mass roll angle are relatively easy to measure, but depend on CG height and suspension properties in addition to those that the axle roll angle depends upon. Finally, the steering factor is also

relatively easy to measure but depends upon all the listed design parameters making it the least reliable rollover indicator. Using the Yaw/Roll Model and a 'typical' five-axle tractor semi-trailer, they conducted a more thorough sensitivity analysis between the RSF_s and a more complete set of rollover indicators. The set of rollover indicators included separate tractor and trailer lateral accelerations, tractor and trailer roll angles, axle roll angles and the steering factor. It was also of interest to examine the phase, or lag/lead time, of the various indicators in relation to the RSF_s, since knowledge of the phase indicates the predictive power of the indicator on an impending rollover situation. Three different maneuvers were simulated for the vehicle including a steady-state cornering maneuver and two transient ones: trapezoidal steering and sinusoidal steering. In general, results indicated an inverse relationship with predictive power, i.e. large lead times, and strength of correlation with the RSF_s for all types of simulated maneuvers. For example, one of the stronger correlations existed between the trailer lateral acceleration and the RSF_s, but since the two were nearly in phase with one another, no lead-time was present. On the other hand, the steering factor had the largest lead-time (on the order of 0.7 seconds) but had the worst correlation with the RSF_s. It was concluded that none of the proposed indicators by themselves were good candidates for use with the RSF_s in an early warning device, but perhaps a combination of the indicators may be used successfully.

Table 1: Dependency of rollover indicators on vehicle design parameters (adapted from [39])

	Vehicle Design Parameters						
Rollover Indicator	Vehicle Weight	Track Width	Tire Properties	CG Height	Suspension Properties	Handling Properties	Reliability/ Measurability
RSF							1 / Poor
ϕ_u	√	√	√				2 / Good
A_y	√	√	√	√	√		3 / Good
ϕ_s	√	√	√	√	√		3 / Good
$\delta_f V^2$	√	√	√	√	√	√	4 / Good

Preston-Thomas and Woodrooffe [41] conducted a feasibility study on the production of an on-board rollover-warning device for heavy trucks. Using previous data from a feasibility study performed by Sparks and Berthelot [42], it was estimated that a warning device would need to cost at most $633 for the device to be cost effective if based on the economic factors alone. Note that this price was based on the economic benefits gained from the eliminating all 'preventable' and 25% of the 'potentially preventable' heavy truck rollover accidents as defined by Sparks and Berthelot. Preston-Thomas and Woodrooffe conclude that a rollover-warning device using the LTR as the sole indicator of incipient rollover was economically feasible if and only if incorporated with an existing on-board weigh scale system. The cost of on-board weigh scale systems at the time of the report ranged from $4000 to $6000 and they believe that the incremental cost of a warning device incorporated with one would be small.

III.2.3 Existing Commercial Warning Devices

An existing commercial warning device called the Stabe-Alert: Stability Monitoring and Alarm System, developed by an apparently now defunct company called Roadway Safety Systems, Inc., is presented in the feasibility report conducted by Preston-Thomas and Woodrooffe [41]. The Stabe-Alert system uses speed sensors and a strain-gauged disk load sensor [43] for measurement of load-transfer. Neither the specific functionality nor the degree of success of the device is mentioned in the report and the author could find no further information. However, the report did describe the need for future research including full scale testing of the device on a suitable tractor semi-trailer. This would imply that the device had not used in actual service yet and development may be relatively preliminary.

Another warning device, called the LGAlert™ and developed by Stability Dynamics Ltd., is described in a report [44] submitted to the Dallas-Fort Worth Airport. LGAlert™ was developed for use with airport fire trucks and uses lateral acceleration, measured simultaneously at six different locations on the vehicle, as its sole input to determine a warning level. The device then uses a look-up table containing values of maximum tilt-angle experimentally measured using a tilt-table test. As described in the report, the testing of the device was performed exclusively to determine the ideal locations of the six lateral accelerometers and not to access the rollover performance of the vehicles or the effectiveness of the device in warning the drivers.

Finally, another commercial device developed in the Netherlands at the Vehicle Dynamics Department of TNO Road-Vehicle Research Institute is called the Tilt Monitoring System [45]. Specifically developed for heavy articulated vehicles with pneumatic suspension systems, the device measures and uses a combination of lateral acceleration and pressure in the suspension (for measuring load) to make a decision on the maximum lateral acceleration the vehicle may undergo. If the device detects an unstable or dangerous situation, a warning to the driver is provided. Furthermore, the Tilt Monitoring System records information about the vehicle's use during a trip such as distance and speed, which may then be used for post analysis. Though the specifics of how the device actually uses the measured information (e.g. computing load-transfer) nor its effectiveness as a warning device are known to the author, the System has been used successfully by the BK-Gas Corporation to assess dangers with their in-service vehicles and analyze the performance of their drivers.

III.3 Conclusions based on Literature Review

Based on the work performed by Liu et. al [39] and summarized in Table 1, it appears that the RSF would provide an ideal measure of the instantaneous roll stability of the vehicle. However, as also pointed out by Liu et. al, the RSF is very difficult to measure in practice due to the physical limitation of placing load sensors in appropriate positions (tire/road interface). A solution to this problem would be to indirectly measure or predict wheel loads based upon other dynamic information about the vehicle that is more easily and accurately measured in practice. Intelligent systems, such as Artificial Neural Networks, have the ability to model highly nonlinear systems quite easily and accurately,

and therefore may have the ability to provide a solution to the problem of determining wheel loads for road vehicles during dynamic maneuvers. Determining the validity of this postulate is considered one of the most important features of this research.

If an Artificial Neural Network is able to provide an accurate measure of wheel loads based upon other (more easily measured) dynamic properties of the vehicle, the RSF (or LTR) may then be computed. However, for a RWD to have the ability to actually *warn* the driver, it would need to be more than just the instantaneous stability level. Perhaps a measure of the rate at which the RSF is changing would be useful, or the instantaneous position of the wheel angles, to name a few. A human driving the vehicle uses many different inputs to make decisions on corrective action, but trying to model mathematically how a human would or should react to a potential roll-unstable condition would be quite difficult. The idea behind another intelligent system, a so-called Fuzzy Logic Inference System, can be developed to specifically address that problem. Fuzzy Logic has the ability to provide crisp mathematical answers to problems using a system based upon the way humans solve problems, i.e., using their experience and 'common sense'. A relatively detailed description of both Artificial Neural Networks and Fuzzy Logic may be found in Appendix A and B, respectively.

IV. Tools

The tools used for the research in this proposal consisted of two software packages, TruckSim and Matlab, and a solid-state gyroscope for measurement of dynamic variables.

IV.1 TruckSim

As described by the TruckSim User Manual [46],

> 'TruckSim is a software package for simulating and analyzing the behavior of trucks, buses, and articulated vehicles in response to steering, braking, and acceleration inputs. It produces the same kinds of outputs that might be measured with physical tests involving instrumented vehicles.'

The kernel for TruckSim is based upon the software engine developed for the CarSim simulation package [47] which fully solves 3-dimensional nonlinear rigid body motion. The major nonlinear kinematical and compliances of the steering and suspension system are integrated into the model via look-up tables and is based upon experimentally measured data. Furthermore, the tire model is implemented via nonlinear tabular functions and uses Pacejka's Magic Formula [48] for necessary modifications to the data. Finally, the model has been validated in numerous studies (e.g., [49]).

IV.2 Matlab

Matlab is a high-performance software development package used extensively in industry for all types of research involving numerical computation. Two components or *toolboxes* of Matlab were used for this research: The Neural Network Toolbox and the Fuzzy Logic Toolbox. Both were used for all ANN and FLIS modeling, development and testing.

IV.3 VG300CB Solid-state Gyro

The VG300CB solid-state gyro (also known as a Dynamic Measurement Unit, or DMU) was developed by Crossbow technologies and offers an extremely convenient and accurate way to dynamic measurement data. Shown in Figure 23, the DMU provides pitch and roll angle measurements, along with three angular rates and three linear acceleration measurements with an interface that connects directly to the RS-232 (serial) port of a PC. The user has the option of using analog outputs, digital outputs, or both.

Internal to the DMU are three independent linear accelerometers and silicon micro-electro-mechanical devices, or MEMS that use the Coriolis force and its effect on vibratory structures during rotation to provide a measurement of angular rates. The DMU integrates the angular rate data to provide roll and pitch measurements. It is not clear why the DMU does not provide yaw angle measurement.

The DMU requires a 10-15V DC power supply and interface cable. Software is included with the DMU that provides a convenient way to acquire and save data.

Figure 23: VG300CB Solid-state Gyro

V. Analytical Results

V.1 ANN Model of the Vehicle

In order to determine the states of the vehicle that are essential to predicting the wheel loads, and therefore LTR, a first-order sensitivity analysis of the 2-layered function-approximating ANN was performed. Determining the 'essential' predictor states is critical for real applications of ANNs since it is often difficult to measure various states of a system directly, not to mention time-consuming and expensive. Therefore, determining the minimum amount of predictor states for input to the ANN that still provides adequate prediction properties is beneficial for a variety of reasons. Although some research has been performed into determining the importance of inputs on outputs within ANNs, to the author's knowledge, this type of sensitivity analysis has never been developed, particularly one as general in scope. Therefore, this is considered one of the more important finding of this research.

It should be pointed out, however, that a major limitation to the analysis that follows is the fact that it requires the ANN to have already been trained. This is because the sensitivity of the outputs from changes of the inputs is a function of the trained weights and biases of an ANN. Therefore, a trained network implies that the inputs have already been provided which then asks the question: if we already have the important or 'essential' inputs to predict the outputs in the first place, what is the purpose of finding them through the sensitivity analysis? The answer is that this analysis is really only useful when the training data is first provided through computer simulation -- where all the states may be available. Assuming the training data is relatively accurate, the ANN designed in software using this sensitivity analysis may then be the model for a real-world version. In other words, the sensitivity analysis of the analytical ANN would provide the starting point for designing the experimental version.

V.1.1 Sensitivity Analysis of ANNs

Although the analysis was restricted to a 2-layered ANN with tan-sigmoid activation functions in the hidden layer and linear neurons at the output layer (a commonly used function approximator [58]), the number of neurons within each layer was left to be arbitrary. However, the analysis may easily be extended to an arbitrary number of hidden layers without much effort. The procedure is unnecessary, however, since it may be shown that an ANN with multiple hidden layers is equivalent to an ANN with a single hidden layer with additional neurons. Therefore, the result of this sensitivity analysis will be relatively general in scope.

In general, the equation describing the input/output mapping for an ANN with tan-sigmoid neurons in the hidden layer and linear neurons (with scaling factors of 1) in the output layer can be shown to be,

$$Y = W^2\left(tansig\left(W^1P + B^1\right) + B^2\right)$$

where Y is the output vector [zx1], P is the input vector [nx1], W^L and B^L is the weight matrix and bias vector, respectively, for the L*th* layer. Note that for n inputs, z outputs and h neurons in the hidden layer, W^1 is [hxn] and W^2 is [zxh]. Furthermore, the sizes of B^1 and B^2 are [hx1] and [zx1], respectively.

Applying the definition of the *tansig* function described in a previous section, we find

$$tansig(W^1P + B^1) = \frac{2}{1+e^{-2\left(W^1P+B^1\right)}} - 1$$

with an output ranging from –1 to 1. The argument may be written as

$$M = W^1P^1 + B^1 = \begin{bmatrix} w^1_{1,1} & w^1_{1,2} & \cdots & w^1_{1,n} \\ w^1_{2,1} & w^1_{2,2} & \cdots & w^1_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w^1_{h,1} & w^1_{h,2} & \cdots & w^1_{h,n} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} + \begin{bmatrix} b^1_1 \\ b^1_2 \\ \vdots \\ b^1_n \end{bmatrix} = \begin{bmatrix} w^1_{1,1}p_1 + w^1_{1,2}p_2 + \cdots + w^1_{1,n}p_n + b^1_1 \\ w^1_{2,1}p_1 + w^1_{2,2}p_2 + \cdots + w^1_{2,n}p_n + b^1_2 \\ \vdots \\ w^1_{h,1}p_1 + w^1_{h,2}p_2 + \cdots + w^1_{h,n}p_n + b^1_h \end{bmatrix}_{hx1}$$

Applying the *tansig* function to M results in

$$tansig(M) = \begin{bmatrix} \frac{2}{1+e^{-2M_1}} - 1 \\ \frac{2}{1+e^{-2M_2}} - 1 \\ \vdots \\ \frac{2}{1+e^{-2M_h}} - 1 \end{bmatrix}_{hx1}$$

where M_i is the ith row of M. Pre-multiplying by the output weight matrix W^2 and adding the output layer bias vector b^2 results in

$$Y = \begin{bmatrix} w^2_{1,1} & w^2_{1,2} & \cdots & w^2_{1,h} \\ w^2_{2,1} & w^2_{2,2} & \cdots & w^2_{2,h} \\ \vdots & \vdots & \ddots & \vdots \\ w^2_{z,1} & w^2_{z,2} & \cdots & w^2_{z,h} \end{bmatrix} \begin{bmatrix} \frac{2}{1+e^{-2M_1}} - 1 \\ \frac{2}{1+e^{-2M_2}} - 1 \\ \vdots \\ \frac{2}{1+e^{-2M_h}} - 1 \end{bmatrix} + \begin{bmatrix} b^2_1 \\ b^2_2 \\ \vdots \\ b^2_h \end{bmatrix}.$$

Therefore, the sensitivity of the output from changes to the input is simply the partial derivative of the output with respect to the input (Jacobian). Since the weights and biases are independent of the input *P*, the sensitivity of *Y* due to *P* is

$$\frac{\partial Y}{\partial P}=\begin{bmatrix}\frac{\partial y_1}{\partial p_1} & \frac{\partial y_1}{\partial p_2} & \cdots & \frac{\partial y_1}{\partial p_n}\\ \frac{\partial y_2}{\partial p_1} & \frac{\partial y_1}{\partial p_2} & \cdots & \frac{\partial y_1}{\partial p_n}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial y_z}{\partial p_1} & \frac{\partial y_z}{\partial p_2} & \cdots & \frac{\partial y_z}{\partial p_n}\end{bmatrix}=\begin{bmatrix}w^2_{1,1} & w^2_{1,2} & \cdots & w^2_{1,h}\\ w^2_{2,1} & w^2_{2,2} & \cdots & w^2_{2,h}\\ \vdots & \vdots & \ddots & \vdots\\ w^2_{z,1} & w^2_{z,2} & \cdots & w^2_{z,h}\end{bmatrix}\frac{\partial}{\partial P}\begin{bmatrix}\frac{2}{1+e^{-2M_1}}-1\\ \frac{2}{1+e^{-2M_2}}-1\\ \vdots\\ \frac{2}{1+e^{-2M_h}}-1\end{bmatrix}.$$

Symbolically computing the Jacobian full of exponentials would result in an extremely large matrix. However, to elucidate, it may be shown that the partial derivative with respect to the first input p_1 is

$$\frac{\partial}{\partial p_1}\begin{bmatrix}\frac{2}{1+e^{-2M_1}}-1\\ \frac{2}{1+e^{-2M_2}}-1\\ \vdots\\ \frac{2}{1+e^{-2M_h}}-1\end{bmatrix}=4\begin{bmatrix}\frac{e^{-2M_1}}{\left(1+e^{-2M_1}\right)^2}w^1_{1,1}\\ \frac{e^{-2M_2}}{\left(1+e^{-2M_2}\right)^2}w^1_{2,1}\\ \vdots\\ \frac{e^{-2M_h}}{\left(1+e^{-2M_h}\right)^2}w^1_{h,1}\end{bmatrix}=4Q_1 \quad \text{where } Q_1=\begin{bmatrix}\frac{e^{-2M_1}}{\left(1+e^{-2M_1}\right)^2}w^1_{1,1}\\ \frac{e^{-2M_2}}{\left(1+e^{-2M_2}\right)^2}w^1_{2,1}\\ \vdots\\ \frac{e^{-2M_h}}{\left(1+e^{-2M_h}\right)^2}w^1_{h,1}\end{bmatrix}.$$

This may then be extended to the *j*th input (*j*=1…*n*)

$$\frac{\partial}{\partial p_j}\begin{bmatrix}\frac{2}{1+e^{-2M_1}}-1\\ \frac{2}{1+e^{-2M_2}}-1\\ \vdots\\ \frac{2}{1+e^{-2M_h}}-1\end{bmatrix}=4\begin{bmatrix}\frac{e^{-2M_1}}{\left(1+e^{-2M_1}\right)^2}w^1_{1,j}\\ \frac{e^{-2M_2}}{\left(1+e^{-2M_2}\right)^2}w^1_{2,j}\\ \vdots\\ \frac{e^{-2M_h}}{\left(1+e^{-2M_h}\right)^2}w^1_{h,j}\end{bmatrix}=4Q_j.$$

Therefore, the individual output sensitivities with respect to a fixed input is simply the rows of W^2 pre-multiplying the $4Q_j$ vector. For example, the sensitivity of the first output to changes in the first input is

$$\frac{\partial y_1}{\partial p_1} = 4\begin{bmatrix} w_{1,1}^2 & w_{1,2}^2 & \cdots & w_{1,h}^2 \end{bmatrix} Q_1 .$$

The generalized sensitivity equation may be expanded to

$$\frac{\partial y_i}{\partial p_j} = 4\left(\begin{array}{l} w_{i,1}^2 w_{1,j}^1 e^{-2M_1}\left(1+e^{-2M_1}\right)^{-2} + w_{i,2}^2 w_{2,j}^1 e^{-2M_2}\left(1+e^{-2M_2}\right)^{-2} + \cdots \\ \qquad\qquad + w_{i,h}^2 w_{h,j}^1 e^{-2M_h}\left(1+e^{-2M_h}\right)^{-2} \end{array} \right)$$

$$i = 1 \ldots z \qquad j = 1 \ldots n$$

Finally, this result may be put into a more convenient form for computer simulation as

$$\frac{\partial y_i}{\partial p_j} = 4\sum_{k=1}^{h} w_{i,k}^2 w_{k,j}^1 e^{-2M_k}\left(1+e^{-2M_k}\right)^{-2} \qquad \text{where } M_k = \sum_{c=1}^{n} w_{k,c}^1 p_c + b_k$$

$$i = 1 \ldots z \qquad j = 1 \ldots n$$

Note that this result is independent of time and that it computes the effect on the *i*th output y_i due to changes in the *j*th input p_j at a discrete point in time. Therefore, for each instant in time, the above formulation will compute a single Jacobian. Furthermore, this sensitivity analysis can only provide a measure of the importance of the various inputs for a network that has already been trained on that input data. Therefore, and as mentioned previously, it is really only practical when the training data comes from computer simulation. It should also be mentioned that since the input to a neural network is typically normalized, and therefore unitless, this sensitivity analysis provides sensitivities that are directly comparable as long as the output of the ANN is unitless and/or normalized as well (the LTR is inherently unitless). If the data were not normalized, a normalized version of the sensitivity analysis would be required to produce comparable quantities. For example, the partial derivative of the output with respect to the input would need to be multiplied by the ratio of inputs to outputs [50], say

$$\frac{\partial Y}{\partial P}\frac{P}{Y}.$$

The most efficient methodology for determining the essential input data that still provide adequate results is to choose variables that the user has some intuitive insight with. For example, since this project concerns rollover of a road vehicle, one might feel that the roll angle rates and accelerations might have a first-order effect on the prediction of the LTR. Therefore, the network designer would train a network based on these variables and, assuming the result was adequate, run the sensitivity analysis that would provide a measure of the 'importance' of each input. This would be useful in a situation where some of the input variables might be difficult or impractical to obtain but also not provide much help in the prediction of the output. Therefore, a decision to neglect the expensive inputs would be in order.

Example: Application of the Sensitivity Analysis

To exemplify its usefulness in determining essential inputs for prediction of outputs, the sensitivity analysis will be used to rank the importance of two channels of input data for prediction of a LTR from a straight truck during a lane-change maneuver. The first channel will be a dynamic property of the vehicle expected to have some relation to the LTR: the lateral acceleration, A_y, of the CG. The second will be a channel of random data, r, ranging from –1 to 1, which obviously has no relation to the LTR. Maneuver data was designed and the vehicle response simulated using TruckSim for a standard straight truck at 70 km/hr. Figure 24 is the layout of the lane-change maneuver.

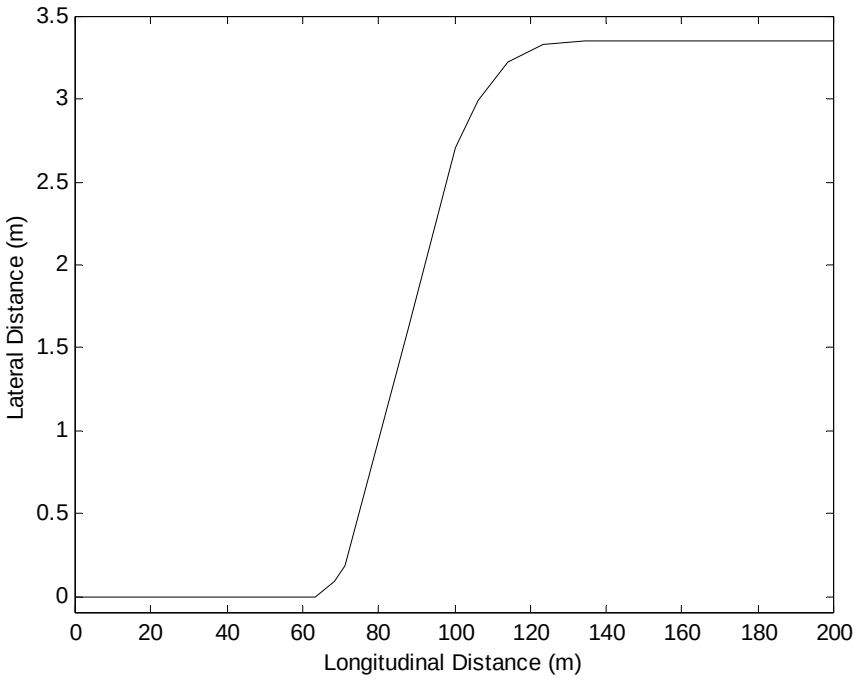


Figure 24: Lane-change maneuver to the left.

Table 2 displays the weights and biases of the trained ANN with five tan-sigmoid neurons in the hidden layer and a single linear neuron at the output. The predicted LTR is plotted along with the training data in Figure 25. Examining the weight and bias values individually provides little insight into the importance of either input. However, after the sensitivity analysis is applied to the data and results plotted in Figure 26, it is clear that the lateral acceleration has a far stronger influence on the LTR than the random data. Specifically, the sensitivity of the predicted LTR to changes in lateral acceleration of the vehicle hovers around 0.4 while that for the random data is effectively zero.

Table 2: Weights and biases for trained network

W^1		B^1	W^2	B^2
A_y	r			
0.3883	2.7209	-2.0431	-0.0082	1.6565
-1.4031	-0.5185	-1.0570	-0.3603	
-0.3915	-0.0128	1.1300	-2.1164	
-4.3986	-1.7005	-2.7858	0.0392	
0.7808	0.5862	0.8387	0.0392	

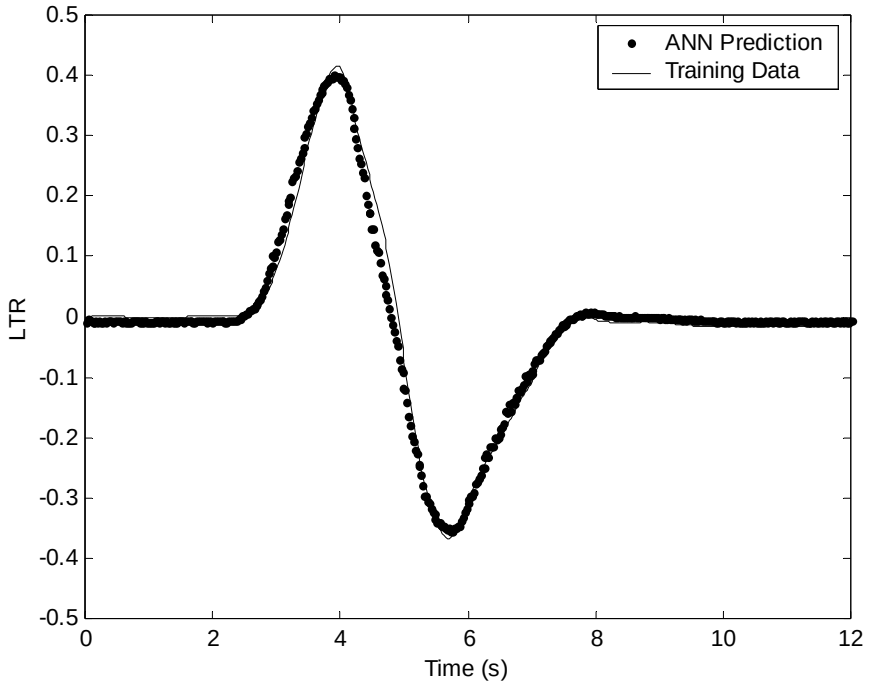


Figure 25: Result of ANN training

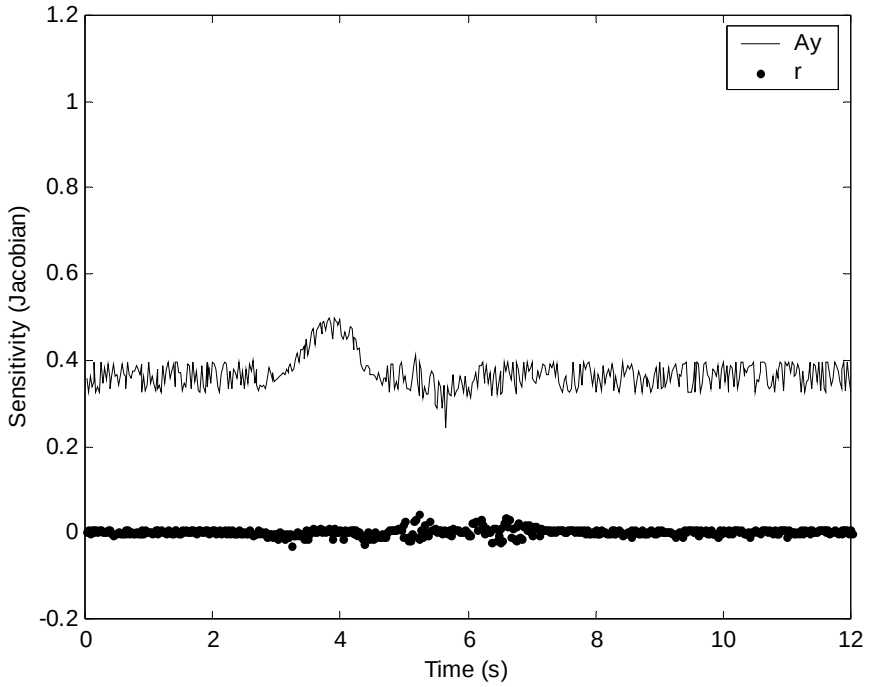


Figure 26: LTR sensitivity to vehicle lateral acceleration, *Ay*, and a channel of random data, *r*.

Time-based Derivatives

The time derivative of the LTR, or the LTR *rate*, will be shown in a later section to be an important factor when designing the warning device using Fuzzy Logic. Preliminary research involved computing the LTR rate from ANN predicted LTR. Since this quantity will be calculated on-board the vehicle in real-time from the predicted LTR, it is imperative that the predicted LTR be not only accurate, but also follow the slope of the actual LTR as closely as possible. This is evident from the predicted LTR of Figure 25 using the vehicle lateral acceleration and a channel of random data. Although the overall fit to the LTR training data was relatively good, the approximation of the LTR rate, shown in Figure 27 is very poor. This is mainly due to the fact that there are too many neurons in the hidden layer allowing the network to partially use some of the random data. Though the contribution of the random data to predicting the LTR was extremely small compared to the lateral acceleration, it was enough to cause abrupt transitions with the predicted LTR. This, in turn, causes large errors with the calculation of the LTR rate. Therefore, the practice of using as few neurons in the hidden layer as possible is typically beneficial.

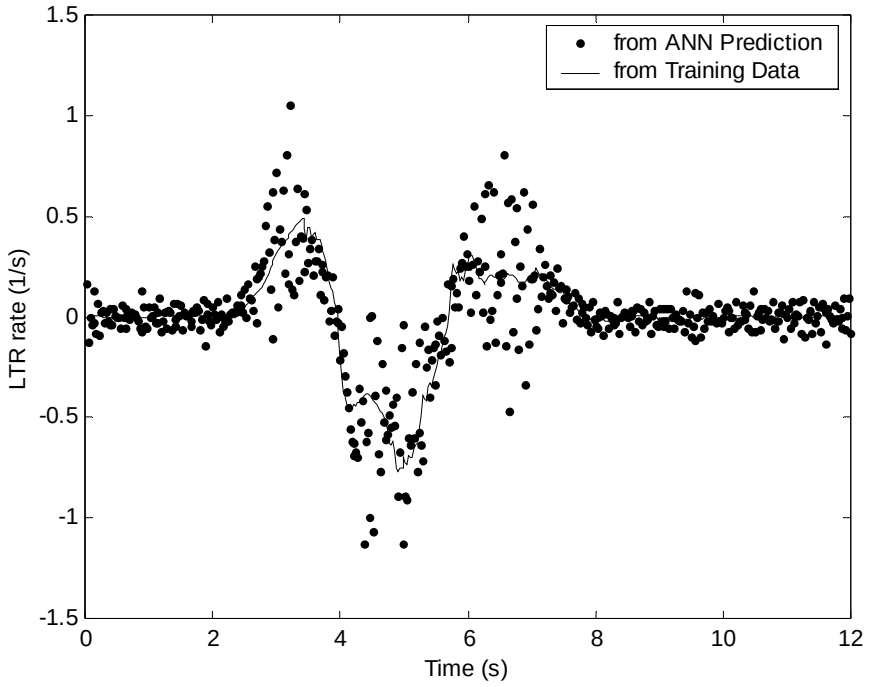


Figure 27: LTR rate computed from ANN predicted LTR

It was determined much better results for the LTR rate may be obtained by simply allowing the network to predict it along with the LTR and using the same input data. This involves adding an additional output neuron and is effectively a separate network with its own layer weights. The benefits are two-fold: first, the results are much more accurate where the network has the potential to match the training LTR rate almost perfectly, and second, the LTR rate may be computed in real-time alongside the LTR and eliminates the need for delay to calculate a time derivative. Conserving time is of course a priority when designing a warning device such as this.

V.1.2 Network Performance

In general, the performance of an ANN should always be measured in at least two ways:

1. How well does the ANN predict the training data?
2. How well does the ANN predict new data it has never 'seen' before?

The first performance question is typically answered during training by some mathematical function that measures the amount of error in the prediction, such as the mean-squared-error (MSE). Though a MSE of zero would be ideal, this is rarely the case. Therefore in practice, training is usually considered complete when the MSE reaches or surpasses some threshold value defined by the network designer. As may be expected, the performance or MSE depends on a number of factors such as the number of neurons in the hidden layer, number of epochs the network is allowed to train and fundamental relationship between the input and output data, to name a few.

The second performance measure, the networks ability to predict new data, is generally considered the most important. Again, the MSE is a convenient way to measure the robustness of a trained network. The key to producing an ANN that is robust enough to respond correctly to a wide range of new data is having good training data. In other words, if the ANN is trained upon data that spans a wide range of possibilities, it will probably fare well with data it has never seen.

When an ANN is considered 'well-balanced' or robust it is commonly referred to as a network that can *generalize* well. If a network passes the first performance test, but fails the second, it is considered a *specialized* network and can really only be used for input data that falls in a very restricted range to produce reasonable outputs. As an example of specialization, an ANN was trained using the results of a vehicle performing the sharp-left turning maneuver shown in Figure 28a at 40 km/hr. Excellent results were obtained as shown in Figure 29 where the ANN learned how to predict the training data almost perfectly. The trained network was then asked to predict the LTR for the exact opposite maneuver shown in Figure 28b with the results shown in Figure 30. It is clear that the network essentially never learned how to predict the wheel loads (and therefore LTR) when the vehicle turned to the right.

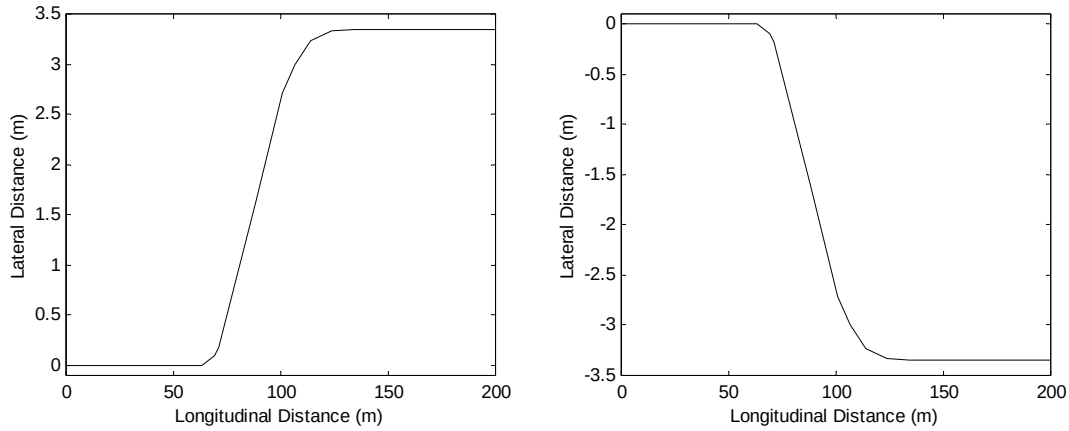


Figure 28: (a) Sharp left-turn maneuver (b) Sharp right-turn maneuver

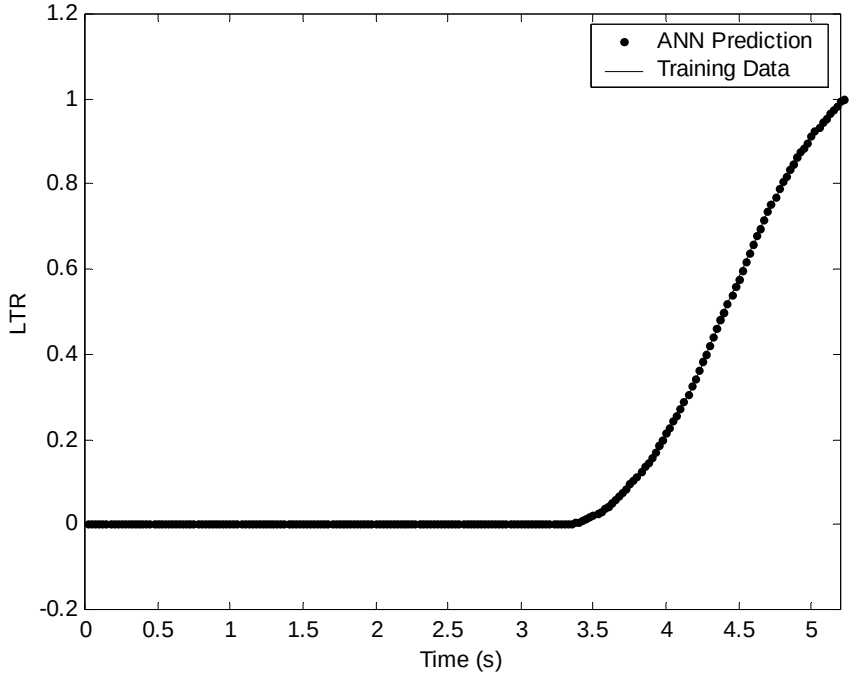


Figure 29: Predicted LTR for a left-turn maneuver that the ANN was trained with.

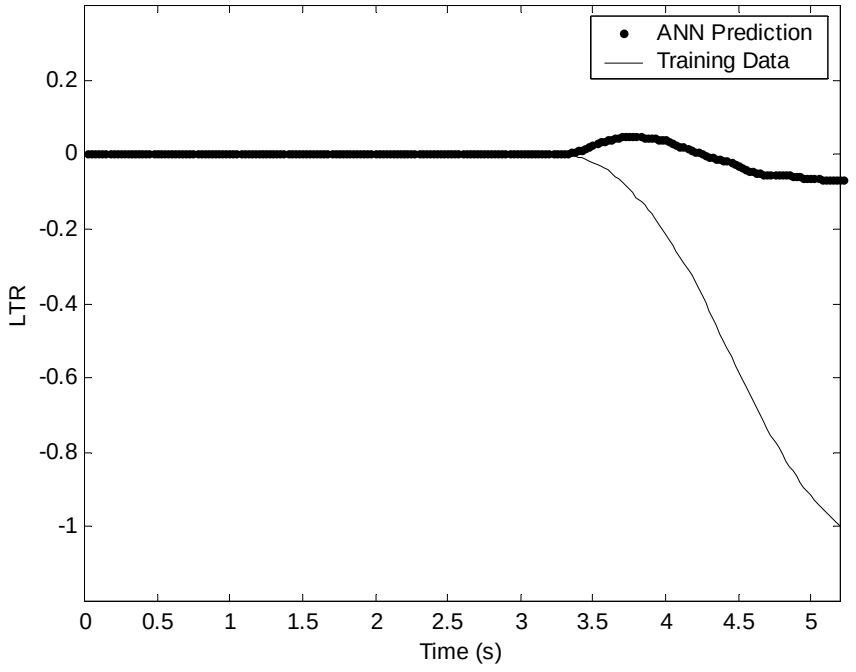


Figure 30: Predicted LTR using a specialized ANN.

Training Maneuvers

To expose the network to a wide range of inputs for training (improving generalization while preventing specialization), two maneuvers were developed that allow the vehicle to roll to both sides, and then perform a critical maneuver that forces the vehicle to rollover. Specifically, this was accomplished by maneuvering the vehicle through a lane-change to one side, followed by a lane-change to the other side, allowing the vehicle to settle, then forcing the vehicle into a sharp turn that it cannot recover from. The second maneuver is the complete opposite of the first and both were used in training the same network. This allows the network to have equal exposure to both roll possibilities during training. The symmetric maneuvers are shown in Figure 31. Note that although the training data (LTR calculated through TruckSim) for each maneuver was calculated separately, the data was concatenized for input to the network. Recall that the order in which the input/output pairs are presented to the network is unimportant since the network learns by minimizing the MSE over *all* the data for each epoch. As a matter of fact, the data could be scrambled in a random order and then presented to the network for training (but keeping each input with its corresponding output) and the network would converge to the same result.

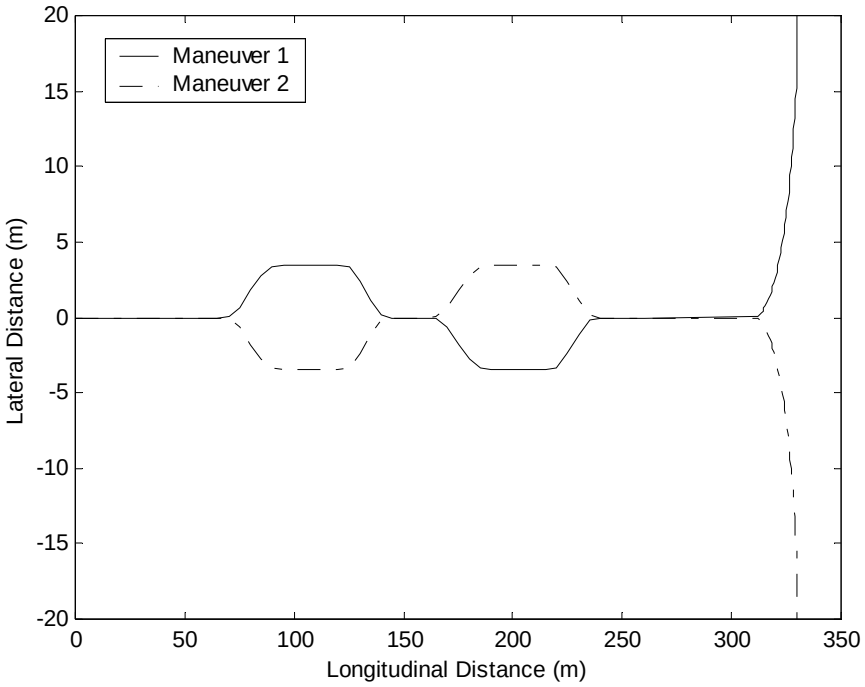


Figure 31: Double lane-change maneuver with sharp turn

V.1.4 Results of the ANN Model

The sensitivities for a wide range of input variables were computed and analyzed, and therefore to display all the results would be impractical. However, Figure 32 displays the typical sensitivities of the LTR using a 2-1 ANN for prediction with four inputs: lateral

acceleration, yaw rate, the roll rate and a channel of random data used as a control group. This plot clearly shows the effectiveness of using the sensitivity relations on the input data. For this trained network, it may be concluded that the lateral acceleration, A_y, has the largest effect on predicting the LTR, with sensitivity range of 0.2 over a static value of 1.85. Tied for second is the yaw rate (YR) with a range of approximately zero and a static value of close to -1. The sensitivity of LTR to changes in roll rate varies slightly more than that of yaw rate changes, but over a static value of zero. Again, the random data clearly has no effect on prediction of the LTR.

Figure 33 displays the sensitivity of the LTR rate prediction using the same four inputs. This time, the sensitivity of the LTR rate varies greatly with changes to the individual inputs. Specifically, the LTR rate was most sensitive to changes in roll rate and lateral acceleration, with results having a range of 0.5 and 1.5 over static values of approximately 0.9 and 0.25, respectively. A close second was the sensitivity to changes in the yaw rate with a range of 0.25 over a static value of almost 0.5. The random data has virtually no usefulness in predicting the LTR rate.

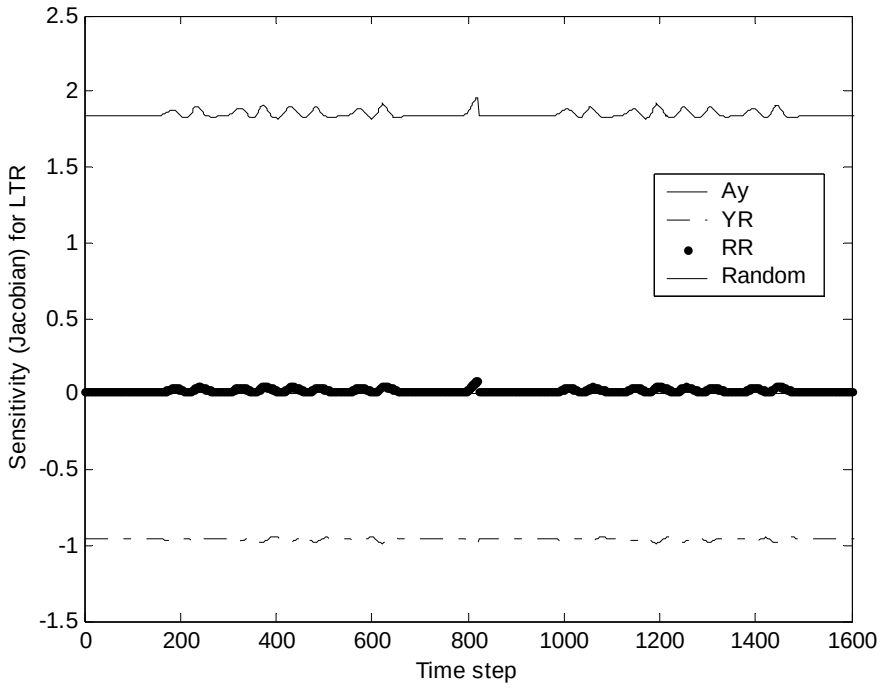


Figure 32: LTR sensitivity to various inputs

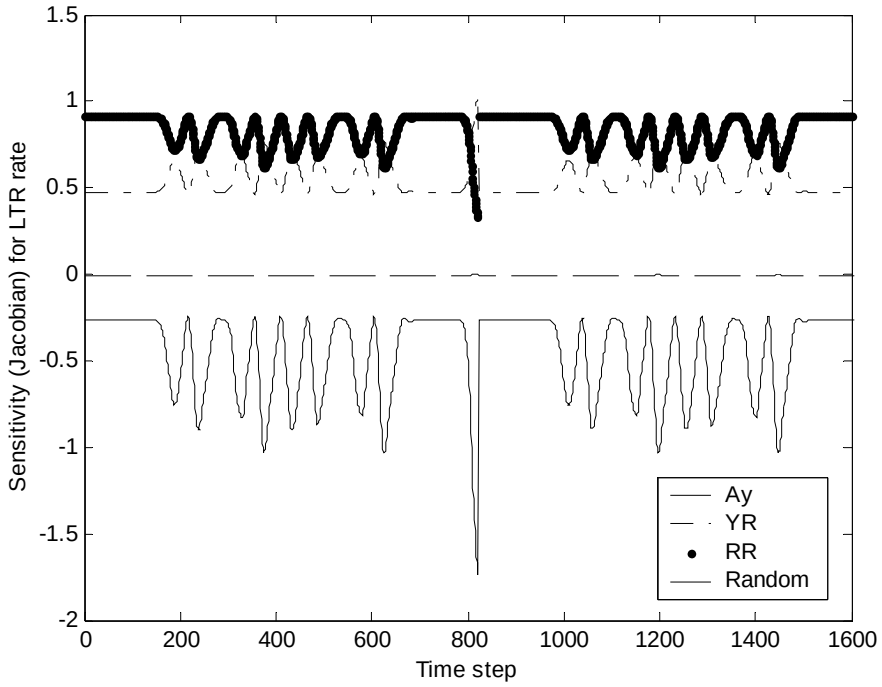


Figure 33: LTR rate sensitivity to various inputs

After numerous tests of a multitude of input combinations, it was concluded that the lateral acceleration has the most significant contribution to an accurate prediction of the LTR. This was expected since the lateral acceleration is the direct cause of load transfer and vehicle roll. Four other variables that had a significant contribution to the prediction of the LTR were, in order of importance:

- Steering factor (SF)
- Roll acceleration (RA)
- Yaw rate (YR)
- Roll rate (RR)

Though none of the respective 'important' inputs were a surprise, it was interesting that the network considered the SF more important than the roll information (rate or acceleration). The author expected the roll information to be almost as important as the lateral acceleration, though it clearly was not.

Using the results of the sensitivity analysis, five different architectures of ANNs (2-2, 3-2, 5-2, 10-2 and 20-2) using all possible combinations of A_y and the four other significant input variables were used to predict the LTR and LTR rate for the maneuvers shown in Figure 31. The results are shown in Table 3 and Table 4 for the LTR and computed LTR rate, respectively.

A general trend that is evident from the data is that using more inputs to the network and/or more neurons in the hidden layer improves the predictive power of both the LTR and the LTR rate. Therefore, the simplest network using just the lateral acceleration and two neurons in the hidden layer predicted the LTR with relatively poor results shown in Figure 34 with MSE of 0.0070. The LTR rate is plotted in Figure 35 with a MSE of 0.1036. On the other hand, the most complex network tested using all five inputs and 20 neurons in the hidden layer produced the best results for the LTR and is shown in Figure 36 with a MSE of 0.000032, a decrease of almost 220x. Likewise, the LTR rate with a MSE of 0.00087 (decrease of almost 120x) is shown in Figure 37. Comparing the four plots visually emphasizes the superior ability of the more complex network to predict the LTR and LTR rate.

Table 3: LTR training results using combinations of lateral acceleration with other inputs and various network architectures.

Inputs					LTR (MSE x 10^{-4})				
Ay	**SF**	**YR**	**RR**	**RA**	**2-2**	**3-2**	**5-2**	**10-2**	**20-2**
♦					70.00	67.45	68.00	64.00	63.00
♦	♦				16.67	7.21	8.35	6.10	2.88
♦		♦			27.04	29.46	12.73	16.60	12.78
♦			♦		44.85	27.68	33.58	19.96	13.48
♦				♦	17.93	17.25	15.54	13.09	9.33
♦	♦	♦			6.79	7.10	7.62	5.35	2.16
♦	♦		♦		3.72	3.77	2.72	1.58	1.57
♦	♦			♦	1.79	1.85	1.39	1.05	1.37
♦		♦	♦		24.62	24.39	19.88	7.04	4.75
♦		♦		♦	10.95	10.83	6.17	5.93	3.90
♦			♦	♦	3.46	3.44	2.81	1.11	1.06
♦	♦	♦	♦		3.71	3.81	2.87	2.43	0.81
♦	♦	♦		♦	13.59	11.16	0.92	0.93	0.64
♦		♦	♦	♦	3.45	3.45	2.75	1.30	0.87
♦	♦	♦	♦	♦	0.94	1.07	0.93	0.37	0.32

Table 4: LTR rate training results using combinations of lateral acceleration with other inputs and various network architectures.

Inputs					LTR Rate (MSE x 10^{-3})				
Ay	**SF**	**YR**	**RR**	**RA**	**2-2**	**3-2**	**5-2**	**10-2**	**20-2**
♦					103.60	101.92	101.20	98.80	98.20
♦	♦				33.55	34.48	25.10	13.67	10.39
♦		♦			26.15	25.49	18.74	13.68	13.89
♦			♦		3.13	3.29	2.65	2.12	1.83
♦				♦	88.30	87.33	81.46	45.85	40.74
♦	♦	♦			30.28	24.53	15.06	10.19	5.72
♦	♦		♦		2.24	2.18	2.16	1.15	1.06
♦	♦			♦	14.71	13.30	8.14	5.14	2.54
♦		♦	♦		2.55	2.34	2.13	1.63	1.17
♦		♦		♦	16.55	16.25	11.21	9.30	6.40
♦			♦	♦	2.62	2.51	2.30	1.61	1.21
♦	♦	♦	♦		2.19	2.14	1.90	1.38	0.93
♦	♦	♦		♦	5.31	6.94	3.51	1.70	1.32
♦		♦	♦	♦	2.40	2.33	2.10	1.15	1.01
♦	♦	♦	♦	♦	1.85	1.78	1.54	0.74	0.87

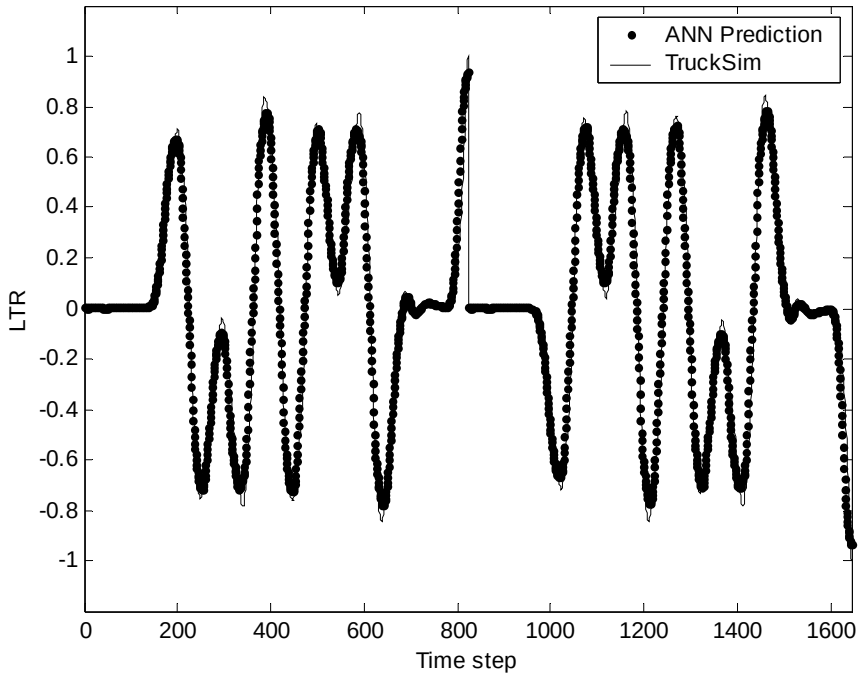


Figure 34: LTR prediction results using a 2-2 ANN with one input, A_y (MSE=0.0070).

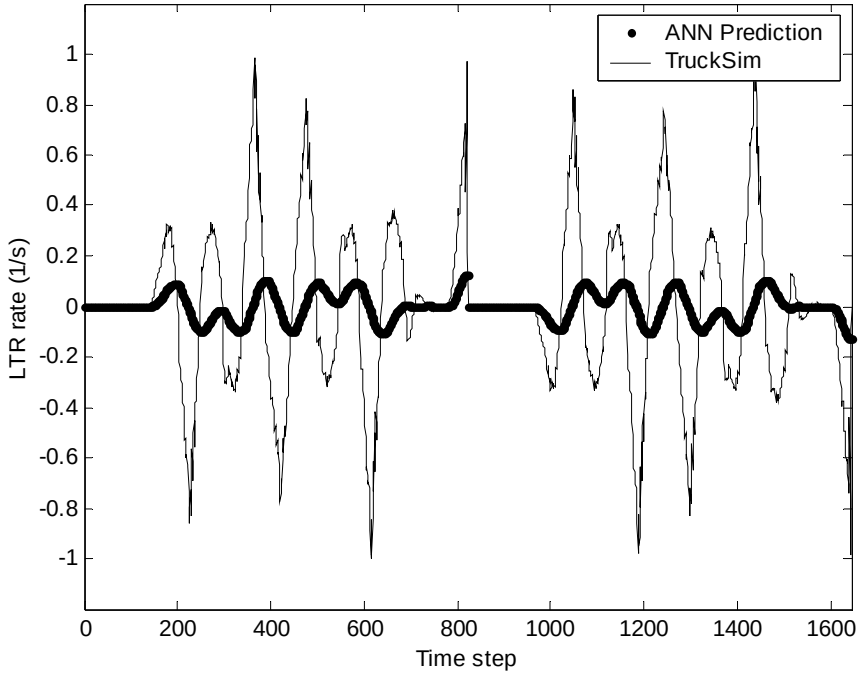


Figure 35: LTR rate prediction results using a 2-2 ANN with one input, A_y (MSE=0.1036).

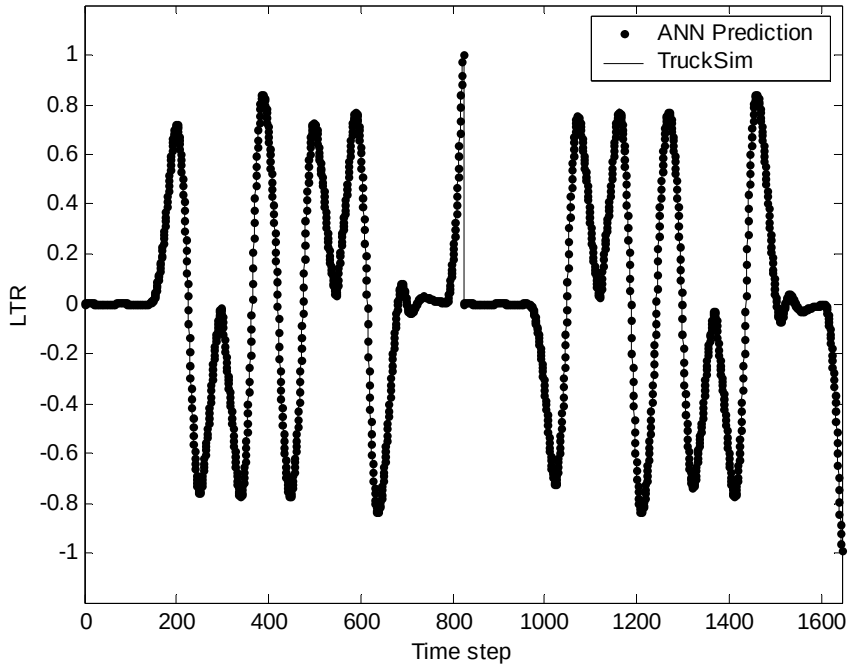


Figure 36: LTR prediction results using a 20-2 ANN with five inputs, A_y, SF, YR, RR & RA (MSE=0.000032).

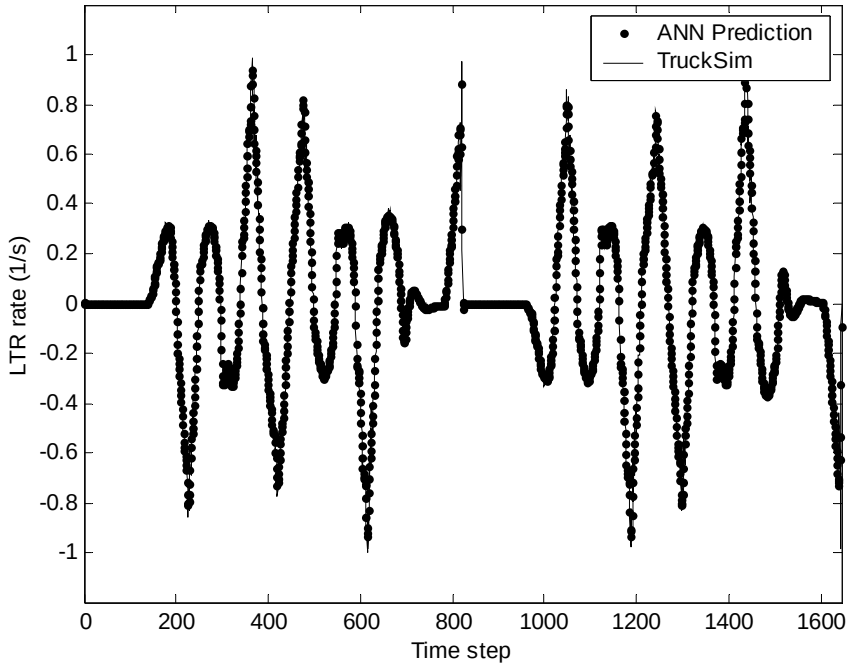


Figure 37: LTR rate prediction results using a 20-2 ANN with five inputs, A_y, SF, YR, RR & RA (MSE=0.00087).

Although the trend of the results indicates that more inputs and/or neurons decrease the MSE in general, the results of the same 2-2 network with the addition of the roll rate and roll acceleration as an input yielded a LTR-MSE of 0.000346 and a LTR rate-MSE of 0.00262; a decrease of more than 20x and 40x, respectively. A plot of the predicted LTR using this network is shown in Figure 38 and the predicted LTR rate in Figure 39 which shows a clear improvement over the LTR results shown in Figure 34, and a drastic improvement over the LTR rate results shown in Figure 35. Furthermore, the predictions of the LTR and LTR rate are not too far from that shown in Figure 36 and Figure 37, respectively, though a far simpler network.

Finally, it was noticed that the addition of the roll acceleration tended to reduce some phase lag in the predicted LTR. Figure 40 is an example of the phase lag that was commonly present when the roll acceleration was not included as an input to the network. In contrast, Figure 41 shows the training results using the same network but with the addition of the roll acceleration. Though the amount of phase lag is relatively small (on the order of 5 time steps = 0.125 seconds), previous research has shown that the amount of time available to warn the driver before rollover occurs is somewhere between 0.25 and 3 seconds [38]. Therefore, a 0.125 second lag in predicting the LTR may have a significant impact of the performance of the RWD.

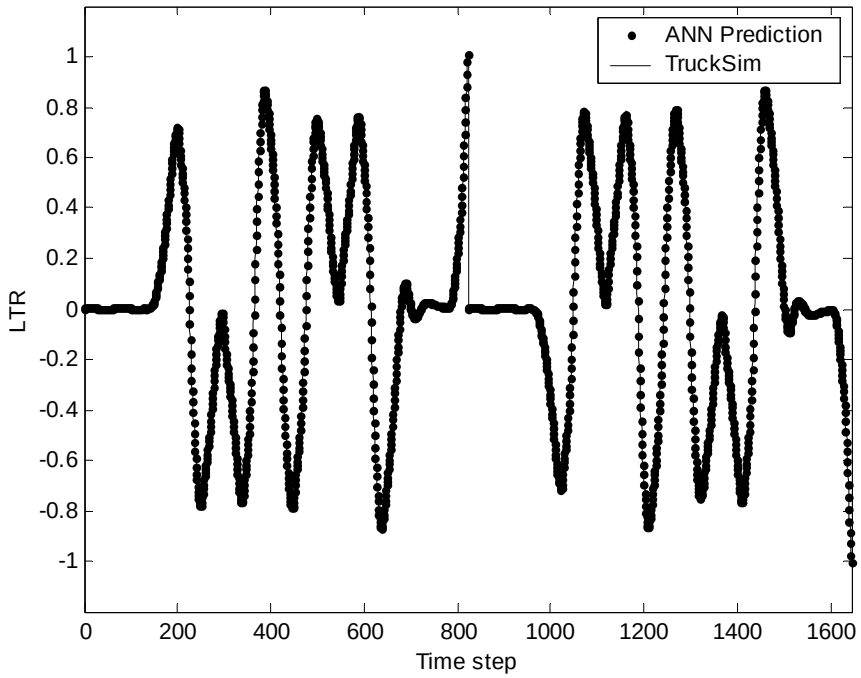


Figure 38: LTR prediction results using a 2-2 ANN with three inputs, Ay, RR & RA (MSE=0.000346).

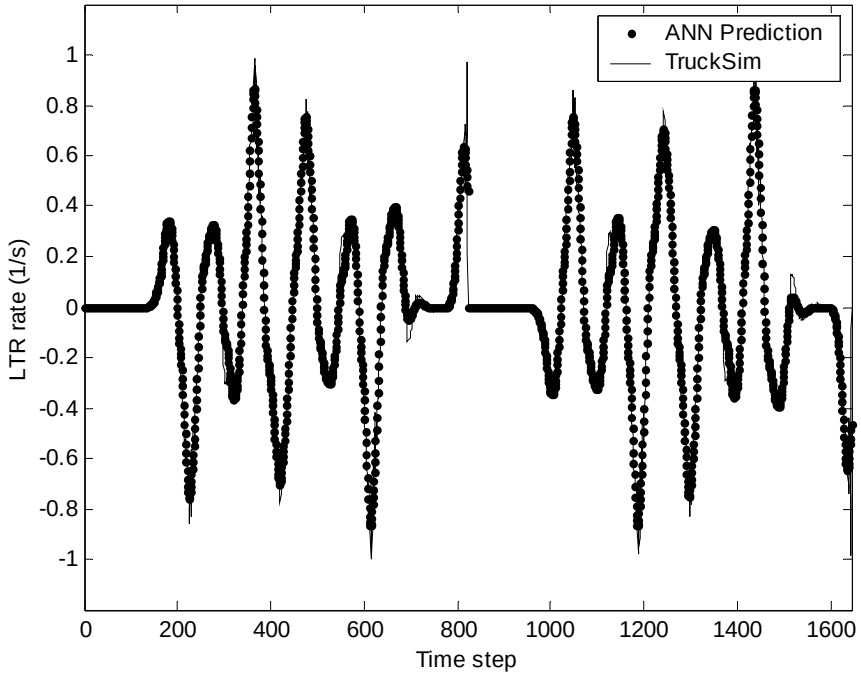


Figure 39: LTR rate prediction results using a 2-2 ANN with three inputs, Ay, RR & RA (MSE=0.00262).

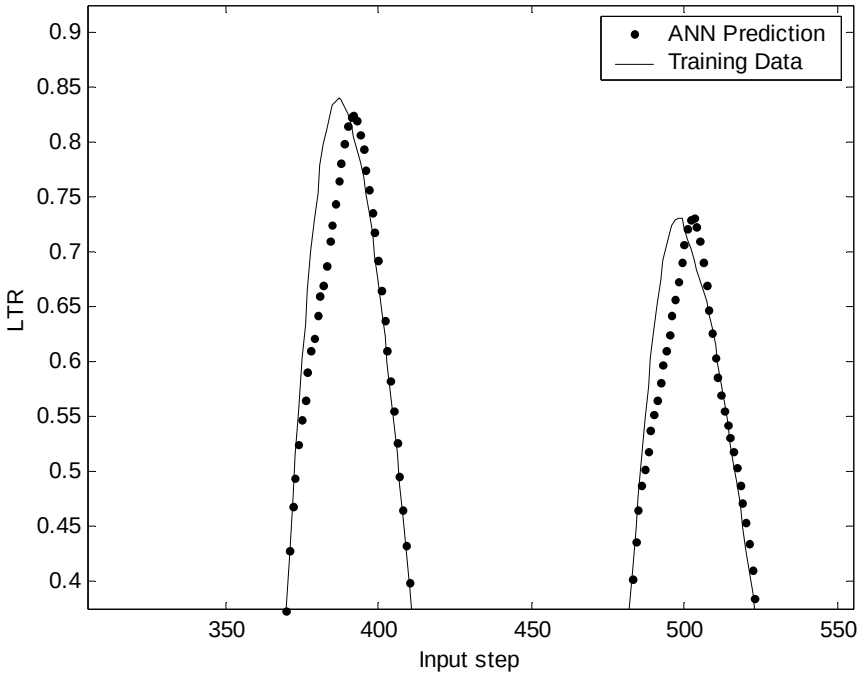


Figure 40: Phase-lag representative of most LTR predictions

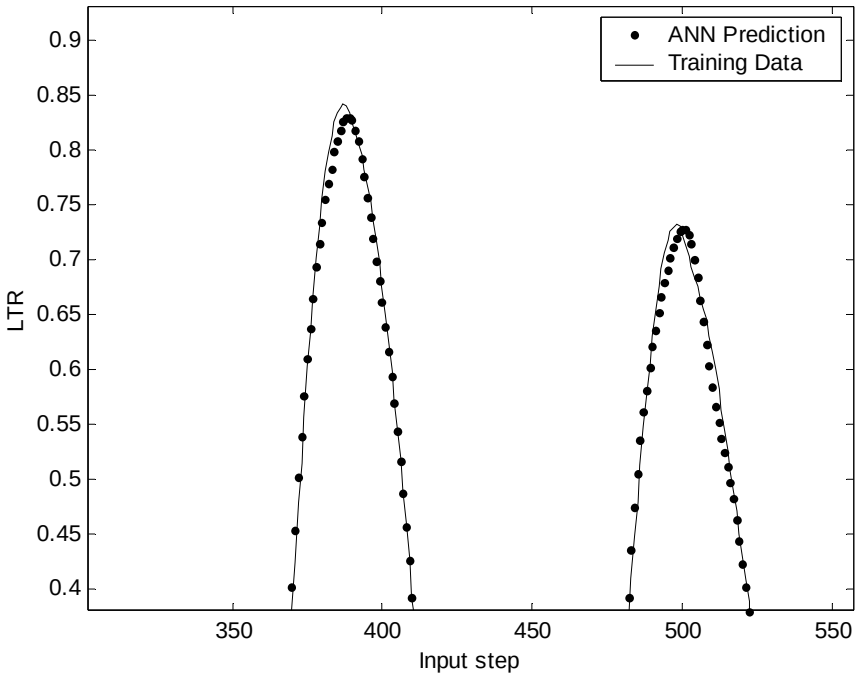


Figure 41: Corrected phase-lag using the roll acceleration (RA) as an input.

Some general conclusions that can be made from the results are:

- In general, results improved (decrease in MSE) as the complexity of the network increased (more inputs and/or neurons in the hidden layer). However, good results were obtained with using only a few inputs and with relatively simple networks.
- The exactness with which a specific network predicted the LTR given a set of inputs did not guarantee its ability to likewise predict the LTR rate to an equal degree of exactness (and vice versa).
- Using the combination of lateral acceleration (A_y), steering factor (SF) and roll acceleration (RA) or lateral acceleration, roll rate (RR) and roll acceleration as inputs produced consistently good results, regardless of the number of neurons used in the hidden layer.
- The addition of the roll acceleration (RA) as an input appears to correct some phase-lag in the predicted LTR results. This is probably due to the fact that the roll acceleration is in-phase with the load transfer, i.e., they both become non-zero at the same instant.

Seven of the 15 input architectures may be considered to have performed best, based on LTR-MSE from Table 3 the LTR rate MSE from Table 4, regardless of the number of neurons in their hidden layer. Ranked in order of performance, these are:

1. A_y-SF-YR-RR-RA
2. A_y-SF-YR-RR
3. A_y-YR-RR-RA
4. A_y-SF-RR
5. A_y-RR-RA
6. A_y-SF-RA
7. A_y-YR-RR

resulting in a total of 35 trained networks (7 input x 5 hidden layer architectures). Although the LTR-MSE for the A_y-YR-RR networks may seem fairly high relative to the other six, it was included because it performed well in a global sense, the LTR rate-MSE was surprisingly low and it consists of using inputs that are easy to measure relative to variables such as the steering factor or roll acceleration.

V.1.4 Robustness/Generalization Tests

As mentioned previously, the ability of a trained network to correctly predict outputs for inputs it has never 'seen' before (though similar to the training inputs) is typically considered the most important aspect for an ANN. This is due to the fact that most ANNs are developed and trained to work as a black-box to compute a wide range of outputs for a wide range of inputs. Furthermore, not every input/output pair may be available for training the ANN where typically only a small set of training data representative of possible inputs may be available. Therefore, the various ANNs trained in the preceding sections were tested for their ability to predict the LTR and subsequent LTR rate for a variety of maneuvers.

Test Maneuvers

Three different path-follow maneuvers were designed to try and represent the most common types of maneuvers that may lead to an unstable situation. Figure 42 is a simple lane-change (LC) maneuver that provides frequent instability with vehicles traveling at high speeds. This is particularly dangerous among heavy trucks with high CGs performing obstacle avoidance where instability occurs rather quickly. Figure 43 is a constant-radius turning (CRT) maneuver that is typical of on and off-ramp trajectories from highways. This maneuver imparts a constant lateral acceleration (assuming constant speed), which may slowly roll the vehicle past its roll-threshold. Finally, Figure 44 is a sharp turning (ST) maneuver used to represent a dangerous situation when a vehicle attempts to a turn a corner too quickly. To make the situation as realistic as possible, braking was used to try and represent how a driver may react to the situation. It should be mentioned that these maneuvers were developed solely for testing the predictive power of the trained ANN and are not necessarily maneuvers where a RWD

would be useful. It was mentioned previously that a RWD would only prove useful if the driver had enough time to take corrective action. Of the three, only the CRT would likely fit this requirement.

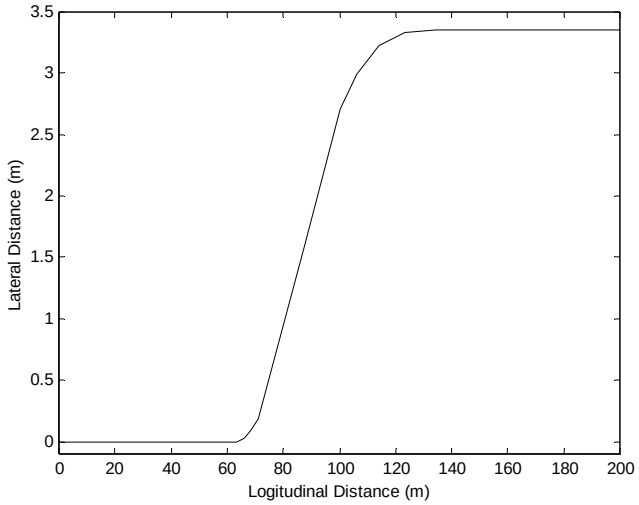


Figure 42: Lane-change maneuver

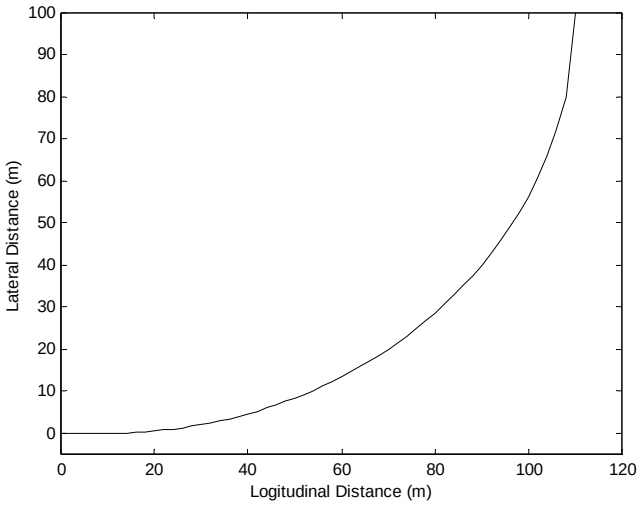


Figure 43: Constant-radius turning maneuver

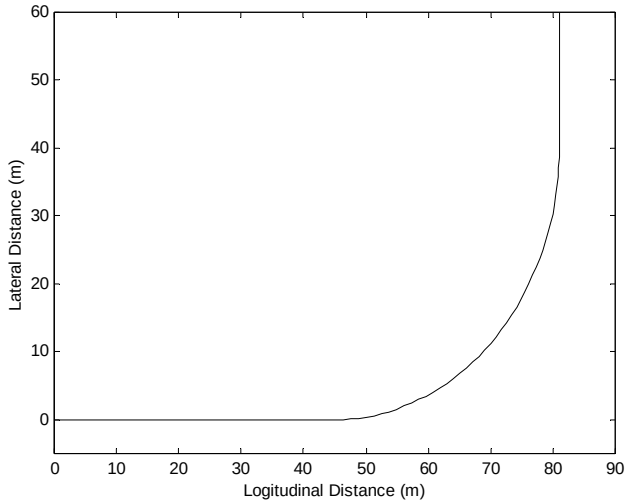


Figure 44: Sharp turning maneuver

The vehicle was put through each of the maneuvers at four different speeds: 40, 50, 60 and 70 km/hr for the CRT and ST maneuver, and 60, 70, 80 and 90 km/hr for the LC maneuver. The speeds chose satisfied vehicle response criteria such as a desired LTR falling between 0.5 and 1.0, and/or forcing the vehicle to roll over during the maneuver.

Each of the 35 trained networks from the previous section was applied to each of the three test maneuvers at all the different speeds. This resulted in a total of 420 robustness tests (35 networks x 4 speeds x 3 maneuvers). Again the MSE performance function was used to judge how well the trained networks were able to predict the LTR for the various test maneuvers. To simplify results, the average MSE over the vehicle speeds was calculated for each maneuver providing an overall measure of how well the trained networks predicted the LTR and LTR rate for a specific maneuver. The results for the LTR and LTR rate are tabulated in Table 5 and Table 6, respectively. Note that a filter was applied to the MSE function that ignored the initial steady-state data when the vehicle was moving in a straight line. It was found that including this data falsely reduced the MSE for tests where the vehicle moved slowly compared to those tests run at higher speeds. This is because the network is very good at representing the steady-state 'zero' data and since the MSE is a summation of the squared individual errors, the tests run at slower speeds produced more data points in the steady-state region (since the sampling time stayed constant) and consequently skewed the MSE. Furthermore, the filter ignored data when the LTR reached unity. This is due to the fact that the networks consistently were unable to predict the LTR for some maneuvers where the actual LTR reached unity and stayed there without completely rolling over. These poor results adversely skewed the MSE for the prediction as a whole. Therefore, the individual results tabulated in Table 5 and Table 6 may be compared on a fair basis.

Table 5: Average MSE for the predicted LTR

Predicted LTR (MSE)							
2-N	**SF-RR**	**SF-RA**	**YR-RR**	**RR-RA**	**SF-YR-RR**	**YR-RR-RA**	**SF-YA-RR-RA**
CRT	0.008677	0.003652	0.003503	0.000295	0.008480	0.000406	0.004159
LC	0.007999	0.003268	0.002377	0.000262	0.007825	0.000283	0.003323
ST	0.008220	0.003348	0.037818	0.002488	0.007801	0.002508	0.007122
3-N	**SF-RR**	**SF-RA**	**YR-RR**	**RR-RA**	**SF-YR-RR**	**YR-RR-RA**	**SF-YA-RR-RA**
CRT	0.009465	0.004079	0.008749	0.000287	0.009043	0.000365	0.004461
LC	0.008540	0.003681	0.003182	0.000267	0.008334	0.000293	0.003673
ST	0.008664	0.003307	0.101980	0.002503	0.007980	0.002458	0.005499
5-N	**SF-RR**	**SF-RA**	**YR-RR**	**RR-RA**	**SF-YR-RR**	**YR-RR-RA**	**SF-YA-RR-RA**
CRT	0.007308	0.003384	0.007106	0.000334	0.008705	0.000711	0.003661
LC	0.006050	0.003167	0.003498	0.000278	0.007764	0.000530	0.003104
ST	0.013306	0.003310	0.257090	0.009582	0.035067	0.009603	0.008060
10-N	**SF-RR**	**SF-RA**	**YR-RR**	**RR-RA**	**SF-YR-RR**	**YR-RR-RA**	**SF-YA-RR-RA**
CRT	0.012383	0.003384	0.011057	0.000522	0.011296	0.000987	0.002760
LC	0.010784	0.003288	0.003916	0.000322	0.009849	0.000424	0.002415
ST	0.009316	0.003636	0.373719	0.003027	0.022791	0.002311	0.008154
20-N	**SF-RR**	**SF-RA**	**YR-RR**	**RR-RA**	**SF-YR-RR**	**YR-RR-RA**	**SF-YA-RR-RA**
CRT	0.018515	0.007402	0.014176	0.000950	0.005348	0.001025	0.002092
LC	0.018515	0.007402	0.014176	0.000950	0.005348	0.001025	0.002092
ST	0.009794	0.007055	0.193859	0.003104	0.025910	0.003723	0.007792

Table 6: Average MSE for calculated LTR rate

Predicted LTR Rate (MSE)							
2-N	**SF-RR**	**SF-RA**	**YR-RR**	**RR-RA**	**SF-YR-RR**	**YR-RR-RA**	**SF-YA-RR-RA**
CRT	0.060798	0.062186	0.096478	0.082553	0.065828	0.089802	0.056579
LC	0.015883	0.045923	0.016458	0.016674	0.015658	0.016216	0.014773
ST	1.320577	1.701947	1.333903	1.297166	1.326647	1.332280	1.349160
3-N	**SF-RR**	**SF-RA**	**YR-RR**	**RR-RA**	**SF-YR-RR**	**YR-RR-RA**	**SF-YA-RR-RA**
CRT	0.064195	0.068598	0.140524	0.087228	0.062999	0.094634	0.055934
LC	0.015998	0.047934	0.016581	0.016939	0.015832	0.016419	0.014404
ST	1.002697	1.323767	3.240179	1.000660	1.017740	1.028116	1.053606
5-N	**SF-RR**	**SF-RA**	**YR-RR**	**RR-RA**	**SF-YR-RR**	**YR-RR-RA**	**SF-YA-RR-RA**
CRT	0.187069	0.080453	0.092229	0.087612	0.066888	0.081668	0.104233
LC	0.015668	0.069463	0.016667	0.017681	0.014364	0.015010	0.014604
ST	0.952591	1.315700	1.936839	1.167728	1.326557	1.294812	1.407982
10-N	**SF-RR**	**SF-RA**	**YR-RR**	**RR-RA**	**SF-YR-RR**	**YR-RR-RA**	**SF-YA-RR-RA**
CRT	0.074598	0.068161	0.089209	0.085319	0.082590	0.103340	0.059272
LC	0.012021	0.068563	0.014049	0.014304	0.013665	0.014269	0.013992
ST	1.041174	1.344968	1.043817	1.039646	0.966765	1.130280	1.730928
20-N	**SF-RR**	**SF-RA**	**YR-RR**	**RR-RA**	**SF-YR-RR**	**YR-RR-RA**	**SF-YA-RR-RA**
CRT	0.154145	0.862524	0.079286	0.087218	0.216092	0.081231	0.108005
LC	0.016493	0.051369	0.013344	0.014603	0.010719	0.012367	0.022079
ST	0.992261	1.508159	2.074607	1.163769	1.856947	1.175919	1.043460

Conclusion

A conclusion that may be made from the robustness test results is that in contrast to the training results in Table 3, an increase in hidden layer neurons and/or more inputs does not necessarily produce more accurate predictions of LTR for new data (likewise with the LTR rate prediction). However, consistently good results were obtained using the A_y-RR-RA & A_y-YR-RR-RA combination of inputs using a limited number of hidden layer neurons. To exemplify the robustness of, for example, the A_y-RR-RA inputs using a 2-2 ANN architecture, plots of the LTR and LTR rate for each maneuver performed at 60 km/hr are shown in Figure 45 through Figure 50. It is visually evident that the trained networks perform very well at predicting both the LTR and LTR rate for both the CRT and LC maneuvers. The prediction of the LTR and LTR rate for the ST maneuver may be considered fair when ignoring the data in the region where the test LTR is unity. This is acceptable since the RWD would probably not be very useful when the LTR is one.

From these results, it seems promising that an ANN using a relatively few number of inputs can adequately predict the LTR and LTR rate for an experimental vehicle while performing the type of maneuvers that a RWD would prove useful, such as the CRT and LC maneuvers. Of course this depends on a number of factors such as availability of good training data and the accuracy with which TruckSim models a vehicle performing these types of maneuvers.

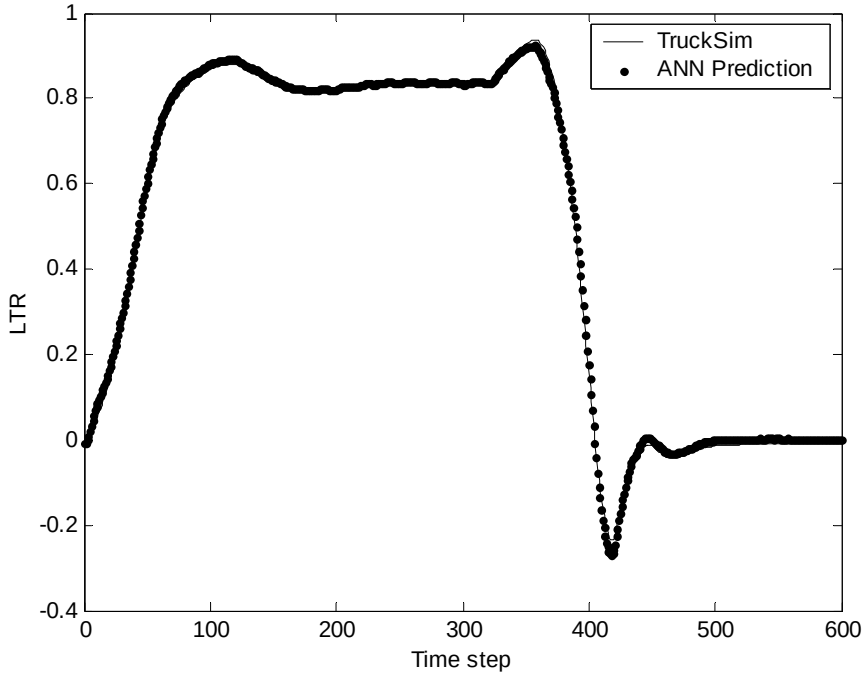


Figure 45: LTR prediction for maneuver CRT performed at 60 km/hr using a 2-2 ANN with inputs A_y-RR-RA

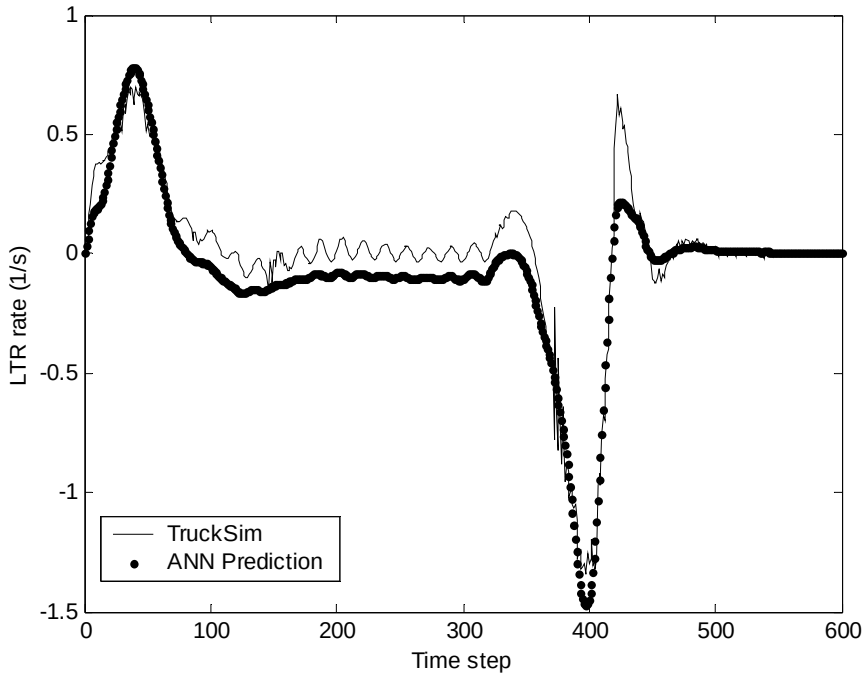


Figure 46: LTR rate prediction for maneuver CRT performed at 60 km/hr using a 2-2 ANN with inputs A_y-RR-RA

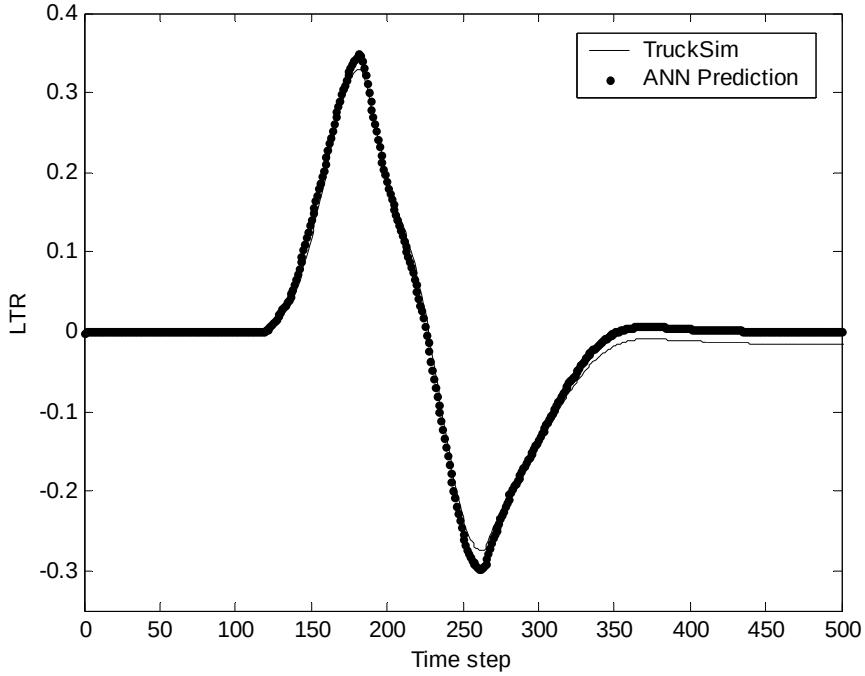


Figure 47: LTR prediction for maneuver LC performed at 60 km/hr using a 2-2 ANN with inputs A_y-RR-RA

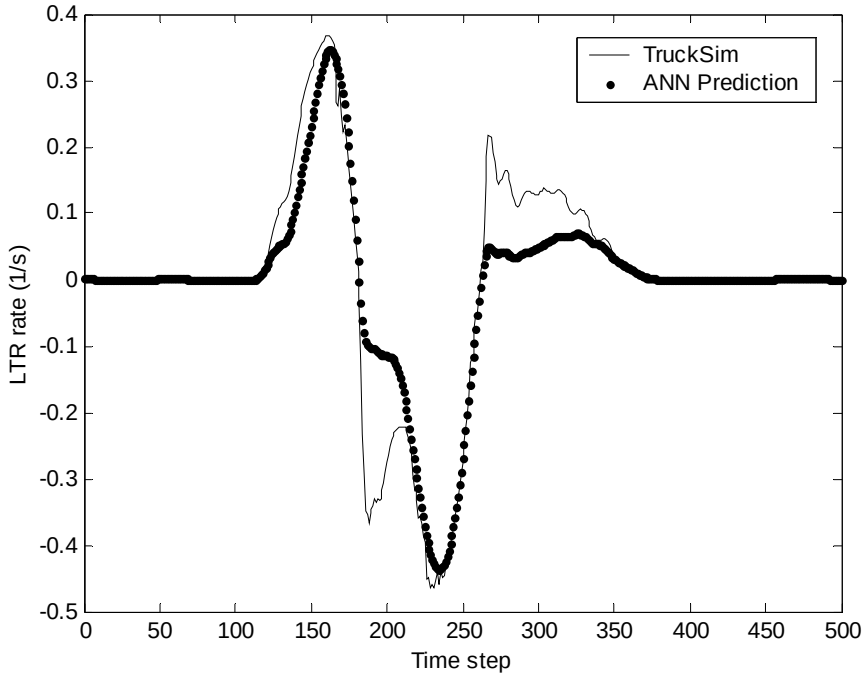


Figure 48: LTR rate prediction for maneuver LC performed at 60 km/hr using a 2-2 ANN with inputs A_y-RR-RA

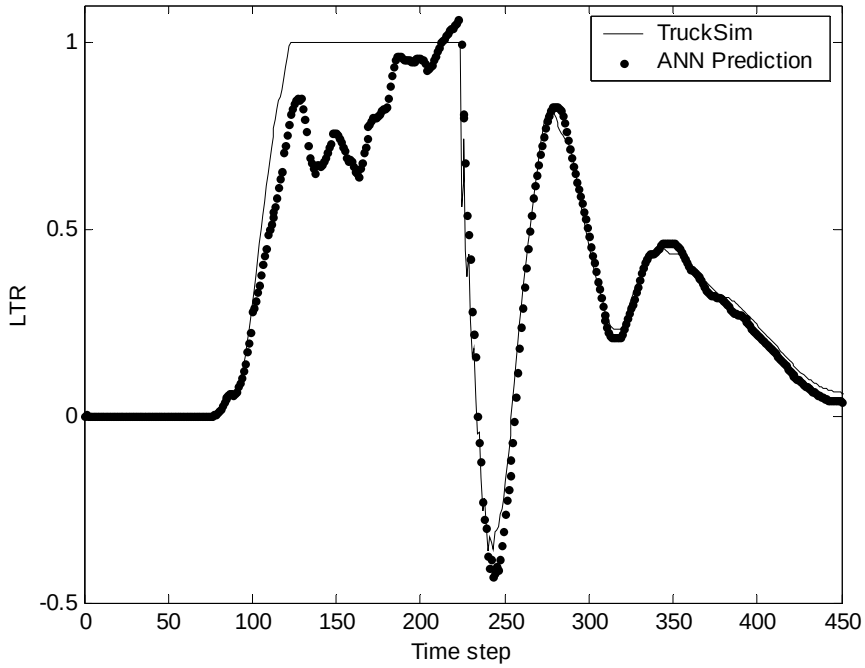


Figure 49: LTR prediction for maneuver ST performed at 60 km/hr using a 2-2 ANN with inputs A_y-RR-RA

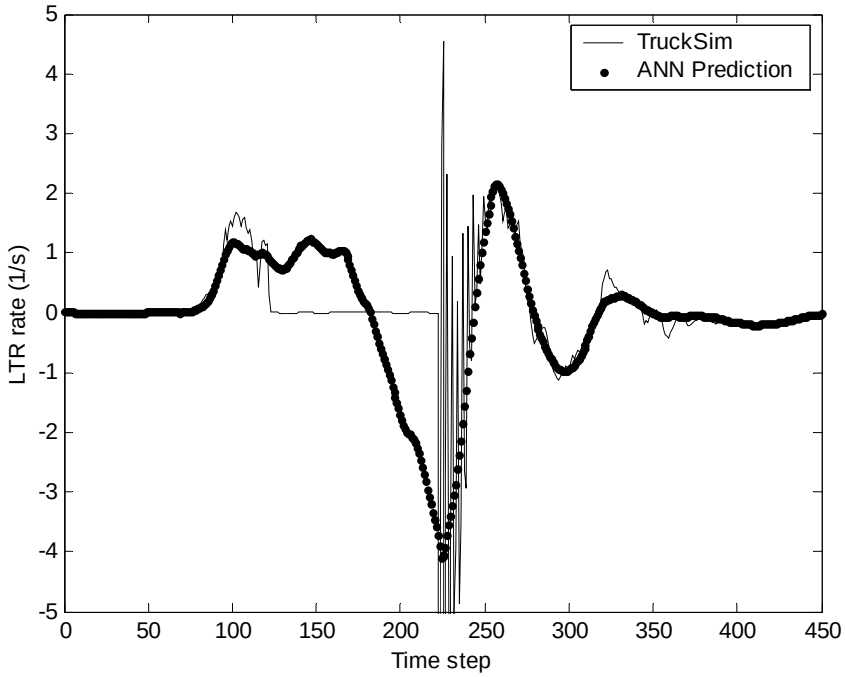


Figure 50: LTR rate prediction for maneuver ST performed at 60 km/hr using a 2-2 ANN with inputs A_y-RR-RA

V.2 Fuzzy Logic Inference System for RWD

The heart of the RWD is a Fuzzy Logic Inference System (FLIS) that is developed using a number of linguistic rules that 'make sense' based on the type of inputs and desired outputs. If the task of determining roll stability was left to the human driver, two questions that might be asked are 'what is the present condition of roll stability?' and 'where is the condition of roll stability headed?'. The inputs that may help in answering the two questions are the LTR and LTR rate, respectively. Therefore a FLIS was developed that uses these two inputs to output a temporal prediction on the roll stability of the vehicle. Furthermore, the FLIS was developed using both the Mamdani and Sugeno Methods for fuzzy inference for comparison.

A convenient yet arbitrary output of the FLIS is a warning that lies between –1 and +1, like that on the dial gauge diagram shown in Figure 51. The sign indicates the direction of the instability (left or right) and the magnitude indicates the severity of a potential instability, where a magnitude of 1 refers to the most dangerous situation and 0 indicates the safest condition. For example, Figure 51 indicates a potential rollover to the left. Note that the situation descriptor 'most dangerous' (when the warning level reaches ± 1), is essentially a subjective value that must be selected by a combination of experience and through trial and error. For example, how does a warning value of ± 1 translate to the amount of time a driver has to react to the situation? This would need to be determined with further research that is beyond the scope of this work.

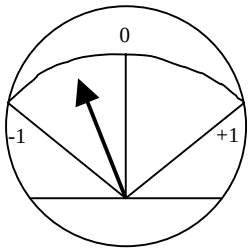


Figure 51: Example of a dial gauge indicating severity and direction of a potential rollover situation

V.2.1 FLIS Development using the Mamdani Inference Method

Using the Fuzzy Logic Toolbox from Matlab, a combination of Triangular and Trapezoidal MFs were chosen for their simplicity to be used for the two inputs, LTR and LTR rate, and the single output, Warning Level. Since the inputs to this FLIS are numeric and not linguistic, we do not have to assign a predefined numeric range as described in Step 0 of the Fuzzy Logic section in this treatise. However, the output warning was arbitrarily chosen to range between ± 1 and the MFs will be designed to reflect this.

Input/Output 'fuzzy' categories

From a requirement of keeping the FLIS initially simple but also providing a certain degree of resolution in fuzzification of the input variables and defuzzification of the output variables, a total of five different membership categories were chosen:

1. Negative Big (NB)
2. Negative Small (NS)
3. Zero (ZERO)
4. Positive Small (PS)
5. Positive Big (PB)

Note, however, that although the names of the categories are the same for inputs LTR and LTR rate, and output Warning Level, the exact shape of the categories are described by the respective MFs and are specific to the individual inputs and output. For example, the NB category for the LTR is quite different from the NB category for the LTR rate because the membership functions describing them are different.

Figure 52 is the main window within the Fuzzy Logic Toolbox that allows the user to adjust various properties of the FLIS system, including the type of Implication, Aggregation and Fuzzification method along with having complete control over how the AND and OR logic functions are defined.

Input 1: LTR

Figure 53 is the palette used for describing the exact shape of the five Triangular and Trapezoidal MFs for the input LTR. Notice that the boundaries for fuzzification of the input LTR are set at –1 and +1, which is the range that this input may span. Also notice that the MFs are symmetric about the ZERO category. This makes sense since the vehicle we are trying to use the RWD with can make symmetric maneuvers, either to the left or right, with essentially equal amounts of LTR.

The Trapezoidal MFs at the extremes (NB and PB) were tailored to provide a maximum output (1) when the LTR was $\pm$0.8. This was chosen as a type of 'safety factor' that provides a more cautious warning system since by doing so effectively increases the overall output Warning Level. Furthermore, the Triangular MFs were skewed towards Zero and the bases widened to help smooth some transitions within the Decision Surface, which will be described in more detail later.

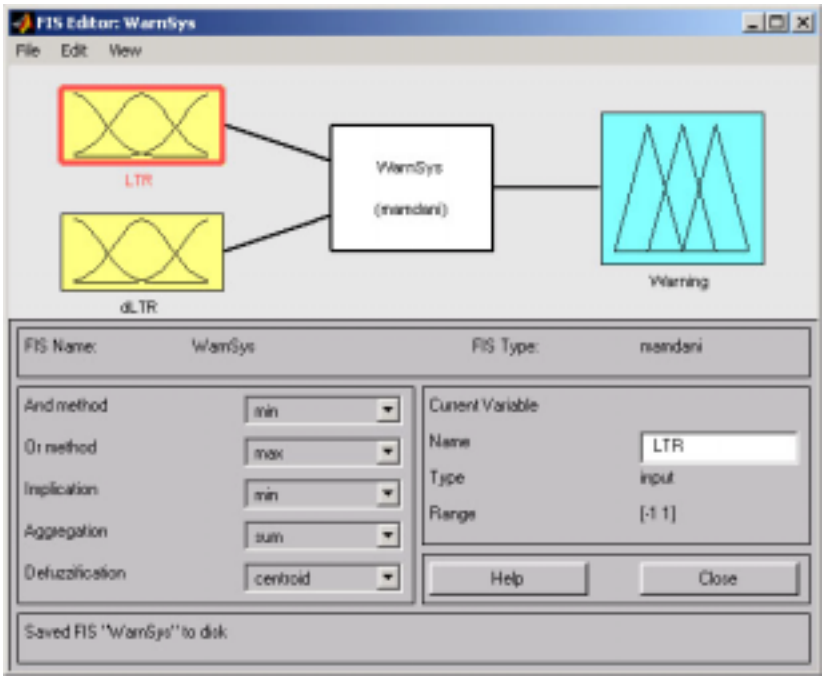


Figure 52: Main window of the Fuzzy Logic Toolbox (Mamdani Inference Method)

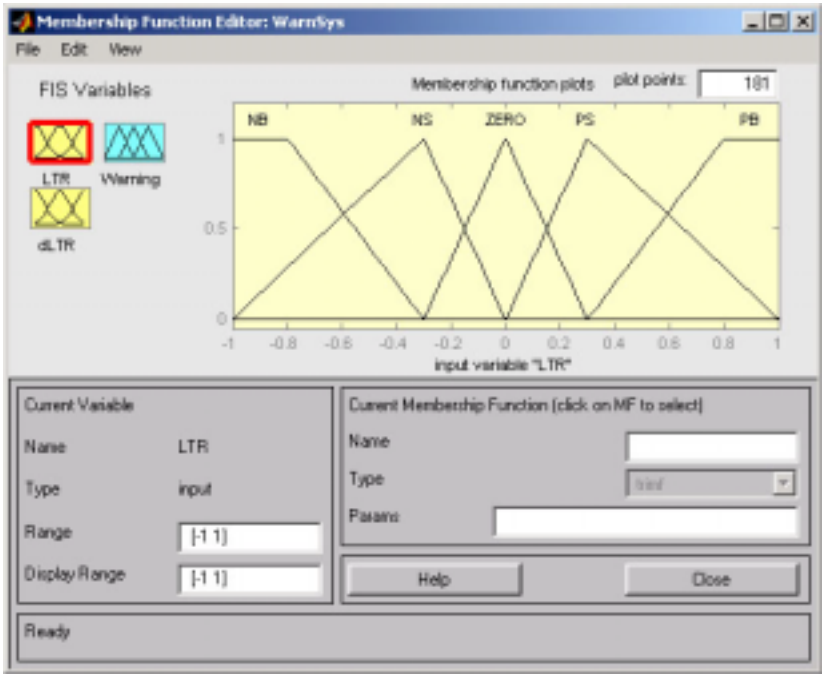


Figure 53: Fuzzy Logic Toolbox palette describing MFs for Input LTR (Mamdani Inference Method)

Input 2: LTR rate

In contrast to the LTR, which has a specific range of possible values that fall between ±1, the LTR rate can literally be anything. However, from analyzing the vehicle maneuver data used for training and testing of the ANNs, a feel for what constitutes a 'Big' or 'Small' LTR rate may be obtained. Hence it was chosen that an LTR rate begins to get 'Big' at ±0.4 and quickly rises to be fully 'Big' at ±1. The open-ended Trapezoid MFs at the extremes ensure that any LTR rate larger than ±1 is considered 'Big'. On the other hand, 'Small' was considered anywhere from 0 to ±1, while the ZERO category spanned ±0.4. Finally, the PS and NS membership functions have been slightly skewed toward the ZERO category to help smooth some transitions within the Decision Surface. Note that the LTR rate has been referred to as 'dLTR' in the simulation.

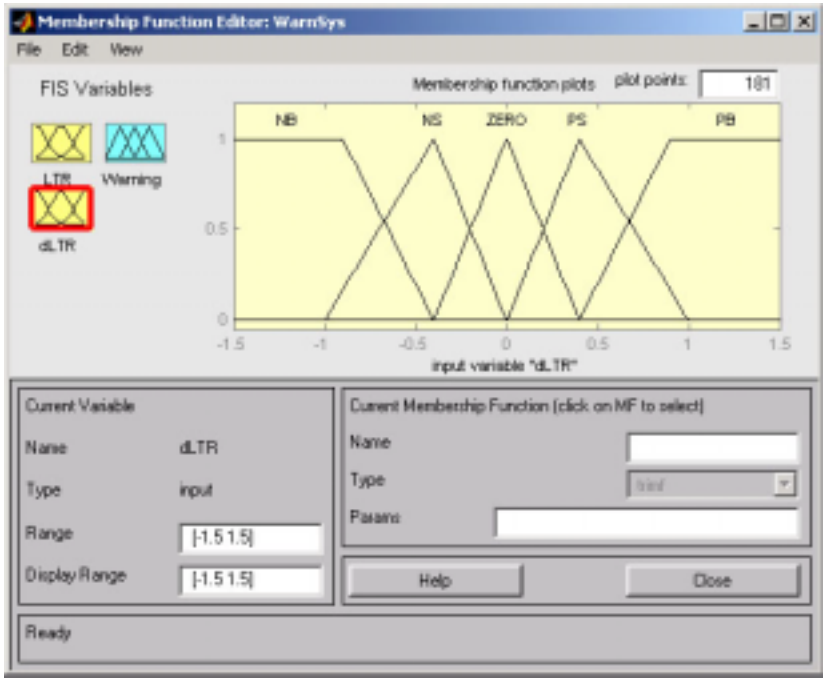


Figure 54: Fuzzy Logic Toolbox palette describing MFs for Input LTR rate (Mamdani Inference Method)

Output: Warning Level

Five Triangular MFs were used to describe the five categories that Warning Level could belong to as shown in Figure 55. They are all equilateral triangles with a base width equal to 1. Since we are using the Centroid Method for defuzzification (see Figure 52) and we want the output to be at most ±1, the NB and PB categories must be centered at ±1.

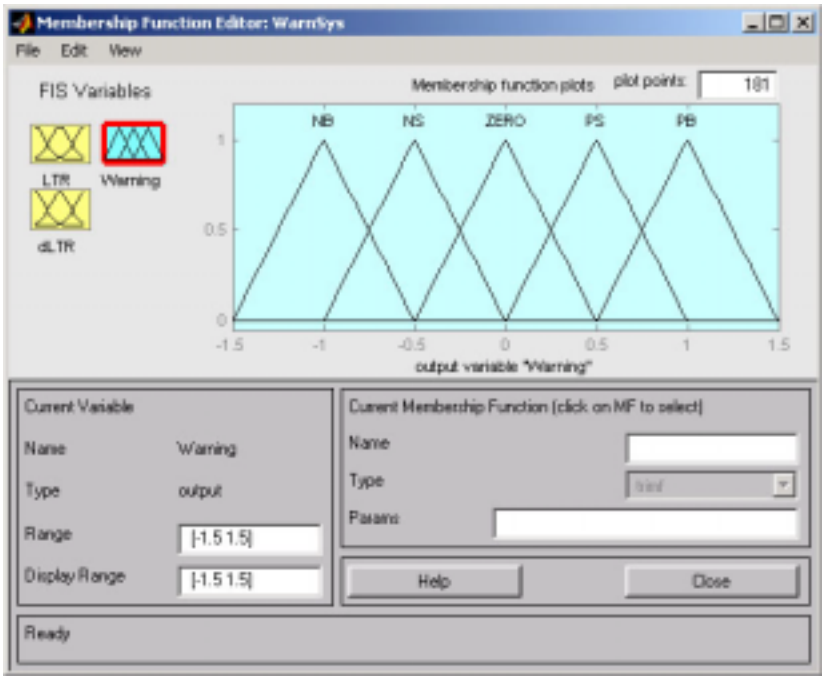


Figure 55: Fuzzy Logic Toolbox palette describing MFs for Output Warning Level (Mamdani Inference Method)

Fuzzy Rules

Because there are five input categories and five output categories, the Rule Base for this FLIS system will require 5x5=25 different rules to be processed in parallel during the Implication and Aggregation steps. Though symmetry in the rules is technically not required for all fuzzy computing, in this case it will only make sense if the Rule Base is symmetric. The following 25 rules were developed using the engineering experience of the author:

1. If (LTR is NB) and (dLTR is NB) then (Warning is NB)
2. If (LTR is NB) and (dLTR is NS) then (Warning is NB)
3. If (LTR is NB) and (dLTR is ZERO) then (Warning is NB)
4. If (LTR is NB) and (dLTR is PS) then (Warning is NS)
5. If (LTR is NB) and (dLTR is PB) then (Warning is ZERO)
6. If (LTR is NS) and (dLTR is NB) then (Warning is NB)
7. If (LTR is NS) and (dLTR is NS) then (Warning is NS)
8. If (LTR is NS) and (dLTR is ZERO) then (Warning is NS)
9. If (LTR is NS) and (dLTR is PS) then (Warning is ZERO)
10. If (LTR is NS) and (dLTR is PB) then (Warning is PS)

11. If (LTR is ZERO) and (dLTR is NB) then (Warning is NB)
12. If (LTR is ZERO) and (dLTR is NS) then (Warning is NS)
13. If (LTR is ZERO) and (dLTR is ZERO) then (Warning is ZERO
14. If (LTR is ZERO) and (dLTR is PS) then (Warning is PS))
15. If (LTR is ZERO) and (dLTR is PB) then (Warning is PB
16. If (LTR is PS) and (dLTR is NB) then (Warning is NS))
17. If (LTR is PS) and (dLTR is NS) then (Warning is ZERO)
18. If (LTR is PS) and (dLTR is ZERO) then (Warning is PS)
19. If (LTR is PS) and (dLTR is PS) then (Warning is PS)
20. If (LTR is PS) and (dLTR is PB) then (Warning is PB)
21. If (LTR is PB) and (dLTR is NB) then (Warning is ZERO)
22. If (LTR is PB) and (dLTR is NS) then (Warning is PS)
23. If (LTR is PB) and (dLTR is ZERO) then (Warning is PB)
24. If (LTR is PB) and (dLTR is PS) then (Warning is PB)
25. If (LTR is PB) and (dLTR is PB) then (Warning is PB)

As with the membership functions, the Rule Base may easily be adjusted to fit just about any preference as to how conservative or liberal the RWD should be. For example, if a more relaxed RWD is desired, Rule 23 could be changed to as to output PS instead of PB for the inputs of LTR=PB and dLTR=ZERO (likewise Rule 3 would need to be changed to keep the Rule Base symmetric).

Decision Surface

The Decision Surface is a convenient way to visualize the FLIS. It is essentially a graphical representation of the FLIS, where it displays the output value for an infinite number of input values. Based on the FLIS developed in the preceding sections, with two inputs, one output and the 25 rules in the Rule Base, the Decision Surface is shown in Figure 56. Therefore, for any pair of inputs [LTR, dLTR], there is a corresponding output Warning Level found somewhere on the Decision Surface. As may be expected, changing any part of the FLIS, such as modifications to the Rule Base, MF shapes for the inputs and output, or number of descriptive categories may change the shape of the Decision Surface dramatically.

Although the Decision Surface makes sense for the most part and may be considered a satisfactory result, the surface is not as smooth as it should be. For the RWD, it really only makes sense if the surface is completely smooth. For example, there is no practical reason that the warning level should increase non-smoothly for a fixed LTR rate and linearly increasing LTR (and vice-versa), but this is exactly what we have. This is

essentially caused by the fact that we are using linear MFs and trying to form a nonlinear Decision Surface. However, it can be shown that using nonlinear MFs will not completely smooth the Decision Surface as we would like. The problem lies in the fact that we are using the Mamdani Method, which is not always the best type of inference for problems such as this where we desire completely smooth transitions on the Decision Surface.

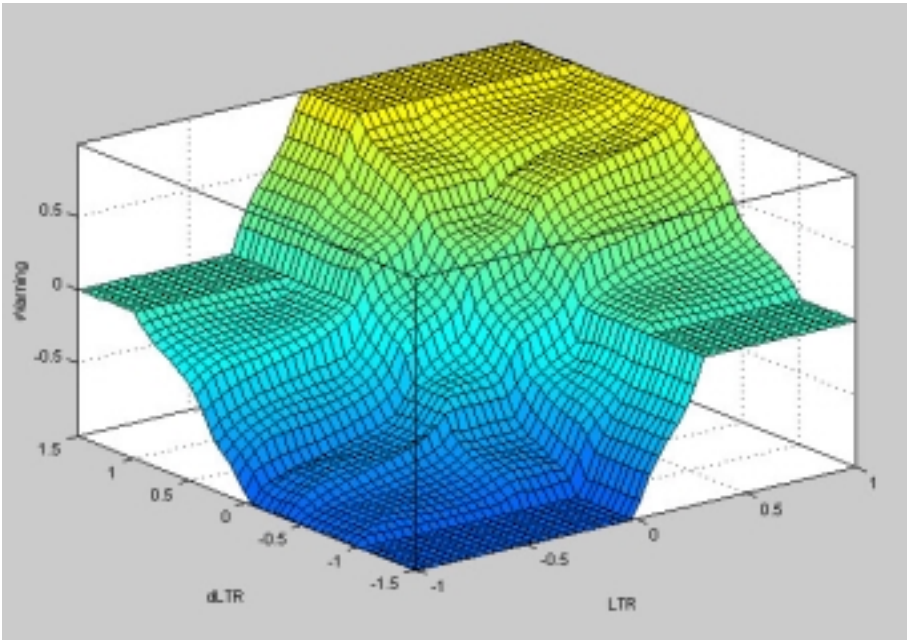


Figure 56: Decision Surface for the RWD-FLIS (Mamdani Inference Method)

V.2.2 FLIS development using the Sugeno Inference Method

Through trial and error, a much simpler FLIS system was developed using the Sugeno Inference Method where the Rule Base consisted of only six rules as opposed to the 25 with the Mamdani Method. The Rule Base consisted of only two input categories for the LTR, Positive and Negative and three categories Positive, Zero and Negative, for the input LTR rate and output Warning Level. The six rules are as follows:

1. If (LTR is N) and (dLTR is N) then (Warning is N)
2. If (LTR is N) and (dLTR is Z) then (Warning is N)
3. If (LTR is N) and (dLTR is P) then (Warning is Z)
4. If (LTR is P) and (dLTR is N) then (Warning is Z)
5. If (LTR is P) and (dLTR is Z) then (Warning is P)
6. If (LTR is P) and (dLTR is P) then (Warning is P)

Furthermore, the MFs were are all nonlinear Gaussian for both the LTR and LTR rate shown in Figure 57 and Figure 58, respectively. This allows completely smooth transitions on the Decision Surface shown in Figure 59.

Testing the RWD-FLIS

The maneuver data for the robustness testing of the ANN predictions in the previous section were used to examine the effectiveness of the RWD-FLIS based upon the Sugeno Inference Method. Figure 60 contains the output from the RWD-FLIS for the CRT maneuver at 60 km/hr. Note that the LTR and LTR rate were predicted using the 2-2 ANN with Ay-RR-RA as inputs. Furthermore, the actual LTR was overlaid to exemplify the amount of forewarning the RWD provides. It appears from Figure 60 that the ANN predicted data follows closely that of the actual data, which is to be expected since the 2-2 ANN did very well at predicting the LC maneuver at 60 km/hr. Furthermore, the RWD provides about a 0.5-1.0 second phase-lead over the actual LTR at the significant points during the maneuver. Another important aspect is that the warning level stays constant (at its maximum) although the LTR slightly decreases to approximately 0.8 after reaching at peak of approximately 0.85. This is due to the fact that RWD-FLIS was programmed to treat an LTR > 0.8 as extremely dangerous although the LTR rate may be zero or slightly decreasing.

The RWD was again tested using the data from the LC maneuver at 60 km/hr and the results shown Figure 61 with the respective actual LTR. Again, a phase lead of approximately 0.5-1.0 seconds is evident throughout the maneuver reaching a maximum warning level of 90% although the LTR only reaches a maximum of 0.3. This is due to the RWD-FLIS reacting to the high LTR rate for this maneuver.

Finally, the RWD is tested using data from the SRT maneuver at 60 km/hr with the results shown in Figure 62. Here, the RWD produces a warning that leads the LTR by approximately 0.25-0.5 seconds. Although the lead time is short and almost certainly too fast for an average driver to make a correction, the RWD was not expected to provide an adequate warning for such a violent maneuver where rollover (LTR=1) occurs so quickly. However, the RWD is successful at providing a maximum warning which precedes the vehicle reaching an LTR = 1 and furthermore stays there for the duration of the vehicle at its roll limit.

In conclusion, the FLIS-RWD is successful at ‘foreseeing’ a dangerous situation and seems to provide a warning proportionally. As expected, the most successful result was with the CRT maneuver since the LTR developed relatively slowly compared to the other two maneuvers and therefore allowed the most warning time of the three.

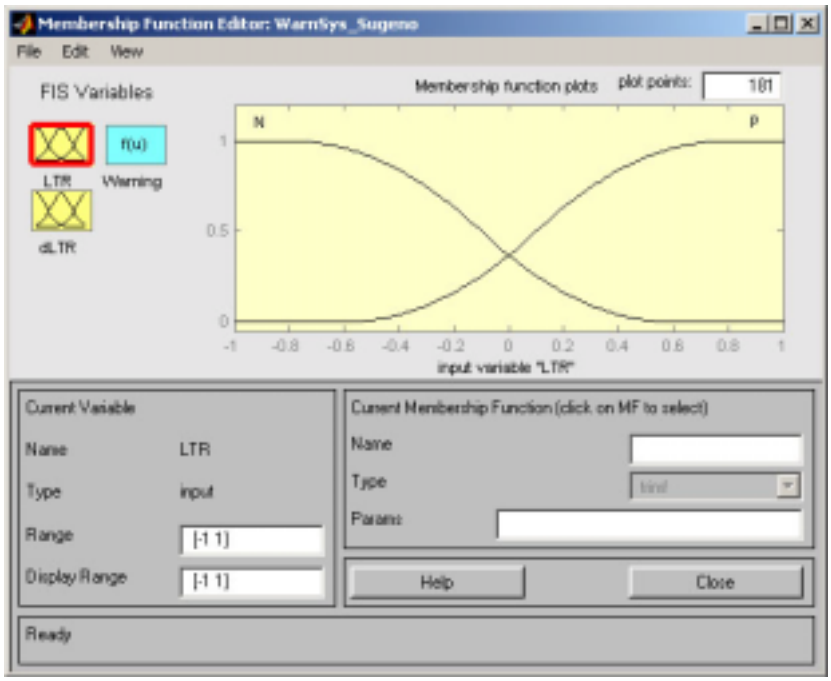


Figure 57: Fuzzy Logic Toolbox palette describing MFs for Input LTR (Sugeno Inference Method)

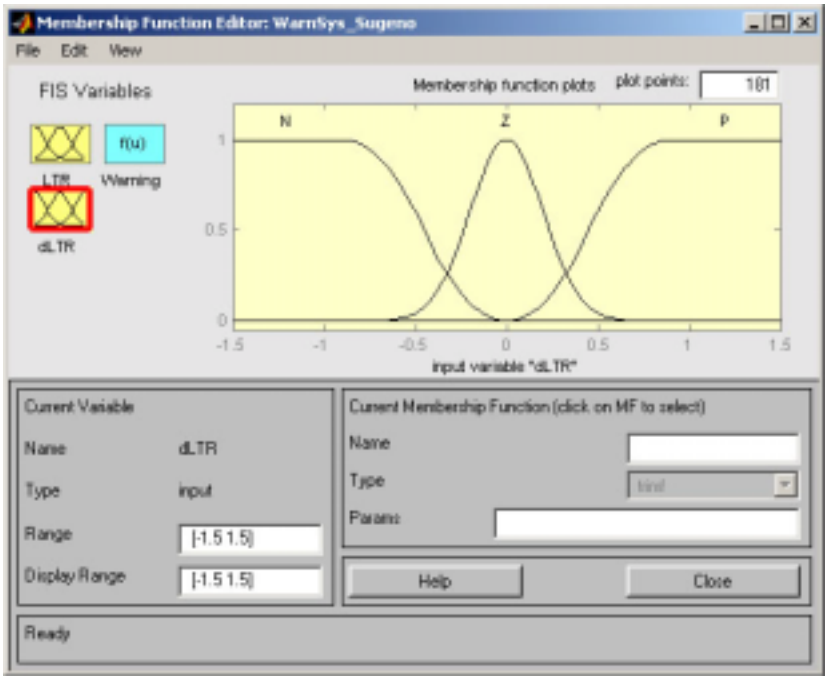


Figure 58: Fuzzy Logic Toolbox palette describing MFs for Input LTR rate (Sugeno Inference Method)

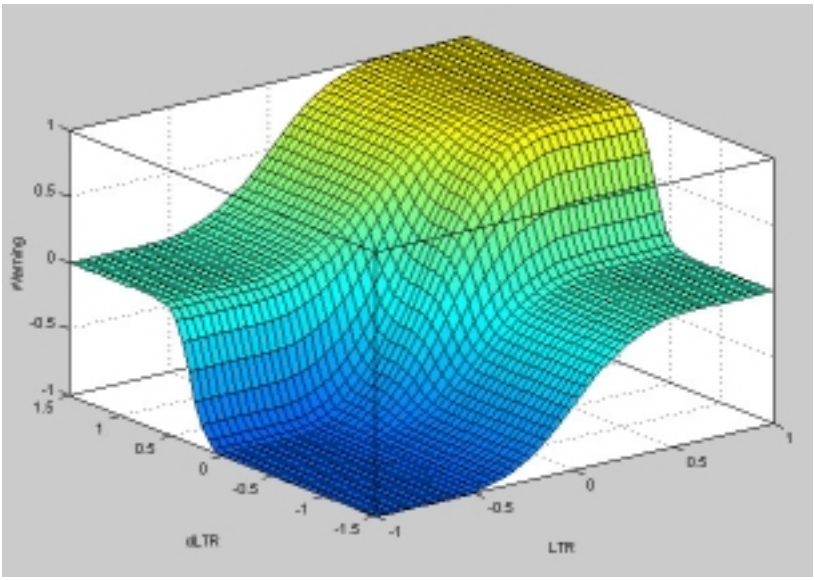


Figure 59: Decision Surface for the RWD-FLIS (Sugeno Inference Method)

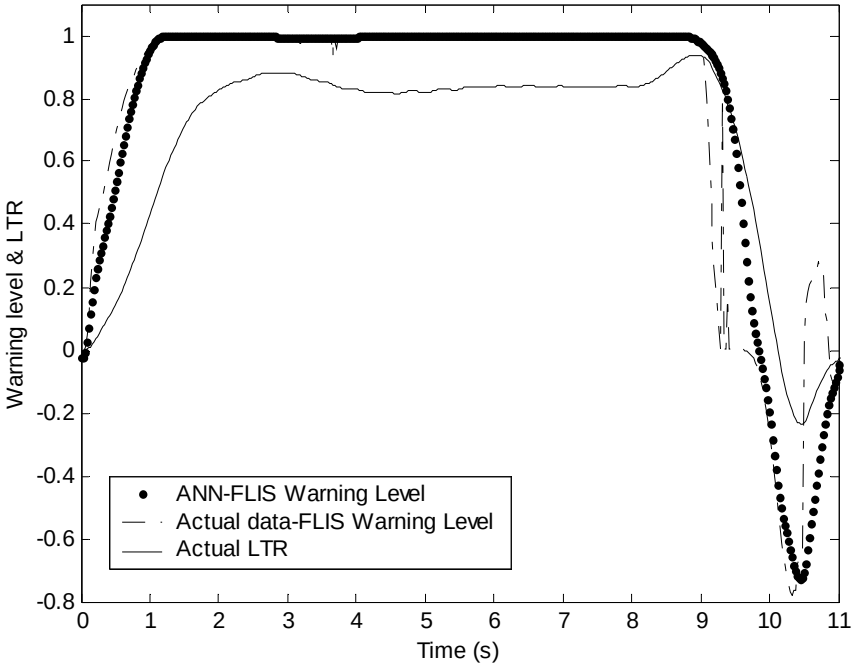


Figure 60: Warning level produced by the FLIS-RWD for the CRT maneuver performed at 60 km/hr

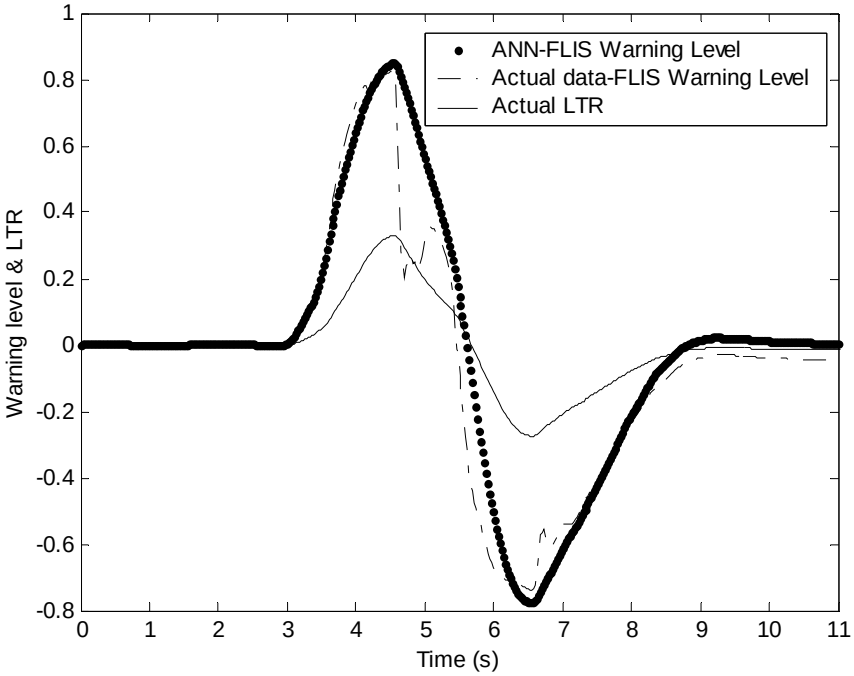


Figure 61: Warning level produced by the FLIS-RWD for the LC maneuver performed at 60 km/hr

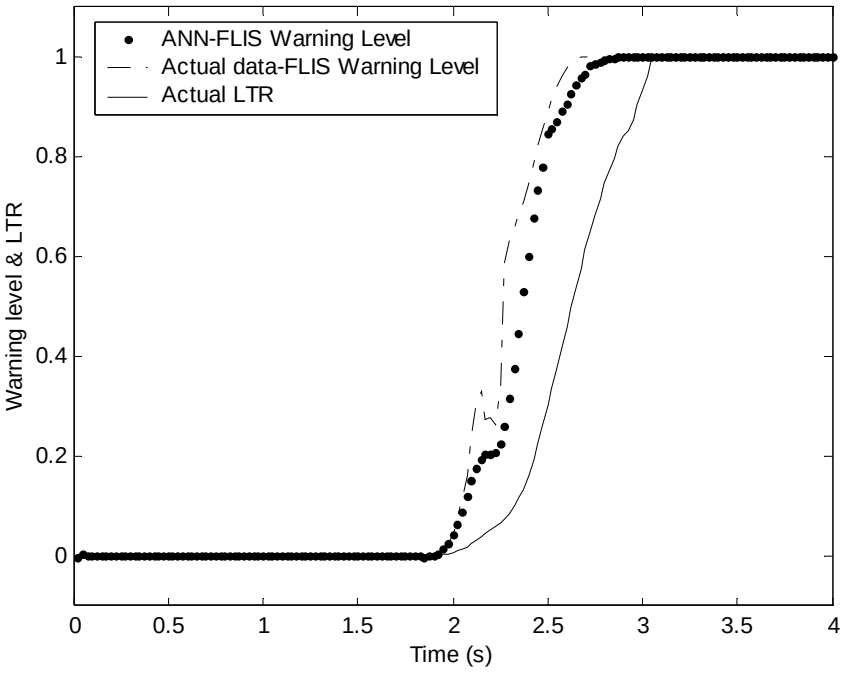


Figure 62: Warning level produced by the FLIS-RWD for the ST maneuver performed at 60 km/hr

VI. Experimental Results

In an attempt to validate the analytical results of the previous section, an experimental vehicle will be equipped with sensors to provide the necessary dynamic variables, such as lateral acceleration, yaw rates, etc. for a set of maneuvers. Due to limited funding and the difficulty that exists with measuring dynamic wheel loads experimentally, TruckSim will be used to provide the LTR and subsequent LTR rate data, which will be used to train the ANN. For TruckSim to provide realistic LTR and LTR rate information for a specific maneuver, accurate vehicle parameters such as weights, moments of inertia, suspension properties, etc., must be either measured directly or accurately estimated for use with the TruckSim model. The procedure for doing so will be explained in the following sections.

VI.1 Experimental Vehicle

The vehicle used for the experimental portion of this research was a 2001 Ford F450 XL Super Duty chassis with an aluminum dump bed manufactured by the Imperial Truck Body & Equipment Company. The vehicle is shown in Figure 63 and Figure 64. The truck was ideal for a number of reasons including a relatively high CG due to the weight dump bed and for the fact that the vehicle was new and various dimensional, mass and suspension properties were readily available on the internet (not to mention that the vehicle was available for testing at the PTI test track).

VI.1.1 Known vehicle parameters

As mentioned previously, for TruckSim to accurately determine dynamic LTR information for the experimental vehicle, the model needs to be updated with accurate parameters that define the Ford F450 dump truck. Specifically, TruckSim requires knowledge of the following:

- Sprung and unsprung masses
- Sprung and unsprung moments of inertia
- Suspension properties (rates and damping)
- Tire properties
- Steering properties
- Sprung mass and unsprung mass CG heights
- Miscellaneous dimensional data (wheel base length, roll center heights, etc.)

Many of the vehicle parameters were obtained directly from Ford via the Internet and by contacting the manufacturer of the dump bed to determine the bed mass. Thankfully, Ford publishes most of the dimensional, weight, steering and suspension properties for every new truck they sell and make this information readily available to the public. Other properties were relatively easy to measure directly such as suspension and damper

Figure 63: Side-view of experimental vehicle (2001 Ford F450)

Figure 64: Front-view of experimental vehicle (2001 Ford F450)

locations, vertical tire stiffness, and lateral and longitudinal CG positions using static wheel load measurements taken with heavy-duty force transducers. However, some of the most important unknowns, such as vertical CG height, cornering stiffness of the tires, and separate sprung and unsprung mass and moment of inertia properties are all very difficult (and typically expensive) to measure in practice. The way with which these unknown parameters were estimated will be explained in the following section.

VI.1.2 Unknown vehicle parameters

Mass & Moment of Inertia

To aid in the estimation of the separate sprung and unsprung mass and moments of inertia, the entire vehicle was modeled in the Mechanical Desktop CAD software package. Mechanical Desktop is based upon the popular AutoCAD software package, but also allows specification of material properties for the calculation of mass and moments of inertia.

All the major components were modeled individually such as wheels, axles, brakes, frame rails, engine, dump bed, etc., then pieced together at the end for overall mass and moment of inertia calculations. The Mechanical Desktop model was then adjusted to match the lateral and longitudinal CG positions measured experimentally while keeping the overall weight of the vehicle constant. Figure 65 and Figure 66 is a diagram of the finished model from Mechanical Desktop. Although this model took months to develop, what resulted is what the author feels is the most feasible method of determining the various mass properties of the vehicle and still able to yield a fairly accurate result. The expense in determining the mass and moment of inertia data experimentally would be enormous, not to mention time consuming.

Tire Properties

Tire properties in general are considered by many to be the most difficult thing to predict analytical or measure experimentally. Tire companies, vehicle manufacturers and independent researchers spends enormous amounts of money trying to do so, and even then the results are problematic. For example, researchers can perform numerous, time consuming measurements, and spend a great deal of money to determine the handling characteristics for a specific tire to only then find that the results cannot necessarily be applied to a different tire of the same make and model. Of course measuring the handling characteristics for each of the six tires on the experimental vehicle is beyond the scope of this research, tires do have certain characteristics and are similar for a limited variety of tires. Therefore, the default tire data found in the TruckSim model, such as the longitudinal force plots shown in Figure 67 (originally obtained from SAE research) will be used and adjustments made where necessary. The actual method of adjusting the data will be explained in a later section.

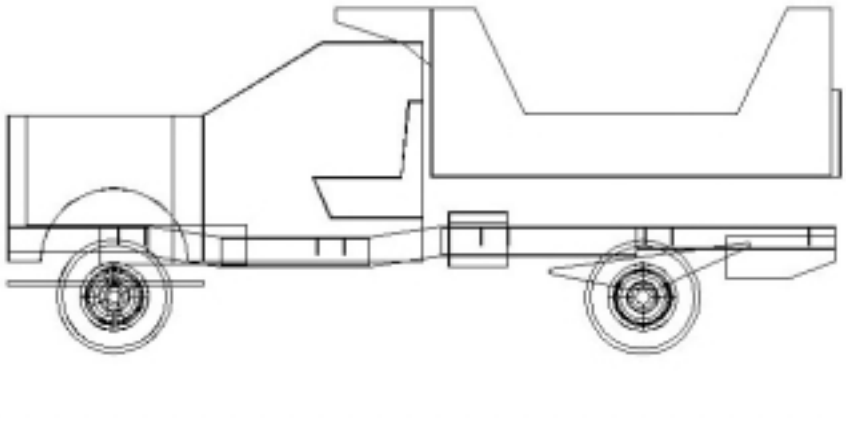


Figure 65: Side view of Mechanical Desktop model of Ford F450 dump truck

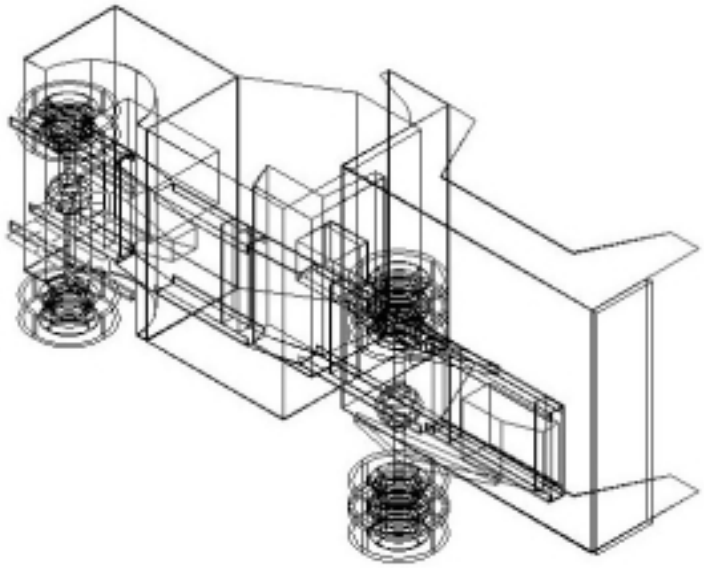


Figure 66: Right-isometric view of Mechanical Desktop model of Ford F450 dump truck

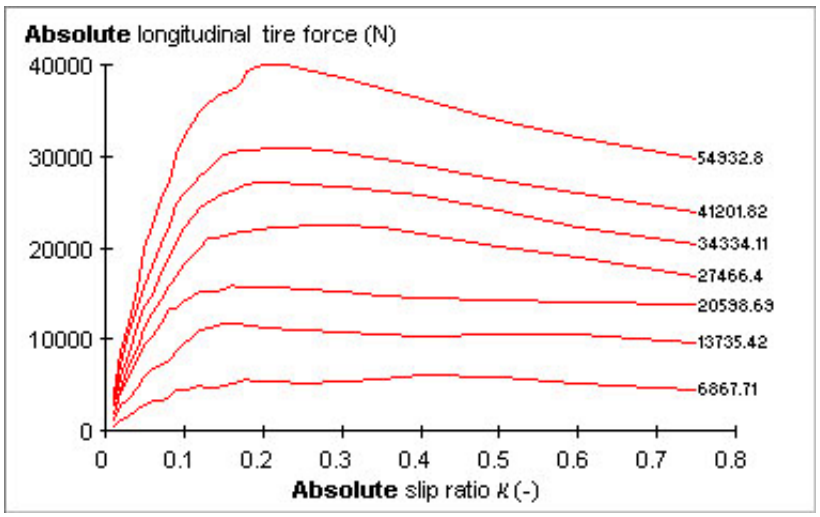


Figure 67: SAE longitudinal force tire data used in TruckSim

VI.1.3 Summary data of vehicle parameters

The following table summarize vehicle data either obtained experimentally (measured), estimated using Mechanical Desktop or obtained directly from Ford specification sheets.

Table 7: Mass and moment of inertia properties of the Ford F450

	Sprung Mass[4]	**Unsprung Mass[4]**	**Front Axle[4]**	**Rear Axle[4]**	**Wheel[3]**
Mass[1]	3263	887	381	506	50
Roll MOI[2]	1682	382	119	247	N/A
Yaw MOI[2]	11749	2949	119	247	N/A
Pitch MOI[2]	11263	3300	N/A	N/A	5.22

[1]units of kg
[2]units of kg-m^2
[3]measured
[4]estimated (Mechanical Desktop)

Table 8: Dimensional data of the Ford F450 truck

	(m)
Wheel Base Length	3.578
SM CG Height	1.148
SM Lateral CG	0.010
SM Longitudinal CG	1.741

	Front (m)	Rear (m)
Track Width	1.752	1.880
Suspension spacing	0.991	1.041
Damper spacing	1.168	1.041
Roll Center Height	0.425	0.432
Tire Spacing	N/A	0.279

Table 9: Dynamic parameter data for the Ford F450 truck

Tire vertical stiffness[1,5]	490
Wheel Rolling Radius[2,5]	0.378

	Front	Rear
Suspension rate[1,6]	82.3[3]	166.2/293[4]
Damping rate[3,7]	2.8	6.3

[1]units of kN/m
[2]units of m
[3]units of kN-s/m
[4]after auxiliary suspension engages
[5]measured
[6]obtained from Ford
[7]from SAE data

VI.2 Maneuver Descriptions

The primary maneuver, which was used to provide all training inputs for the neural networks, was the lane-change path. The reasons for using the lane-change path were two-fold: one, the lane-change maneuver is what all the analytical results were based upon, and two, both a right-to-left and a left-to-right lane-change path was available at the PTI test track. The details of the PTI lane-change paths (right-to-left is shown in Figure 68) were carefully measured and modeled for use in TruckSim. The accuracy of the input path for the TruckSim model was considered very important as the TruckSim LTR (and LTR rate) results would be used to train the neural networks.

A secondary maneuver, used for examining the steering and roll properties of the vehicle, was a 91.4 m (300 ft) circle-turn shown in Figure 69. This path was also conveniently located and laid out at the PTI test track. As will be explained in a later section, determining the flexibility within the steering system was important to successfully following the simulated lane-change path in TruckSim. Furthermore, due to discrepancies in the acquired roll data during the lane-change maneuvers, the circle-turn was used to test and subsequently verify that the measured roll angle data was faulty.

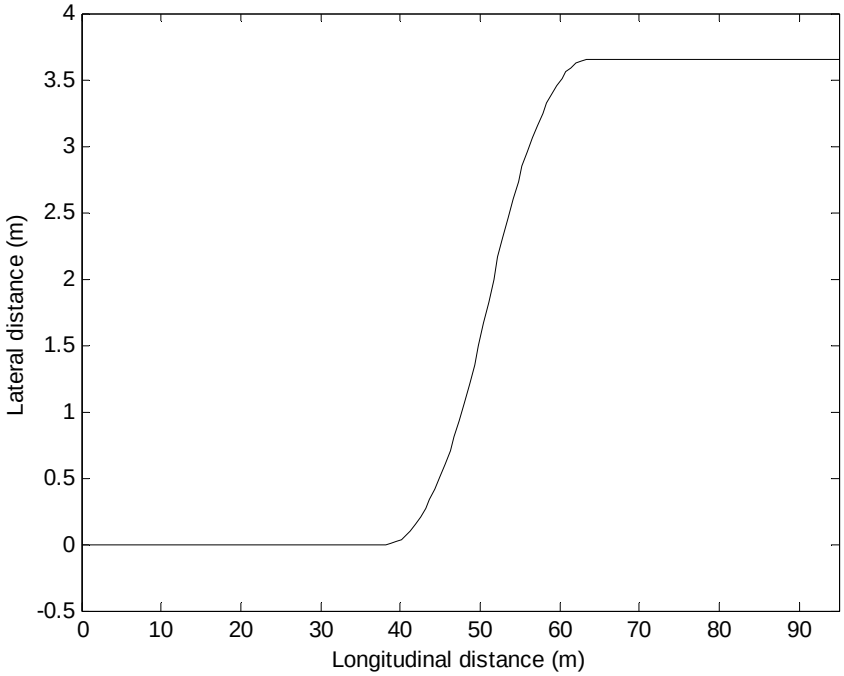


Figure 68: PTI right-left lane-change maneuver

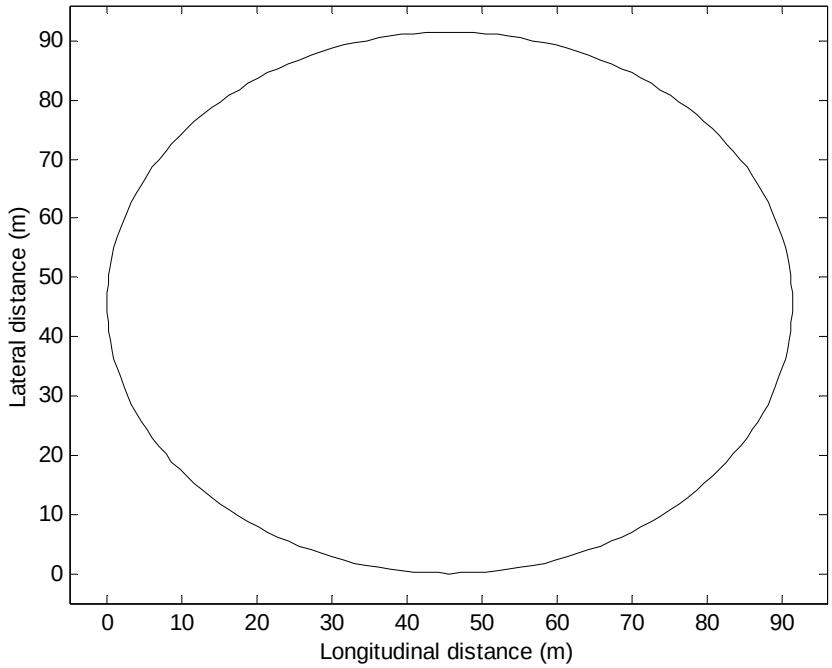


Figure 69: PTI 91.4 m (300 ft) circle-turn maneuver

VI.3 Data Acquisition

VI.3.1 Experimental setup

The DMU (gyroscope) was attached to the centerline of the vehicle within the truck cab as shown in Figure 70. Careful attention was paid to mounting the DMU on a flat surface and oriented properly with the longitudinal and lateral axes of the vehicle. Once the DMU was oriented visually, a level was used to make small corrections to the orientation and then fastened down. The orientation of the DMU was frequently checked during the course of a day of testing. Furthermore, the lateral, longitudinal and vertical position of the DMU within the vehicle was carefully measured. Knowledge of the DMU position within the vehicle is vital if the simulation results using TruckSim may be compared to those acquired through the DMU. Fortunately, TruckSim has the ability to output its variables relative to any arbitrary point within the vehicle and is not restricted to the CG.

Once the DMU was in its correct position, the power was switched on and the serial interface was attached to a laptop within the cab running the data acquisition software. The DMU was allowed to warm-up for a approximately 15 minutes before any testing took place. It was noticed that data acquired before it was able to warm up was inconsistent.

Figure 70: Location of the DMU within truck cab

VI.3.2 Testing

After practicing driving the maneuver a number of times and at different speeds, data acquisition took place. The lane-change maneuvers were performed at 8 km/hr (5 mi/hr) increments from 32.2 to 72.4 km/hr (20 to 45 mi/hr). 72.4 km/hr seemed the maximum speed that still allowed for consistency in the data, not to mention safety. A total of three tests were run at each speed increment. The reverse lane-change (left-to-right) was then completed at the same speeds and speed increments, resulting in a grand total of 36 tests (6 speed increments x 2 directions x 3 tests).

Results for the three lateral acceleration and yaw rate measurements at 64.4 km/hr (40 mi/hr) for the right-to-left lane-change are shown in Figure 71 and Figure 72, respectively. The data appears very consistent and relatively free of noise. All other tests performed with the right-to-left maneuver, except for the maneuvers at 40.2 km/hr (25 mi/hr) were similar in their consistency and lack of noise or scatter. However, the left-to-right lane-change data seemed very inconsistent. The author attributes this to the fact that the left-to-right path is more difficult for the driver to see and therefore follow correctly. The driver, located on the left side of the vehicle, must essentially guess where the path is where as the right-to-left path is easier to see and follow. The inconsistencies were particularly evident at higher speeds. It is not clear as to why the data acquired at 40.2 km/hr (25 mi/hr) is so inconsistent.

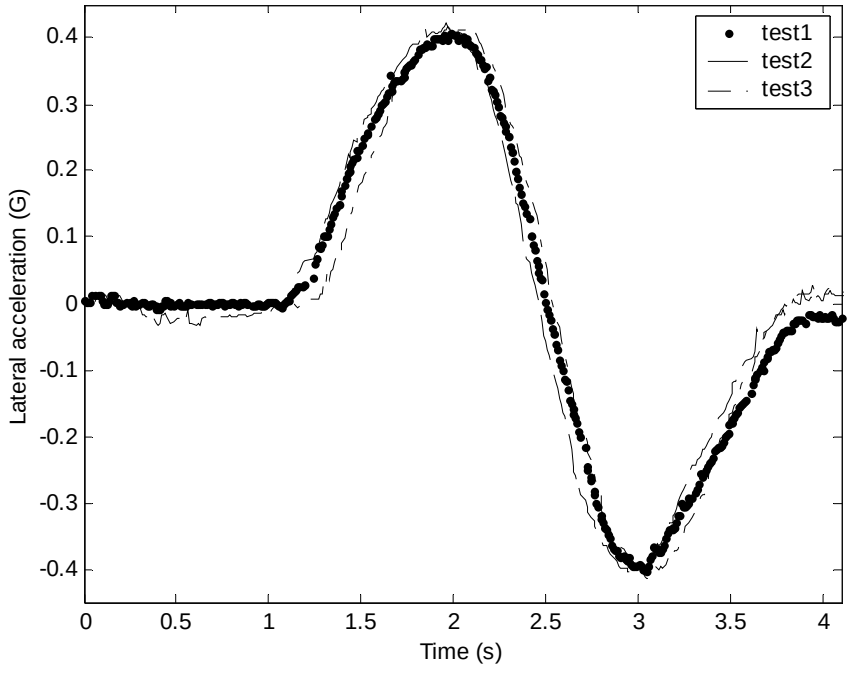


Figure 71: Experimental lateral acceleration measurements for the PTI lane-change maneuver at 64.4 km/hr (40 mi/hr)

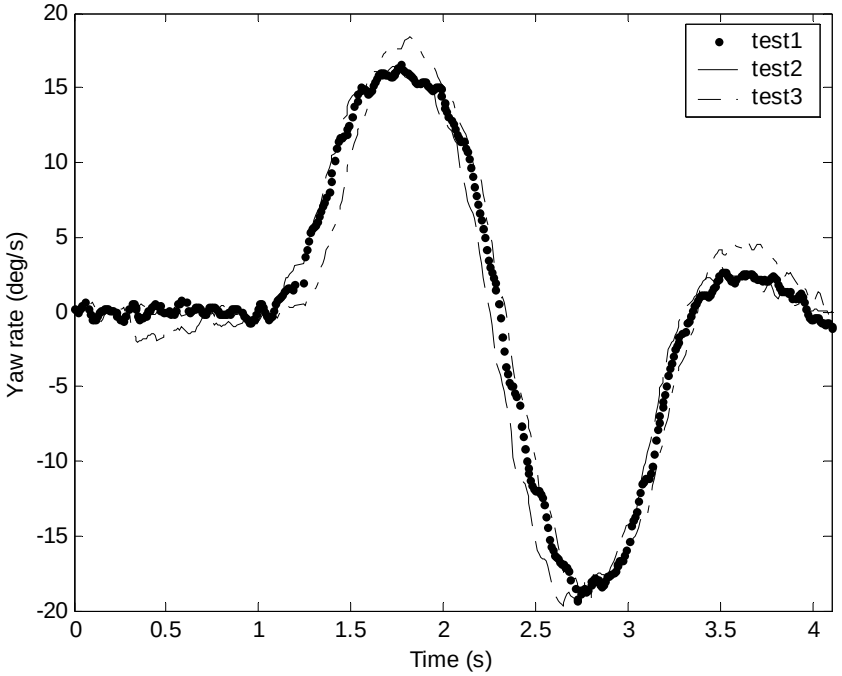


Figure 72: Experimental yaw rate measurements for the PTI lane-change maneuver at 64.4 km/hr (40 mi/hr)

VI.4 Analysis of Experimental Results

VI.4.1 Obtaining the 'experimental' LTR and LTR rate via Trucksim

After the experimental data has been acquired, the next step is to determine the load-transfer properties for the specific maneuvers, i.e., determining the LTR and LTR rate to be used for training the neural networks. This involves modeling not only the vehicle and maneuver in TruckSim, but also attempting to match the driver's reactions during the maneuvers. Due to limitations in equipment (not to mention driving skills), knowledge of the exact path followed experimentally is essentially unknown, though clues to the exact path followed are held with the lateral acceleration and yaw rate information. Matching the (unknown) path followed experimentally to the simulated path followed by adjusting the driver model within TruckSim proved to be one of the most difficult and time consuming tasks for the analysis section of this research.

Driver model

Initially, simulations were run using the default driver model within TruckSim on the lane-change maneuvers to get a feel of the driver model. TruckSim's driver model consists of a closed-loop control scheme that utilizes a preview time and a driver lag time, both which can be modified by the user. The preview time is simply how far ahead the TruckSim's 'driver' looks at the path to be followed. The lag time tries to model the amount of time the driver takes to perform an action after the control decision has been

made. As the preview time increases while keeping the lag time constant, the driver attempts to 'smooth out' the lane-change, thereby decreasing lateral acceleration and yaw rate. Decreasing the preview time has the opposite effect since the driver can only make a control decision based on the path information directly in front of the vehicle. Furthermore, if the lag time is too long and the preview time is too short, the driver reaction becomes unstable as it tries to follow a quickly changing path with a delayed reaction.

Path radius of curvature

Matching the lateral acceleration and yaw rate are the two most important tasks to determine the actual path followed. This is because the radius of curvature for a path (which essentially *defines* the path) is determined by the ratio of lateral acceleration to yaw rate squared, or

$$r = \frac{Ay}{\omega^2}.$$

Therefore, comparing the calculated radius of curvature from the experimental data to the curvature determined by the simulated data creates a visual and analytic tool to aid in modifying the driver model (and for 'tweaking' various vehicle properties). For example, Figure 73 displays the TruckSim result of the experimental vehicle performing the PTI right-to-left lane-change maneuver at 48.3 km/hr (30 mi/hr) using the driver model with 0.9 second preview time and zero lag time. The driver cannot follow the prescribed path exactly, but does the best it can, based upon the performance of the control algorithm with the given preview and lag time.

Using the lateral acceleration and yaw rate data generated by TruckSim for this maneuver along with the experimentally measured data, a plot of the two radii of curvature may be compared like that shown in Figure 74. A sense of how the experimental driver actually performed the maneuver can be inferred from the diagram. Note that only approximately the first half of the maneuver is shown in Figure 74. This is due to the fact that inconsistencies with the experimental driver and limitations to adjusting the TruckSim driver model prevented the *entire* maneuver from being matched. In other words, the experimental driver seemed to change the way he drove the maneuver *during* the maneuver. The TruckSim driver model is static in the sense that the parameters are constant throughout the entire maneuver and cannot be changed. Therefore, attention was focused on only matching the first-half of the maneuver. Consequently, only the first half of the LTR (and LTR rate) will be used for training.

Suspension model

During the course of testing, the roll data (angle and rate) obtained from the DMU seemed very inconsistent and did not match the results for the simulation though the suspension data was obtained directly from Ford. At first, it was assumed the suspension data (stiffness) for the front and rear leaf-springs must be incorrect. However, a constant-radius turning maneuver (circle-turn) shown in Figure 69 was performed using the DMU and a glass of water where the roll angle could be verified or refuted. The results indicated that the roll angle, and particularly the roll rate, were incorrect and therefore the

DMU was faulty about that axis. Furthermore, the roll angle data measured with the glass of water matched almost exactly the data obtained through simulation, verifying the stiffness properties obtained from Ford. The solution to faulty roll data from the DMU was to simply use the roll data produced in the TruckSim simulation. Therefore, all roll rate (and calculated roll acceleration) data used in training was acquired from the simulations once the experimental and simulated paths were matched.

Miscellaneous Model Adjustments

Tire properties

Along with attempting to match the radii of curvature plots like that in Figure 74, attempting to match the lateral acceleration and yaw rate plots directly between the experimental and simulated data proved invaluable. This was overall a trial-and-error process and miscellaneous model properties were adjusted to match the experimental data as close as possible. Though the author feels the mass properties of the vehicle are fairly accurate, a large uncertainty that still existed was with the tires. Again, through trial-and-error, tire properties (particularly cornering stiffness) were modified so as to produce the required lateral acceleration and yaw rate for the lane change maneuvers and at different speeds.

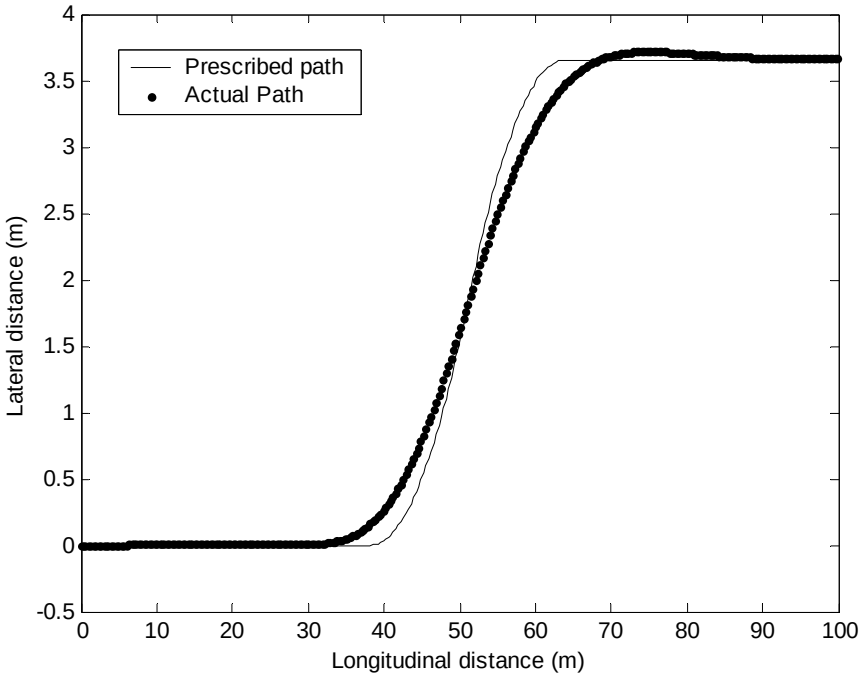


Figure 73: Front-axle tracking through the right-to-left lane-change performed by the simulated vehicle

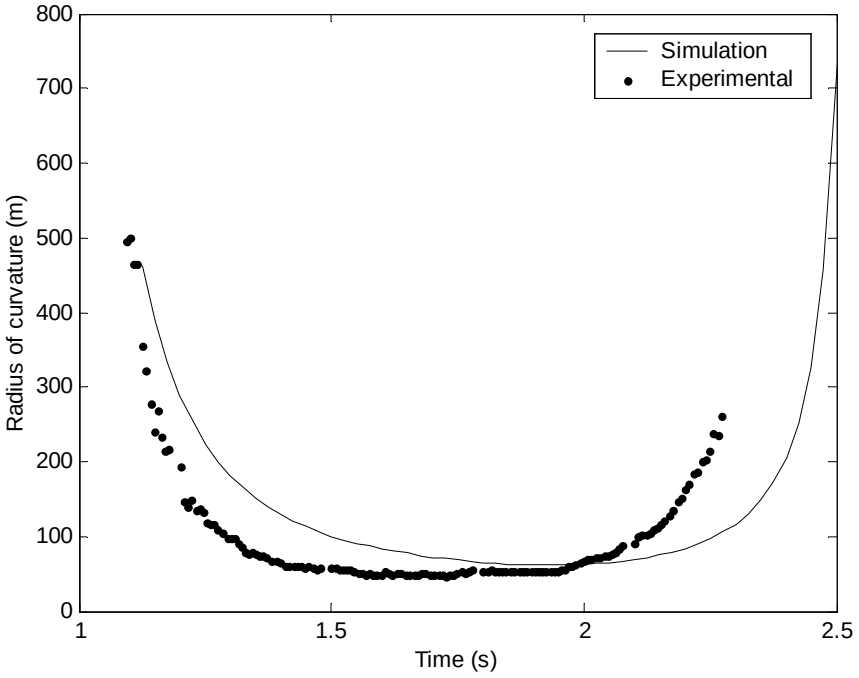


Figure 74: Radii of curvature for the first half of the right-to-left lane-change maneuver

Steering model and vehicle understeer

Another uncertainty existed that had a significant impact of matching the path was with the steering model; specifically compliance in steering system and the overall *understeer* of the vehicle. A vehicle is considered to have understeer properties if the front wheel steer angle must be increased for an increase in speed during a constant-radius turn. On the other hand, a vehicle is considered *oversteer* if the opposite occurs. If no change in steer angle of the front wheels is required, then the vehicle is considered to have *neutralsteer* properties. The amount of understeer in a vehicle is a function of the weight distribution and cornering stiffness of the tires. The test to determine this is called an Understeer test and may be performed a variety of ways [6]. One of the simplest is to perform a number of constant-radius path follow maneuvers at increasing speed increments while recording the steering wheel angle. From the steady-state cornering equation found in the Literature Review section and repeated here, the Understeer coefficient (K_u) may be determined by calculating the slope of the change in front-wheel steering angle to change in lateral acceleration when the data is obtained during a constant-radius turning maneuver.

$$\delta_f = \frac{L}{R} + \frac{K_u A_y}{g} \tag{7}$$

Though the equation was defined for use with the steer angle of the front wheel and not the steering wheel angle, Gillespie [6] suggests that the steering wheel angle *should* be used because compliance in the steering system *is* a major part of the understeer properties for a vehicle. Therefore, the understeer of a vehicle is not only a function of

the weight distribution and tire properties, but of the compliance in the steering system as well.

An understeer test was performed with limited success for this research. Problems arose in keeping the vehicle at a constant speed while following the constant-radius path and recording a steering wheel angle. However, the test did reveal a direct measure of the compliance in the steering system, which was then used for the TruckSim model.

Model Results

As can be seen in Figure 75 TruckSim was able to match the lateral acceleration almost perfectly, where the 'simulation' line is difficult to discern from the experimental data. The yaw rate simulation data shown in Figure 76 was typically off in phase compared to the experimental, but was able to match the overall shape very well. This author could not correct this in TruckSim. The other simulation results for the different speeds were similarly accurate.

At first thought, one might want to simply shift the yaw rate data over to eliminate the phase lag independent of the lateral acceleration to then be used for training. However, this would be incorrect since the LTR data, used as the targets for ANN training, would not be correct since they were calculated in the same time base as the lateral acceleration and yaw rate data within TruckSim. The resulting trained ANN would not correctly model the experimental vehicle. To summarize, the author feels confident that the LTR (and LTR rate) and close to the real values.

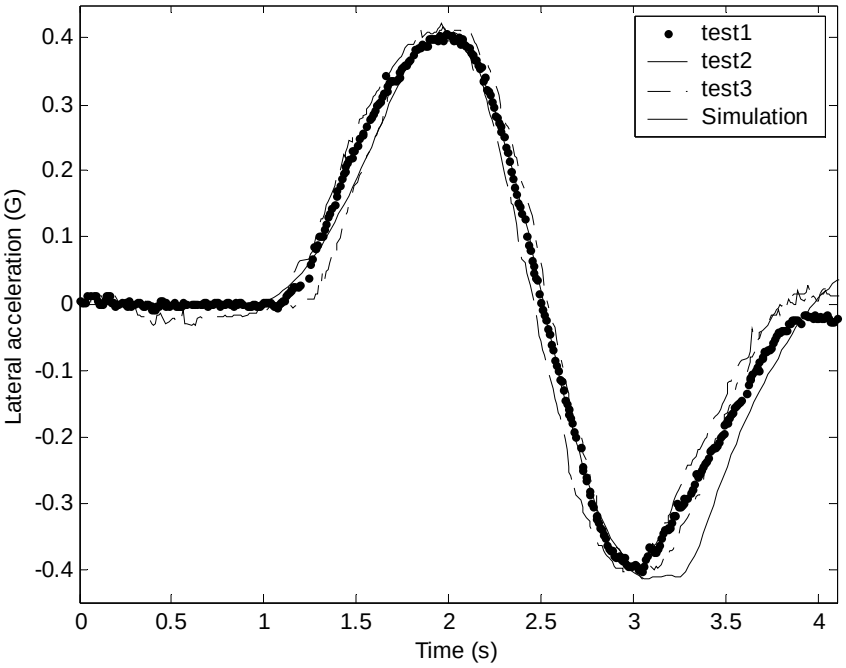


Figure 75: Experimental and simulated lateral acceleration measurements for the PTI lane-change maneuver at 64.4 km/hr (40 mi/hr)

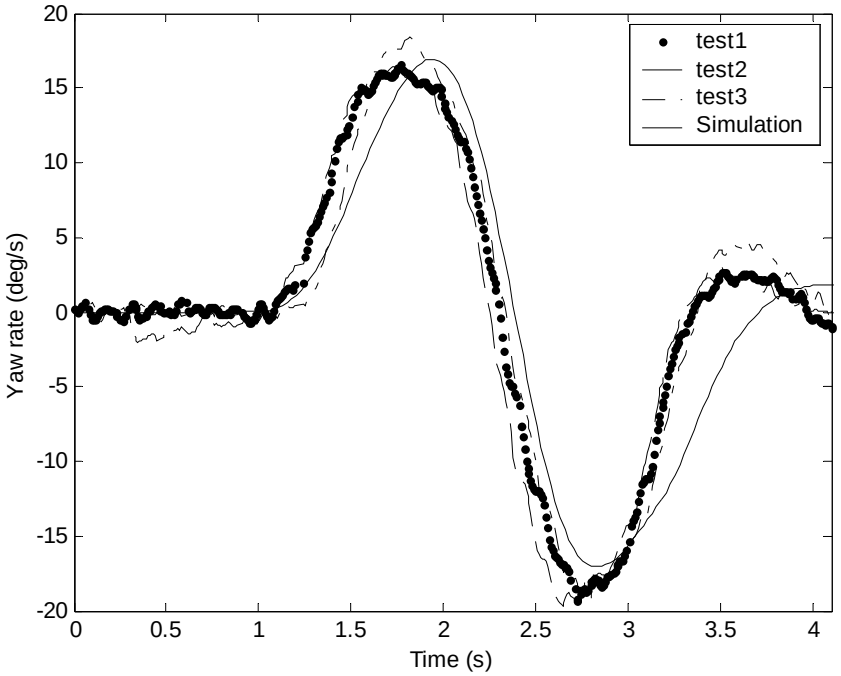


Figure 76: Experimental and simulated yaw rate measurements for the PTI lane-change maneuver at 64.4 km/hr (40 mi/hr)

VI.4.2 ANN training

Once the author felt that the TruckSim model was successful at predicting experimental results, all the lane-change maneuvers simulations were run and LTR properties recorded. The LTR and LTR rate will act as the target data for the ANN to predict using the experimental inputs. As mentioned previously, tremendous difficulty existed in trying to model the *entire* path and therefore only the first half of the lane-change maneuver data will be used for training. Furthermore, the left-to-right maneuver data was too inconsistent to be used for training and therefore, the vehicle will be assumed to be symmetric in all its properties (suspension, tires, etc.). The only exception to this assumption of a completely symmetric vehicle is with the slight offset of lateral CG position determined from the static wheel load measurements. This offset incurred at slight constant load transfer towards the drivers side of the vehicle. Therefore, when the right-to-left training data was inverted to act as the left-to-right training data, this static load-transfer was taken into consideration.

The normalized and concatenated LTR and LTR rate training data is shown in Figure 77. Recall that the order in which the network receives the training data is arbitrary as long as the input data coincides with the training data. In other words, the ANN simply receives input-output pairs and attempts to minimize the MSE of the overall data.

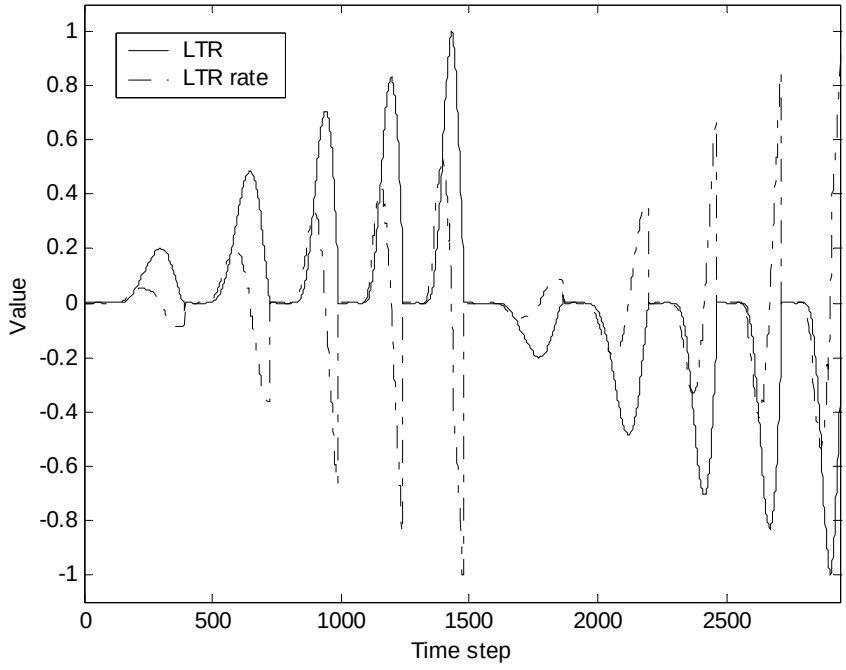


Figure 77: Normalized and concatenated training data (LTR and LTR rate) for the various lane-change maneuvers

Networks selected for training

Two network architectures, Ay-RR-RA and Ay-YR-RR-RA performed best overall during the analytical robustness tests and were selected to attempt to learn the LTR and LTR rate from the experimentally measured data. However, both of these architectures include the roll acceleration, which requires the computation of a derivative that will impart a time delay and will manifest itself through to the warning device. In other words, having to compute the derivative in order to predict a LTR and LTR rate for use with the RWD will delay the output of the device by the amount of time the derivative is calculated for. If the roll rate data is noisy, the derivative will have to be computed over a few sample times to help smooth the result. However, the individual sample times with accumulate and may have an affect on the performance of the RWD. To help alleviate this problem, it is important to minimize the noisy or sporadic roll rate information. To circumvent the problem of having to compute a derivative completely, a third network consisting of the inputs Ay-YR-RR will be trained on the data to examine its effectiveness. Therefore, three networks will be used to attempt to learn to predict the LTR and LTR rate by using experimental input data and are as follows:

- Ay-YR-RR
- Ay-RR-RA
- Ay-YR-RR-RA

Input data smoothing

The average of the three tests for performed at each speed increment for the lane-change maneuver was calculated, then smoothed using a moving average. The moving average simply finds a 'best fit' of the overall experimental data. Of course, the data smoothing only takes place with training data and will not be expected to take place with new input data. An example of how the data smoothing changed the data may be seen in the 'noisy' averaged lateral acceleration data in Figure 78. Each data point is averaged and replaced with sixteen of its closest neighbors; eight in front and eight behind. The typical result is shown in Figure 79 where the data has the appearance of being 'cleaned up'.

Training

The results for training the three networks with different input architecture types of are shown in Table 10. The number of neurons in the hidden layer was varied, as before, from 2 to 20. All the networks performed excellently, which is evident from the small error (all MSE less than 10^{-3}) and visually as the LTR and LTR rate predictions using the Ay-YR-RR inputs are shown in Figure 80 and Figure 81, respectively. Note that these plots are the result of the *worst* of the training results with only two neurons in the hidden layer.

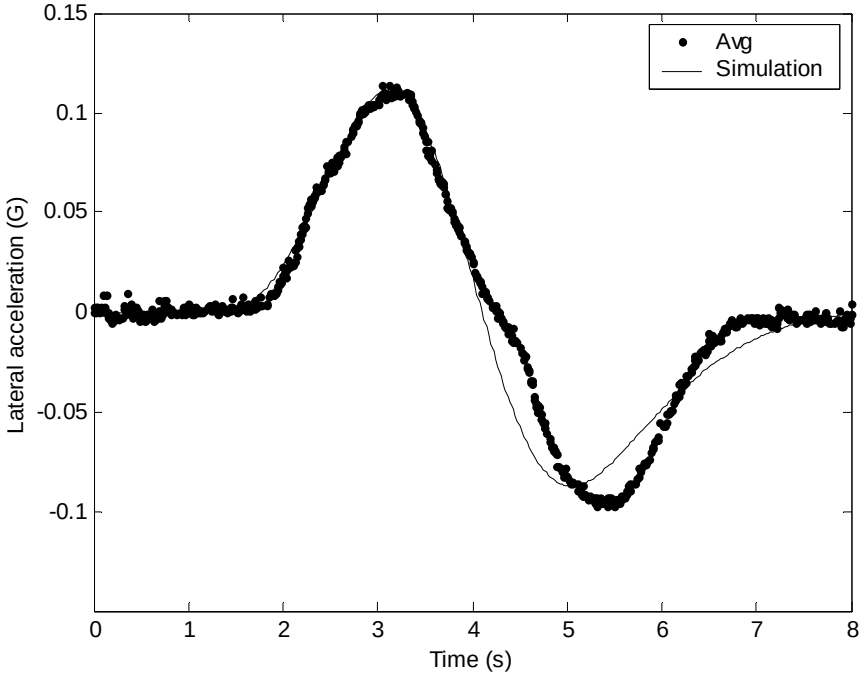


Figure 78: Relatively noisy lateral acceleration data

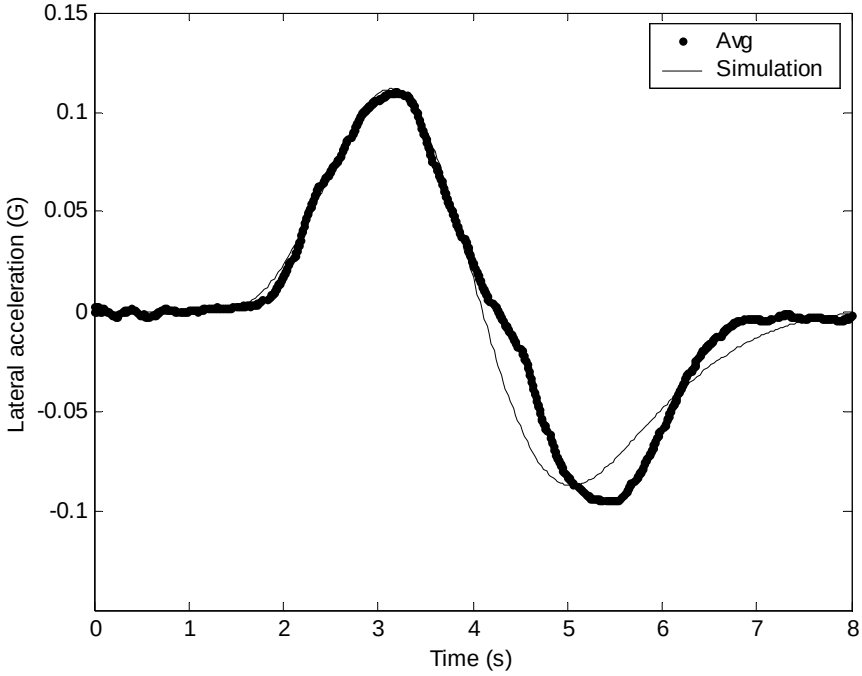


Figure 79: Lateral acceleration data after a sixteen-step moving average was applied

Table 10: Training results using experimental input data

	Predicted LTR (MSE x 10^{-4})		
Neurons	**Ay-YR-RR**	**Ay-RR-RA**	**Ay-YR-RR-RA**
2	7.980436	3.101151	2.985194
3	7.801090	2.303800	2.946309
5	3.350584	2.200773	2.001331
10	0.942747	1.436442	0.803058
20	0.530084	0.619202	0.180763

	Predicted LTR rate (MSE x 10^{-4})		
Neurons	**Ay-YR-RR**	**Ay-RR-RA**	**Ay-YR-RR-RA**
2	2.515031	2.572190	2.277852
3	2.465073	1.820124	2.214555
5	2.169737	1.824495	1.435131
10	1.453605	1.125440	0.618924
20	0.560736	0.610319	0.543111

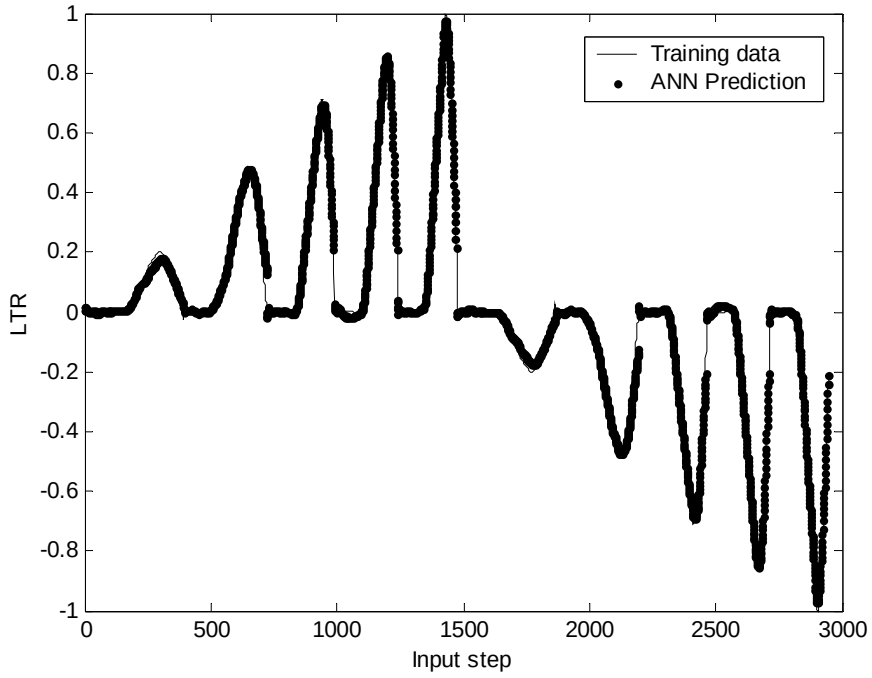


Figure 80: LTR prediction results using the experimental input data

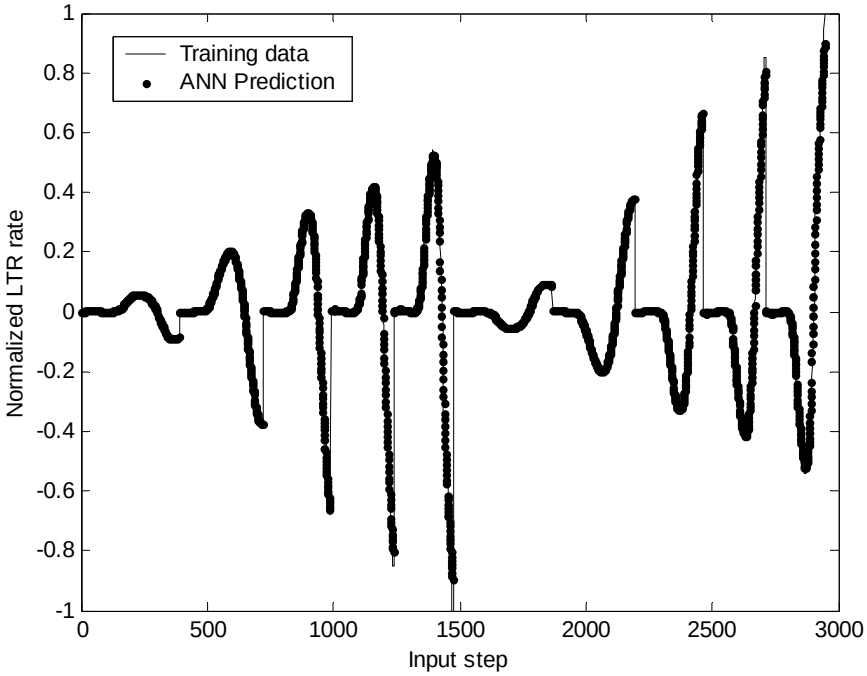


Figure 81: LTR rate prediction results using the experimental input data

VI.4.3 Robust tests

To determine robustness of the trained networks from the preceding section, input data from two maneuvers that the networks were not trained upon will be used to predict the LTR and LTR rate. Input data for the first maneuver, a left-to-right lane-change at 80.5 km/hr (50 mi/hr), was obtained experimentally with the DMU. Note that this lane-change is performed faster than any of the lane changes that the two networks were trained upon. Furthermore, this data is from a legitimate left-to-right lane-change where the training data involved having to invert the right-to-left data due to inconsistencies in performing the left-to-right maneuver.

The input data for the second maneuver, a constant-radius J-turn with increasing speed had to be simulated in TruckSim due to the difficulty and poor repeatability in performing such a maneuver using the equipment available. The J-turn with a 300-ft diameter is shown in Figure 82. To successfully perform the J-turn while increasing speed and also be able to accurately predict the LTR and LTR rate using TruckSim requires knowledge of the path and rate at which the velocity was changing. Although experimentally measuring the yaw rate and lateral acceleration during the maneuver would provide information on how the velocity was changing, simulating the same constant-speed change proved to be extremely difficult since TruckSim requires knowledge of input power to modify speed during a maneuver. Therefore it was decided to simply perform a J-turn within TruckSim and use the lateral acceleration, yaw rate and roll rate information determined by TruckSim as inputs to the networks. This may be considered acceptable since the TruckSim model parameters were altered so to match the experimental data obtained with the test vehicle. Therefore, it is assumed that the

dynamic variables obtained from TruckSim are relatively accurate and may be used with the trained network to predict the LTR and LTR rate. An added benefit of using the simulated data is the ability to allow the vehicle reach rollover.

Lane-change maneuver results

Figure 83 and Figure 84 show the LTR and LTR rate prediction results, respectively, using the 2-2 Ay-YR-RR network. Note that only about half of the maneuver is shown due to the inconsistency of the driver model as explained earlier. Figure 83 shows that the network performs reasonably well at matching the LTR overall, but seems to fail slightly at the extreme points in the maneuver (>0.5 LTR). However, the LTR rate prediction using this network is exceptional, following the TruckSim result almost exactly throughout the important parts of the maneuver. Furthermore, the network appears to be less susceptible to sporadic or noisy data at the input when predicting the LTR rate.

The results using the 2-2 Ay-RR-RA network are shown in Figure 85 and Figure 86. It is clear that the results are almost identical to those when using the Ay-YR-RR as inputs. Overall, the network performed slightly better at predicting the LTR, but a significant

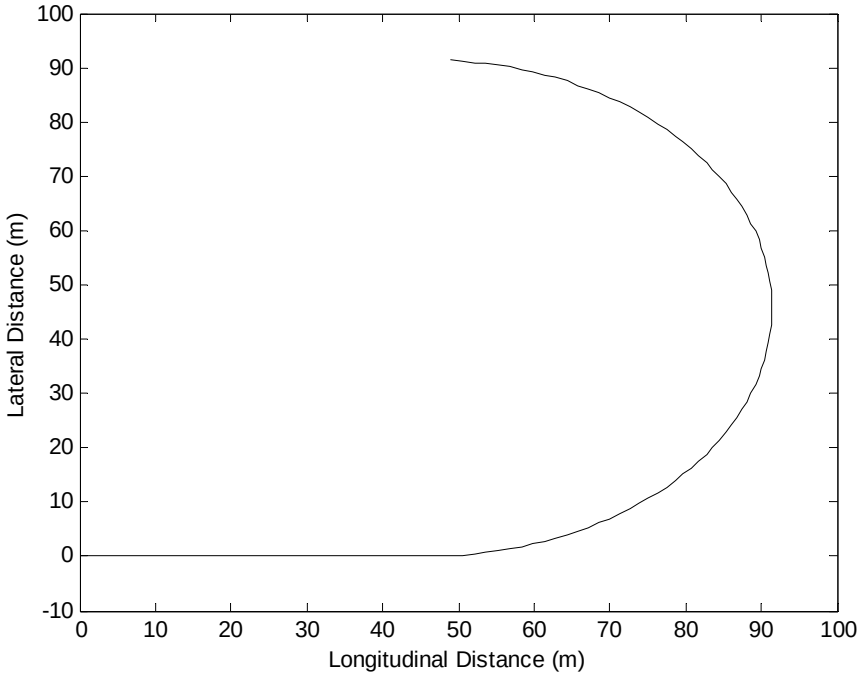


Figure 82: J-turn maneuver layout with 91.4 m (300 ft) diameter

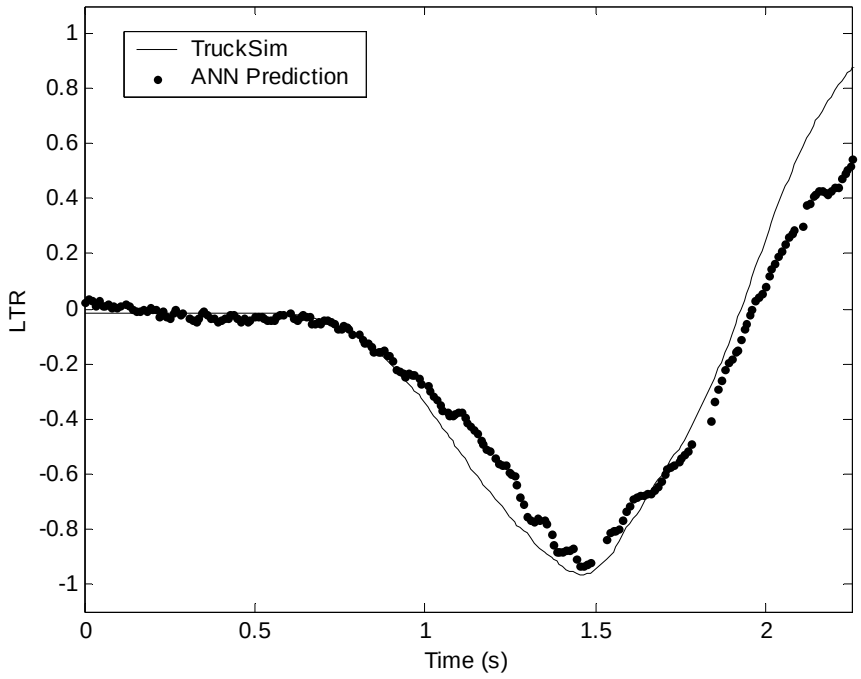


Figure 83: LTR prediction for left-right lane-change maneuver performed at 80.5 km/hr using the 2-2 A_y-YR-RR network

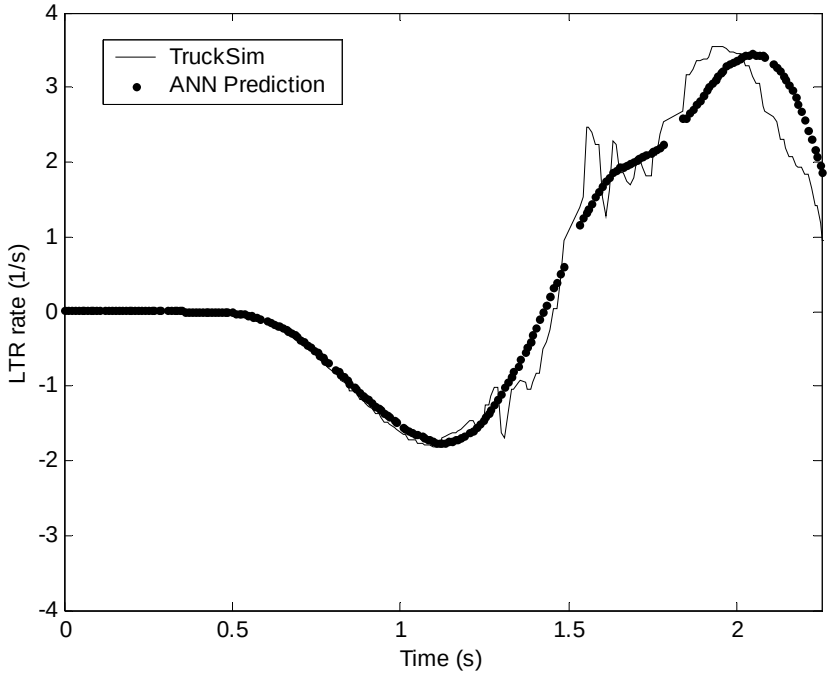


Figure 84: LTR rate prediction for left-right lane-change maneuver performed at 80.5 km/hr using the 2-2 A_y-YR-RR network

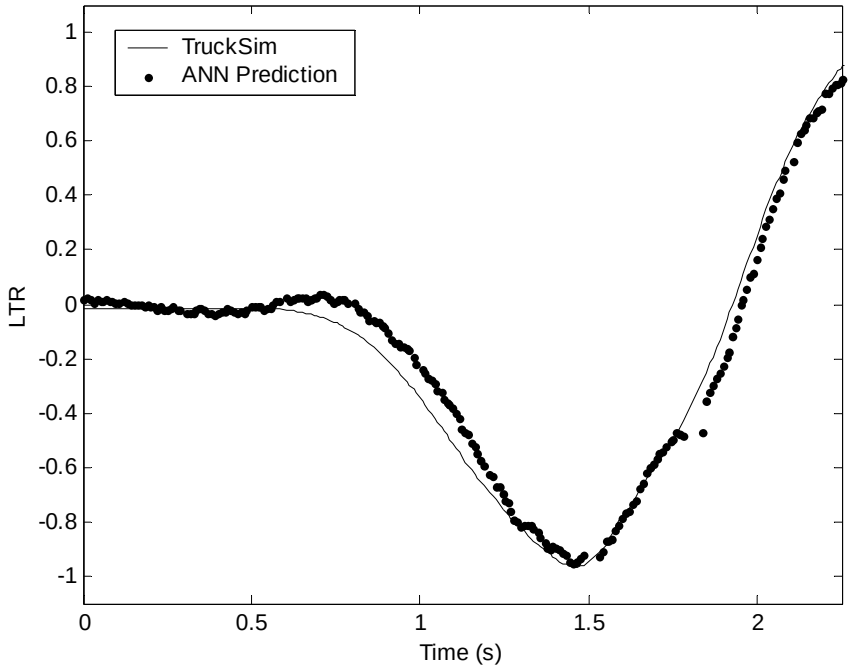


Figure 85: LTR prediction for left-right lane-change maneuver performed at 80.5 km/hr using the 2-2 A_y-RR-RA network

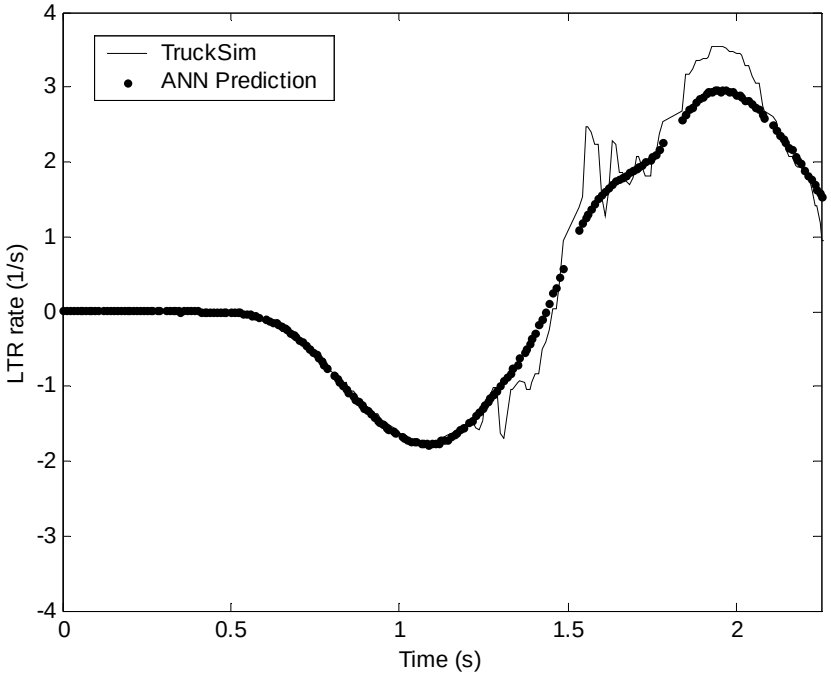


Figure 86: LTR rate prediction for left-right lane-change maneuver performed at 80.5 km/hr using the 2-2 A_y-RR-RA network

improvement was made near the extreme maneuver points (>0.5 LTR). The more accurate prediction in this regime would be very important for use in a warning device. Furthermore, the results or the LTR rate prediction using A_y-RR-RA network are similar to that of the A_y-YR-RR network being extremely accurate and just as resistant to sporadic or noisy data.

Figure 87 and Figure 88 display the results of predicting the LTR and LTR rate, respectively, with the slightly more complex network using all the experimental inputs available: A_y-YR-RR-RA with the 2-2 neuron architecture. The results for the LTR prediction are almost identical to that of the Ay-RR-RA with the differences being almost indiscernible. Likewise, the prediction of the LTR rate is similar to that predicted by the other three networks, again with differences being almost indiscernible.

In general, the addition of neurons in the hidden layer performed no better at predicting the LTR or LTR rate for this maneuver.

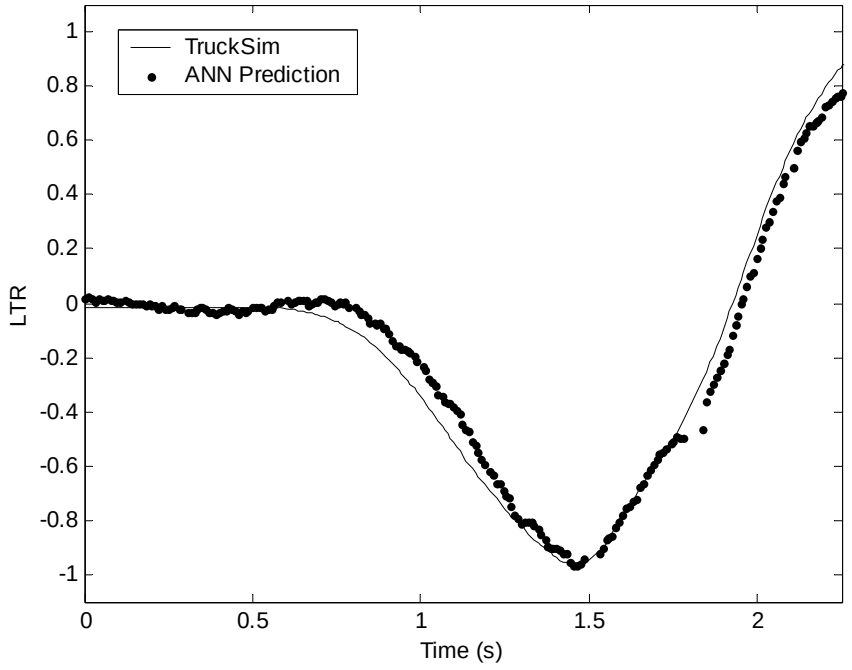


Figure 87: LTR prediction for left-right lane-change maneuver performed at 80.5 km/hr using the 2-2 A_y-YR-RR-RA network

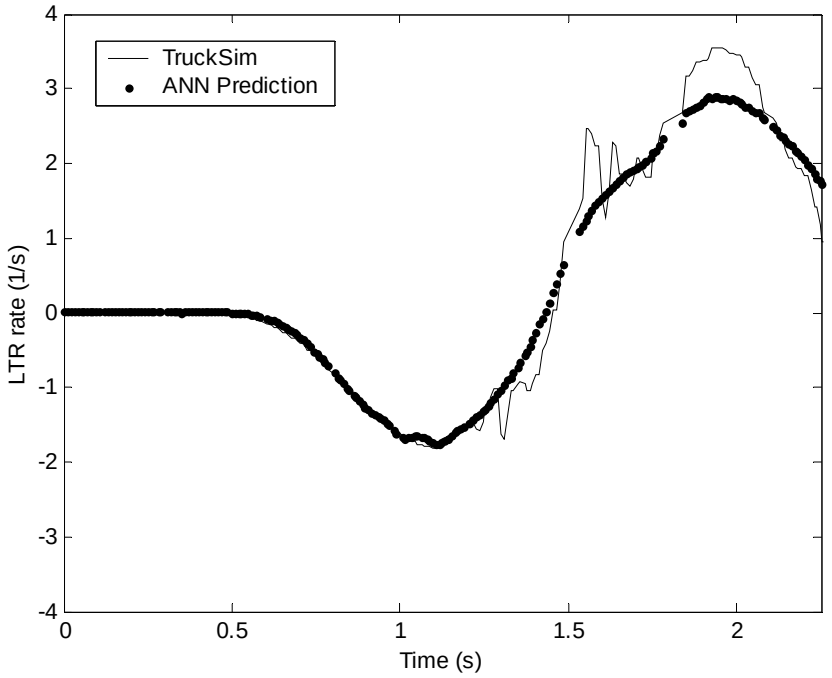


Figure 88: LTR rate prediction for left-right lane-change maneuver performed at 80.5 km/hr using the 2-2 A_y-YR-RR-RA network

J-turn maneuver results

The results shown in Figure 89 using the 2-2 A_y-YR-RR network to predict the LTR indicate that it is not able to deal properly with steadily or linearly increasing input values of the input variables. The network essentially 'knows' that the LTR should increase, but not sure exactly how. Therefore, it appears that using A_y-YR-RR is not enough information and/or the training data was insufficient for it to learn this maneuver with steadily increasing outputs. As with its performance on predicting the LTR rate on the high-speed lane-change maneuver, the A_y-YR-RR network performs exceptionally well as shown in Figure 90. The other architectures with an increased number of neurons in the hidden layer for this set of inputs performed no better and sometimes worse.

Figure 91 displays the results of using the 2-2 A_y-RR-RA network to predict the LTR. It is clear the network performs very well at following the shape of the actual LTR. This result is considered particularly important since the only realistic opportunity for a passive rollover-warning device to prove successful would be in a situation where the driver has adequate time to take corrective action, such as during a maneuver like this. As with the lane-change maneuver results, the LTR rate is predicted very accurately with this network.

The LTR prediction using the 2-2 A_y-YR-RR-RA network shown in Figure 91 performs even better than the former network tested. The predicted LTR follows almost exactly the actual LTR. Furthermore, the LTR rate is predicted very accurately.

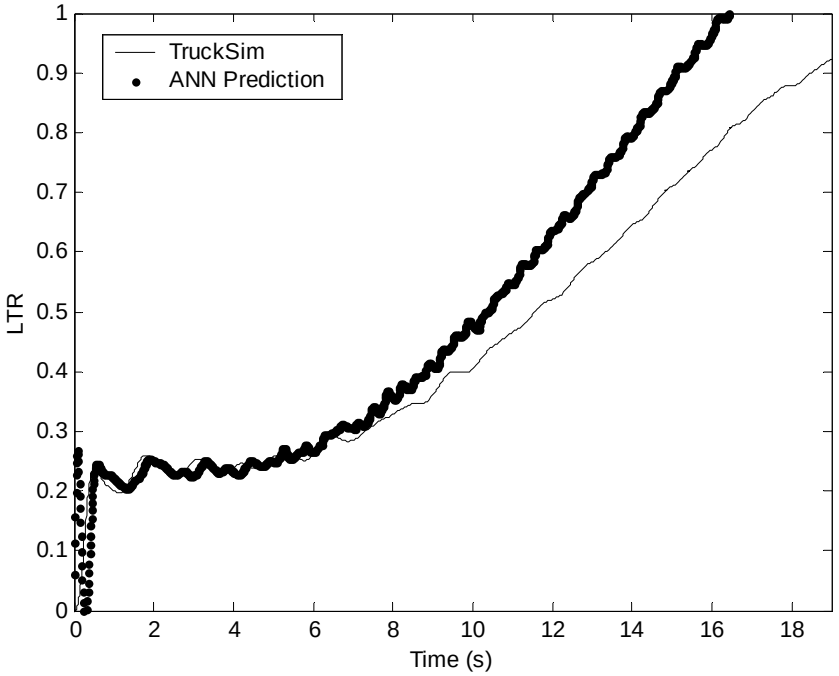


Figure 89: LTR prediction for the J-turn maneuver using the 2-2 A_y-YR-RR network

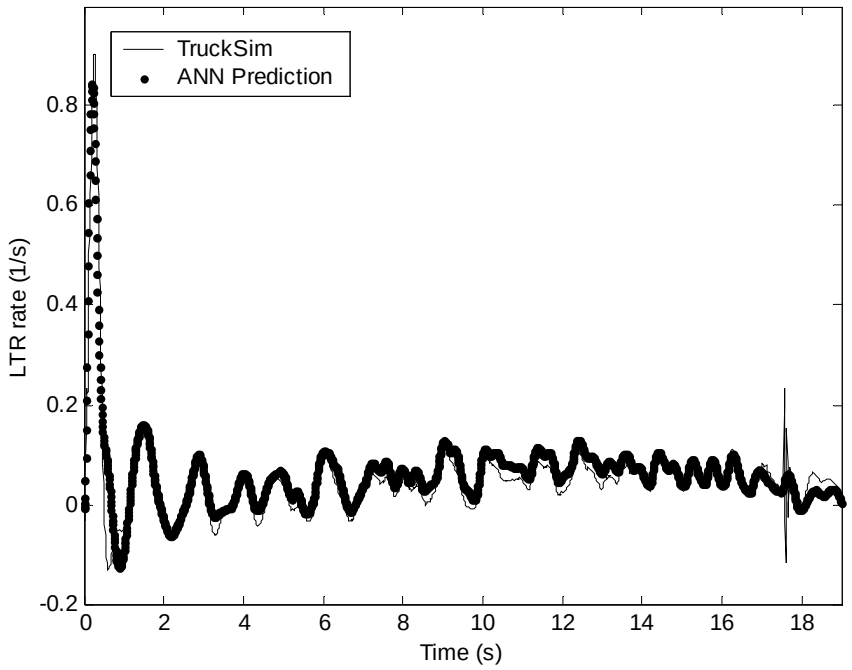


Figure 90: LTR rate prediction for the J-turn maneuver using the 2-2 A_y-YR-RR network

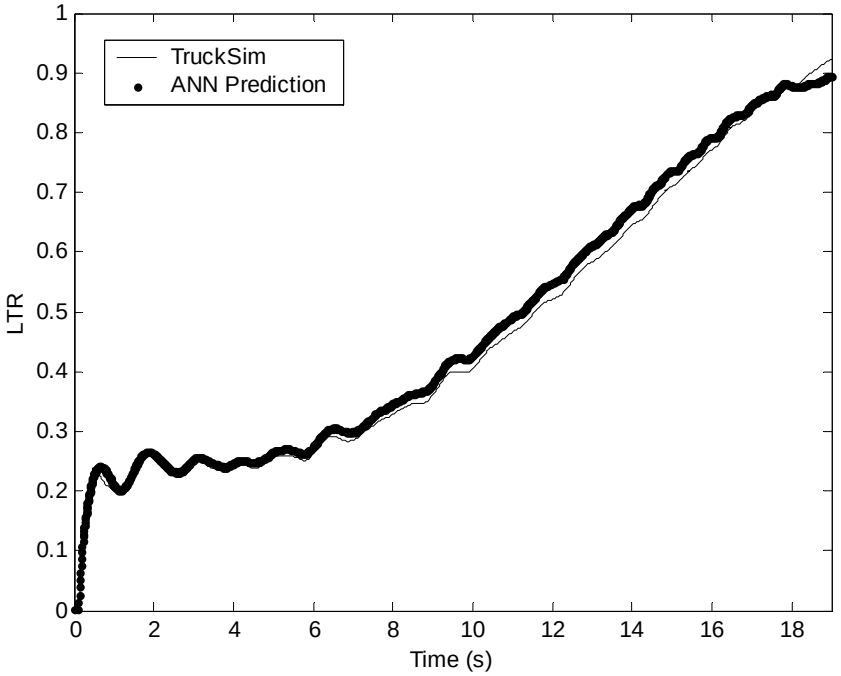


Figure 91: LTR prediction for the J-turn maneuver using the 2-2 A_y-RR-RA network

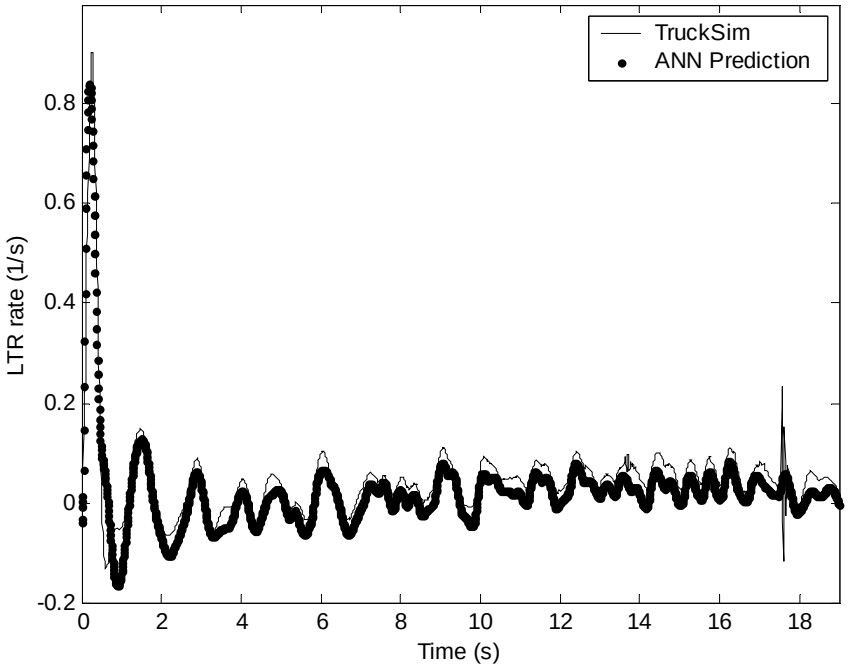


Figure 92: LTR rate prediction for the J-turn maneuver using the 2-2 A_y-RR-RA network

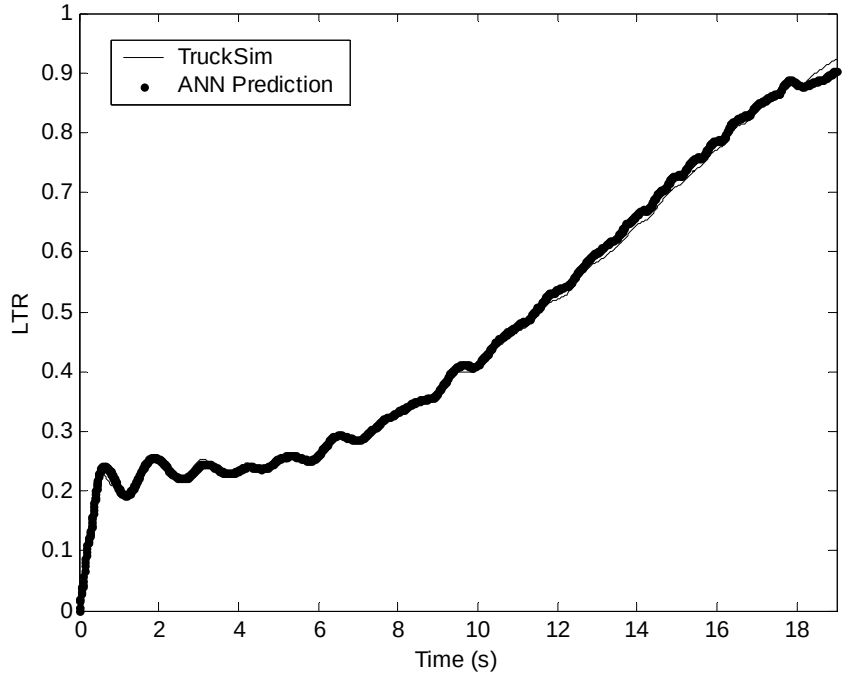


Figure 93: LTR prediction for the J-turn maneuver using the 2-2 A_y-YR-RR-RA network

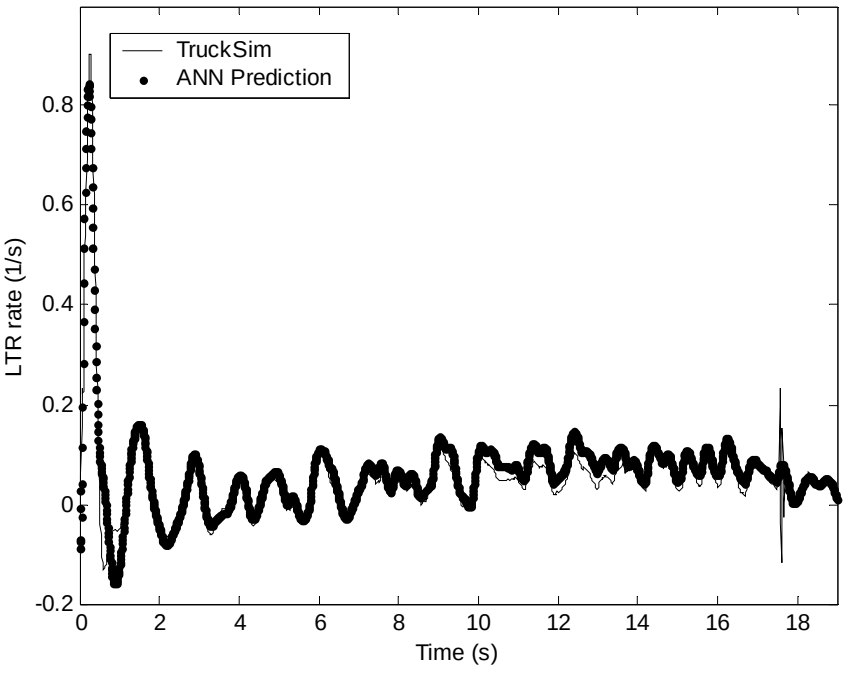


Figure 94: LTR rate prediction for the J-turn maneuver using the 2-2 A_y-YR-RR-RA network

In general, the networks performed surprisingly well using the experimentally acquired data to predict the LTR and LTR rate determined via TruckSim. The best performance overall was with the 2-2 architecture using the A_y-YR-RR-RA as inputs. The only downside with using this architecture is the fact that it contains the roll acceleration, which must be computed from the roll rate. As mentioned previously, this may be a problem when using the RWD in the following section since calculating a derivative will delay the LTR and LTR rate prediction. This delay may prove to be significantly damaging to the performance of the RWD during certain maneuvers.

VI.4.4 FLIS-RWD Results

The LTR and LTR rate predicted from the two more robust networks of the previous section (A_y-RR-RA & A_y-YR-RR-RA) will be used to test the RWD developed in Chapter V.

Figure 95 and Figure 96 displays the RWD output using the LTR and LTR rate predicted for the lane-change maneuver using the 2-2 A_y-RR-RA and A_y-YR-RR-RA networks, respectively. Furthermore, the actual warning level using the actual LTR and LTR rate for this maneuver is also plotted for comparison. Finally, the actual LTR is overlaid to aid in determining whether the warning would be useful, i.e., to help examine the performance of the RWD.

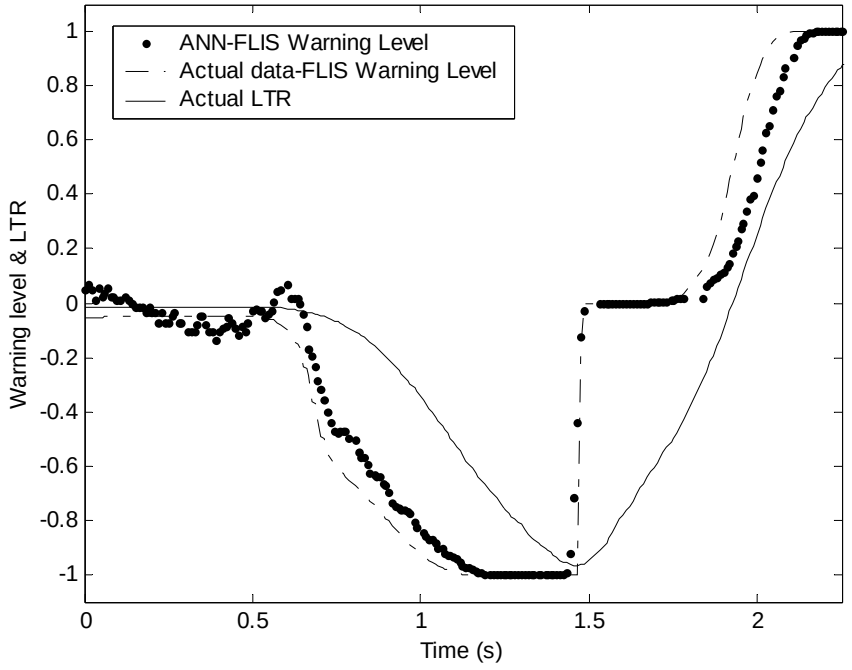


Figure 95: FLIS-RWD output using LTR and LTR rate data predicted from the 2-2 A_y-RR-RA network

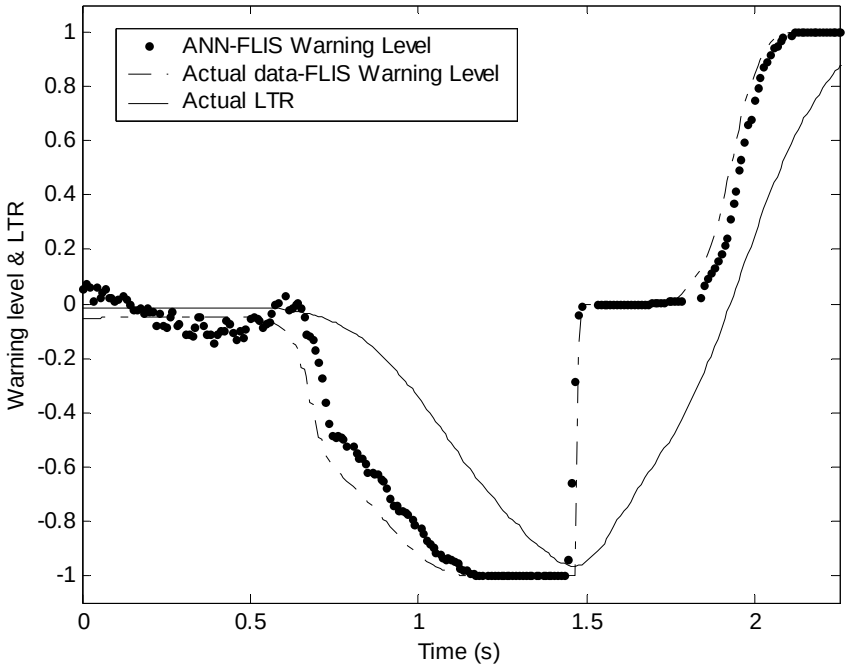


Figure 96: FLIS-RWD output using LTR and LTR rate data predicted from the 2-2 A_y-YR-RR-RA network

As expected, the timing and level of warning are almost identical since the LTR and LTR rates were predicted so closely. Furthermore, both warnings based upon the predicted data follow the actual warning level very closely. Examining the plots in more detail, it appears that the RWD begins to output a non-zero warning approximately 0.9 seconds prior to the LTR reaching its maximum. The maximum warning (-1) is output approximately 0.3 seconds before the LTR reaches it maximum. Though the RWD was not expected to prevent rollover due to maneuvers performed so quickly, the device was able to detect an warn the driver previous to the vehicle reaching its roll limit.

Figure 97 and Figure 98 displays the RWD output using the LTR and LTR rate predicted for the J-turn maneuver using the 2-2 A_y-RR-RA and A_y-YR-RR-RA networks, respectively. As expected again, the results are very similar due to the fact that the LTR and LTR rate predicted for this maneuver were almost identical. However, the network using the A_y-YR-RR-RA set of inputs performs slightly better at following the actual warning level as may be seen in Figure 98. As the vehicle enters the turn, the LTR jumps to 20%, though the output of the RWD is approximately 65% due to the non-zero LTR rate. The warning then stays constant and does the LTR, then starts to increase as the LTR increases. The warning level eventually reaches its maximum and stays there 8 seconds before the vehicle reaches rollover. The RWD is considered to be successful for this type of maneuver since the driver has plenty of time to take corrective action.

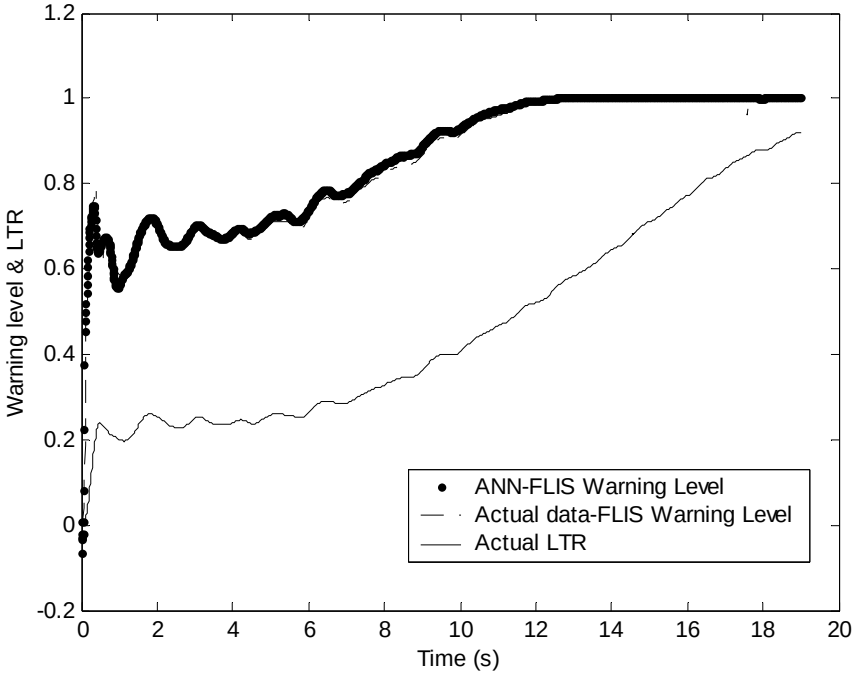


Figure 97: FLIS-RWD output using LTR and LTR rate data predicted from the 2-2 A_y-RR-RA network

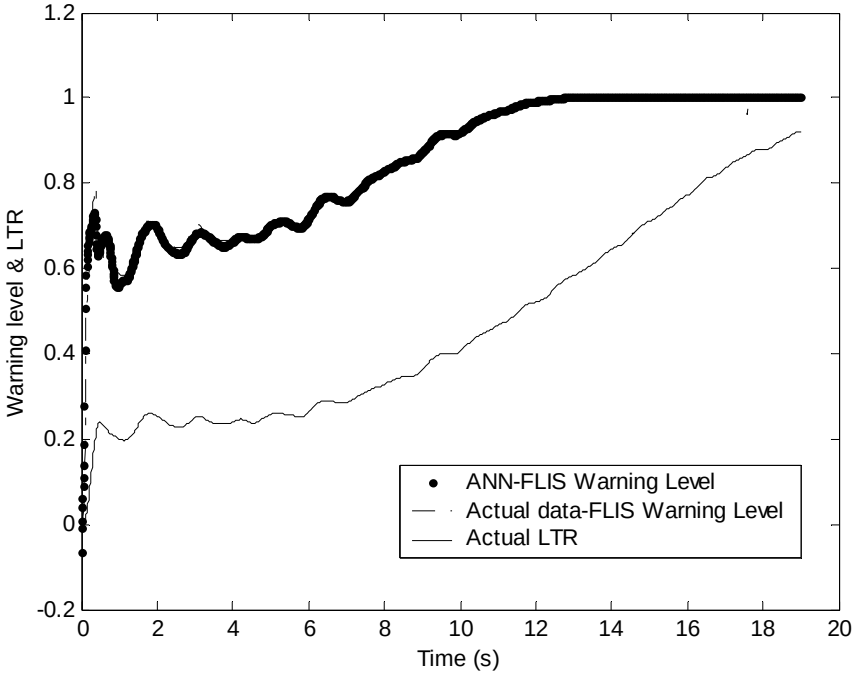


Figure 98: FLIS-RWD output using LTR and LTR rate data predicted from the 2-2 A_y-YR-RR-RA network

VII. Proposal for Hardware Implementation

It has come to the attention of the author that Mathworks, the developers of the Matlab numerical software package, has recently developed a software toolbox that allows Matlab to interface with hardware. This would eliminate the need for specialized ANN and Fuzzy Logic microcontrollers to be purchased, powered, programmed and implemented onto a custom built circuit board. Matlab's new Test & Measurement Suite allows the user to interface the Matlab code with the RS-232 serial port found on most PC's. Furthermore, the DMU used for the experimental data acquisition of this research interfaced via the RS-232 port. Therefore, the ANN and Fuzzy Logic code developed could simply appear 'in the loop' and in real-time as Matlab receives information from the DMU and processes it through the software based FLIS-RWD like that in Figure 99. The warning signal could be output via a gauge on the screen or output to a physical gauge to been seen by the driver. The only additional code that would need to be written would involve translating the digital output of the DMU.

Figure 99: Onboard FLID-RWD setup

As of this writing, the Test & Development Suite costs $3600 and requires Matlab to have already been purchased ($2000 for Commercial, $500 Educational) including the Neural Network and Fuzzy Logic toolboxes ($1000 for Commercial, $200 Educational).

The alternative to using the Matlab hardware interfacing option would involve the purchase of physical ANN and Fuzzy microcontrollers. These chips are very expensive, on the order of $3000 to $8000 each and require separate power supplies and custom-made cables to interface with the output hardware (computer or warning gauge). The ANN chips typically contain a minimum of 50 programmable neurons, which is far more than was required for this research. Furthermore, the support from the companies who designed and built these chips seems to be very lacking. Though the chips are still being sold, many of the companies who designed them have gone defunct.

A possible benefit of using the ANN and Fuzzy microcontrollers would be speed, which of course is vital to the FLIS-RWD performing successfully. Matlab requires a great deal of computational overhead that slows the execution of its routines and may be a factor with online implementation. However, compiling the Matlab code into an executable file can increase the processing speed. The compiled code could then be downloaded and stored in the EEPROM of a commercially available microcontroller that inputs the digital signal (containing input variables for the ANN code) from the DMU and outputs a signal

(warning level) to an external gauge. One such microcontroller is the BasicX-24 shown in Figure 100. The BasicX-24 can perform the required floating point math and the may interface with the DMU and a display gauge through its TTL input/output connections. Furthermore, the cost of the BasicX-24 is minute; less than $50 at the time of this writing. This would be the most ideal solution for on-board implementation.

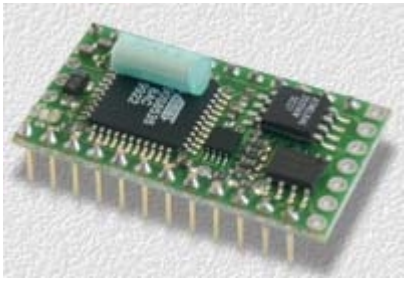


Figure 100: BasicX-24 Microcontroller

VIII. Summary

After an extensive literature review and study of previous work involving the roll stability of road vehicles, an adequate solution to the problems with rollover of heavy trucks and passenger vehicles has not been found. Rollover of heavy vehicles, such as tractor semi-trailers and straight heavy trucks, are particularly common due to their high center of gravity height to track-width ratio. Rollover of passenger cars is less common and furthermore is usually caused by being tripped against a curb or embankment. Tripped rollover is considered the most difficult type of rollover to prevent and this research did not address that problem. However, heavy trucks rollover frequently without being tripped and therefore a solution that has been suggested by previous researchers was the development of a passive rollover-warning device (RWD). The device would simply warn the driver of an impending rollover situation, leaving it up to the operator to take corrective action. Therefore, for a RWD device to be successful, the driver would require a warning in an adequate amount of time so as to take the corrective action. In other words, the RWD would probably only be successful for dangerous maneuvers that develop slowly, such as the curved on-ramp to a highway.

To this date and to the author's knowledge, a satisfactory RWD has not been developed. This is probably due to the fact that determining the vital motion cues that help predict roll stability is very difficult. Previous research suggested that measuring the load-transfer ratio (LTR) is the ideal indicator of instantaneous roll stability. However, determining the LTR requires measurement of dynamic wheel loads during the operation of the vehicle. In practice this is extremely difficult to do with any type of accuracy. Therefore, the research in this treatise focused on determining the LTR by other, more feasible means.

In this research, it was determined that Artificial Neural Networks, due to their powerful nonlinear mapping capabilities, had the potential to essentially 'learn' the dynamics of a road vehicle. Therefore, using as inputs to the ANN variables that are relatively easy to measure compared to dynamic wheel loads, a prediction of the wheel loads and LTR may be made.

A great number of computer simulations were performed to determine the feasibility of predicting the LTR using ANN's. Furthermore, a sensitivity (perturbation) analysis of a general nonlinear ANN with an arbitrary number of neurons was performed to determine the most likely candidates for inputs and their effect on predicting the LTR. The simulation results indicated that using measured lateral acceleration in various combinations with yaw and roll information, and a parameter used for steady-state cornering analysis called the Steering Factor, were quite successful in predicting the LTR.

Fuzzy Logic was used for the development of the warning device, taking as inputs the output from the ANN (LTR and LTR rate) to make a decision on the amount of warning that should be given. Fuzzy Logic was chosen for its ease of use and for the fact that it deals quite well with systems that are mathematically unclear.

Further work involved experimentally acquiring data with a vehicle during low and high-speed lane-change maneuvers. The data was used for updating and validating the simulation model used in this research and for the inputs to the ANN. The ANN then learned to predict the LTR and LTR rate, given by the simulation model, based on the experimentally acquired data.

The development of a software-based RWD was successful. Using the easily measured dynamic variables such as lateral acceleration, yaw rate, roll rate and roll acceleration, the ANN was able to predict the LTR and LTR rate to a very high degree of accuracy. As expected, the best results occurred for warning during maneuvers that developed slowly and showed that the RWD would have limited usefulness with high-speed maneuvers.

Finally, a proposed method for on-board hardware implementation of this RWD is presented.

Appendix A. Artificial Neural Networks

A.1 Introduction

Simply stated, an Artificial Neural Network (ANN) is an interconnected system of simple processing units (artificial neurons) all working in parallel to achieve some result. For most applications, an ANN may be considered a 'black-box' that receives an input and produces a desired output, i.e., it provides a mapping from an input space to an output space. ANNs have been successfully applied to a variety of applications including, but not limited to, pattern recognition, signal processing and filtering, control systems, robotics and function approximation. Although the roots of neural networks go back for almost a century [51], the majority of successful ANN implementations have occurred within the past two decades due to advances in digital computing. The field of ANNs is extremely vast and this short treatise cannot provide a full discourse. However, an introduction to some of the most common types of artificial neural systems will be discussed.

A.1.1 The Artificial & Biological Neuron

An artificial neuron is an extremely simplified model of a biological nerve cell such as that found in the human brain. Though our knowledge of how the brain functions as a whole is relatively limited, many of the most important biochemical reactions have been identified[52]. Figure 101 is a diagram of a biological nerve cell where the important features for this study have been labeled.

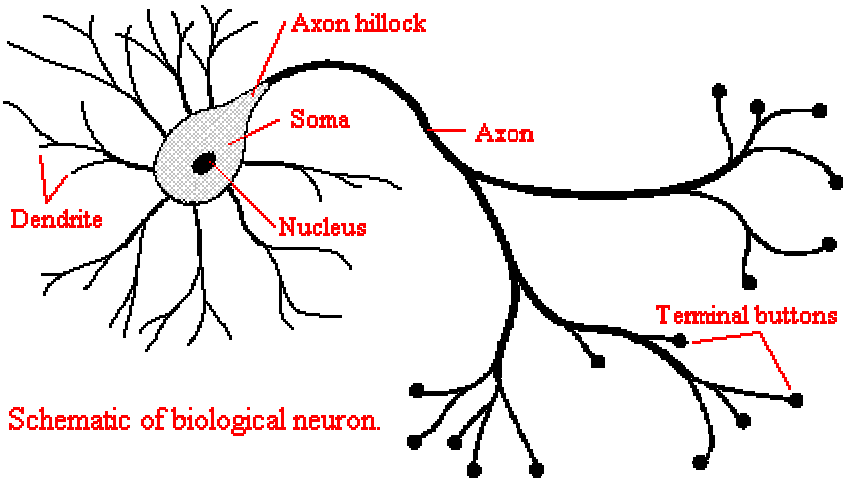


Figure 101: Schematic of a biological neuron

The soma is simply the cell body and is where the nucleus of the cell may be found. The role of the soma and nucleus is rather limited with regard to the processing of

information; they are mainly involved with cell maintenance to keep the cell alive and functioning properly. The dendrites are weighted input channels to the cell body that increase or decrease the individual signals passing through them. Once the weighted signals arrive at the cell body they are combined into a single value (a kind of summation) and the resulting signal is then passed along the surface of the cell body to the axon hillock, which acts as a function processor. Mathematically, the axon hillock is a simple nonlinear step-function that compares the magnitude of the incoming signal to an internal threshold value. If the signal is weaker than the threshold, nothing happens. However, if the signal is stronger than the threshold value, the axon hillock 'fires' and sends a separate constant-valued signal along the axon. It should be noted that this constant-valued signal is independent of the signal strength of the input to the axon hillock. Therefore, the output signal strength is the same regardless of whether the input was slightly above the threshold, or many times greater. Finally, the output signal is passed along the axon to the terminal buttons that act as the interfaces to the dendrites of other neurons through the synaptic gap (not shown). The input/output process then repeats itself for the many other neurons affected by the output signal of this single neuron.

Figure 102 is a diagram of a standard artificial neuron, based on the original model proposed by McCulloch and Pitts [53], that attempts to mathematically model the signal processing characteristics of the biological nerve cell. As shown in the diagram, the input channels have their own individual weights, w_i, that increase or decrease the incoming signal and are analogous to the dendritic arms of the biological neuron. After the input signals are weighted, they pass into a summing block that simply adds up all the individual weighted signals and produces a single output signal. This signal is then passed into an activation function (to be described later) for processing analogous to that performed by the axon hillock. Finally, the output of the activation function may be passed to other artificial neurons, including being passed back to itself.

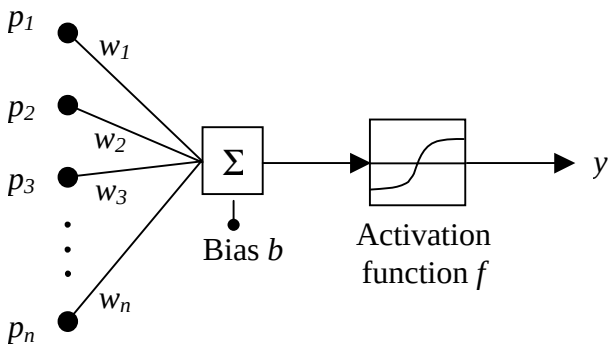


Figure 102: Artificial Neuron Model

Ironically, typical activation functions in artificial neurons are not limited to the relatively simple 'fire or no-fire' functions present in the biological neurons, and may be functionally more complex than their biological complement found in the axon hillock.

As will be shown, the ability to use different types of activation functions is one of the most powerful aspects of artificial neural computing.

A.1.2 The Artificial Neural Network Structure

Artificial neurons are typically implemented within individual layers where each neuron within a layer receives not only the same inputs, but receives these inputs at the same instant (for parallel processing). Furthermore, every neuron within an individual layer typically has the same activation function. One or more layers of neurons that are connected to one another through their respective weighted channels are considered a *network* of artificial neuron layers, or artificial neural network (ANN). Figure 3 is a diagram of a single-layered ANN that processes information in parallel from left to right in a three-step process. It should be noted that some ANN literature refer to the set inputs are as being a layer of its own and therefore Figure 103 would be referred to as a two-layer neural network; having an input layer and an output layer. However, that nomenclature will not be used in this treatise since technically the input 'layer' does not process any information; it is just the source of information *to be* processed. Furthermore, Figure 103 is an example of a *fully-interconnected feedforward* ANN, which simply means that every input is connected to every neuron in an individual layer (through the weighted channels) and that information is processed or 'fed' from left to right, which is typically considered 'forward'. Note that there are no connections between neurons in the same layer, but only between neurons of different layers.

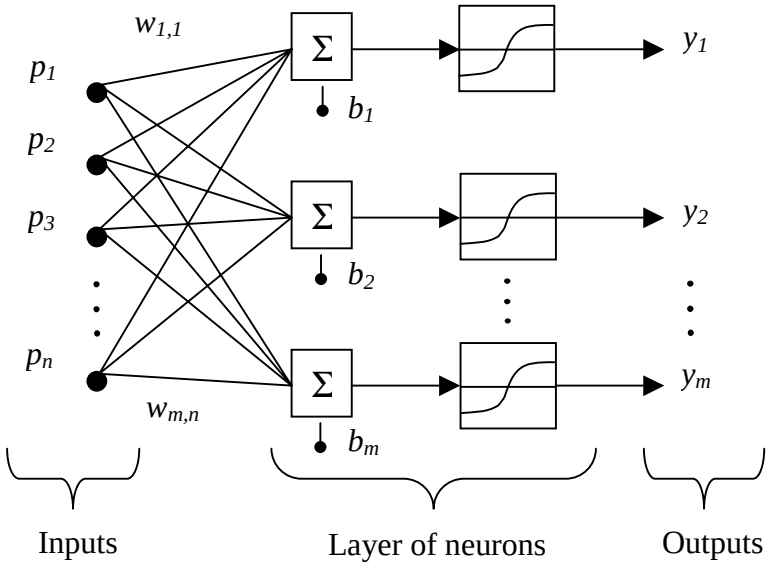


Figure 103: Diagram of a single-layer fully-interconnected feedforward ANN.

The first step in the computation process involves the scaling of the inputs p_i through the individual channels by the weights $w_{i,j}$. The individual weighted signals are then passed to every neuron in the layer (since it is fully-interconnected) where they are summed. For convergence properties [52], which will be explained in more detail later, an additional scalar value called a *bias*, b_i, is added to summation for each neuron. The result then enters an arbitrary function block (chosen by the network designer) and produces an output y_i. Note that there is a single output for each neuron in a layer, i.e., if there are *m* neurons in a layer, there will be *m* outputs of that layer. However, the number of inputs is independent of the number of neurons in a given layer.

For an ANN with *multiple* layers to be fully-interconnected, every output from each neuron within an individual layer is then input to every neuron of the next layer, and so on. Figure 104 is an example of a fully-interconnected feedforward ANN consisting of three layers with an arbitrary number of neurons, *m*, *r* and *z*, in layers one, two and three, respectively. A convenient notation used to describe an ANN like this is *m*-*r*-*z*, referring to the number of neurons in the individual layers. Some notes about Figure 104:

1. Layers one and two are considered *hidden* layers since they are concealed between the input 'layer' and output layer.
2. To simplify the diagram, arrows have been omitted on the connecting channels since it is assumed information is flowing from left to right, and weights on the individual channels are not shown.
3. The biases for each neuron are shown in the diagram with a superscript, indicating the layer the bias belongs to, and a subscript indicating the specific neuron the bias belongs to
4. The number of neurons in each layer is arbitrary, but is chosen by the designer and typically based upon experience.

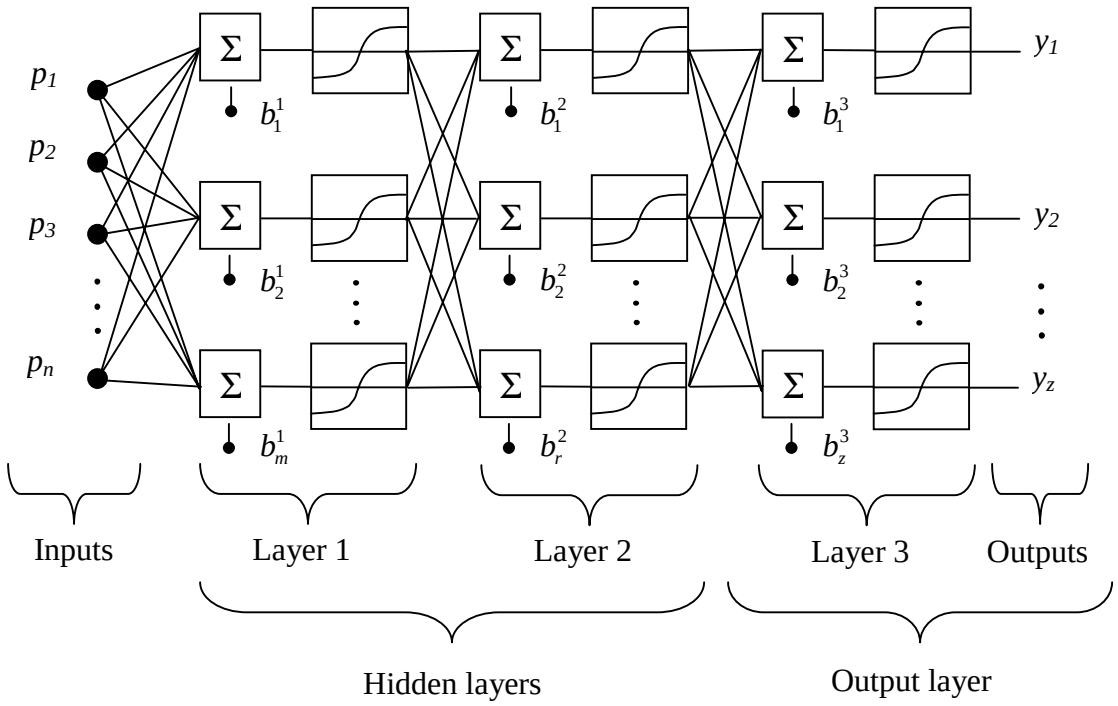


Figure 104: Diagram of a three-layer fully-interconnected feedforward ANN.

A.2 Mathematics of Neural Computing

The parallel processing, interconnected structure of an ANN lends itself well to be described with matrix algebra and for many applications the mathematics of neural computing is relatively simple. Bookkeeping tends to be the most challenging aspect in many applications of ANNs; however, computers are able handle the bookkeeping quite well.

For the single neuron in Figure 102 with multiple inputs and a single scalar output, the expression relating the input/output mapping may be shown to be

$$y = f(w_1 p_1 + w_2 p_2 + \ldots + w_n p_n + b)$$

where f is the activation functional operating on the sum of weighted inputs and a single scalar bias. In matrix form this relation may be expressed as

$$y = f(WP + b)$$

where

$$W = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix} \qquad P = \begin{bmatrix} p_1 & p_2 & \cdots & p_n \end{bmatrix}^T$$

The matrix algebra may be extended to the more general three-layer ANN in Figure 104 where each successive output from a layer acts as the input to the next layer. For *n* inputs and *z* outputs of the three-layer ANN in Figure 104, it is relatively simple to show that with *m*, *r* and *z* neurons in layers 1, 2 and 3, respectively, the output is

$$Y^3 = f^3\left(W^3 f^2\left(W^2 f^1\left(W^1 P + B^1\right) + B^2\right) + B^3\right) \tag{8}$$

where

$$W^1 = \begin{bmatrix} w^1_{1,1} & w^1_{1,2} & \cdots & w^1_{1,n} \\ w^1_{2,1} & w^1_{2,2} & \cdots & w^1_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w^1_{m,1} & w^1_{m,2} & \cdots & w^1_{m,n} \end{bmatrix} \quad W^2 = \begin{bmatrix} w^2_{1,1} & w^2_{1,2} & \cdots & w^2_{1,m} \\ w^2_{2,1} & w^2_{2,2} & \cdots & w^2_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ w^2_{r,1} & w^2_{r,2} & \cdots & w^2_{r,m} \end{bmatrix} \quad W^3 = \begin{bmatrix} w^3_{1,1} & w^3_{1,2} & \cdots & w^3_{1,r} \\ w^3_{2,1} & w^3_{2,2} & \cdots & w^3_{2,r} \\ \vdots & \vdots & \ddots & \vdots \\ w^3_{z,1} & w^3_{z,2} & \cdots & w^3_{z,r} \end{bmatrix}$$

$$B^1 = \begin{bmatrix} b^1_1 & b^1_2 & \cdots & b^1_m \end{bmatrix}^T \qquad P = \begin{bmatrix} p_1 & p_2 & \cdots & p_n \end{bmatrix}^T$$

$$B^2 = \begin{bmatrix} b^2_1 & b^2_2 & \cdots & b^2_r \end{bmatrix}^T \qquad Y^3 = \begin{bmatrix} y^3_1 & y^3_2 & \cdots & y^3_z \end{bmatrix}^T$$

$$B^3 = \begin{bmatrix} b^3_1 & b^3_2 & \cdots & b^3_z \end{bmatrix}^T$$

Note that the superscripts on the matrices denote the layer to which they belong and the subscript denotes the neuron that the individual weight or bias is associated with. Furthermore, the letter *P* has been reserved to denote the original input to the ANN where its subscript refers the *i*th input (i=1…n).

The output Y^3 in Equation 8 may be input to yet another layer, if so desired, making it a four-layer ANN with an output Y^4. As a matter of fact, there is no theoretical limit to the size of the network, nor is the ANN limited to having a two-dimensional structure (notice that the weight matrices are two-dimensional). Three-dimensional ANNs are commonly used in optical pattern recognition and self-organizing maps[52]. However, this proposal is meant to include an introduction to neural computing and will not discuss ANNs larger than three dimensions.

For completeness sake, an ANN with an arbitrary amount of layers, number of inputs and neurons within the layers would have to same nested structure as that in Equation 8, but

described more generally as

$$Y^k = f^k\left(W^k f^{k-1}\left(W^{k-1} f^{k-2}\left(W^{k-2} f^{k-3}\left(W^{k-3}(\cdots) + B^{k-3}\right) + B^{k-2}\right) + B^{k-1}\right) + B^k\right)$$

Output from layer *k-4* as a function of layers *k-5*, *k-6*,…,*P*

where *k* is the number of layers.

A.3 ANN Training

Training an ANN is a critical factor in designing a network to perform correctly and is really the key behind why ANNs are considered 'intelligent systems'. Thus far, no mention of how the weights and biases of an ANN are obtained to allow the correct mapping from a given input to a desired output has been provided. However, the adjustment of weights and biases to produce the satisfactory output for a given input is exactly what the training of a network involves.

Training an ANN is, in many ways, similar to the way humans are trained to catch a baseball, or the way a dog is trained to sit. The ability of the biological system within ours brains to adjust the weights and biases on nerve cells is why we are able to learn things.

There are essentially two types of training for ANNs: *supervised* and *unsupervised.* Supervised training requires information about what the correct output should be for a given input. A comparison of the input/output pair is made and some sort of algorithm makes adjustments as needed to the weights and biases. This type of training is analogous to the way a dog learns to sit on command. For example, when the command to sit is made (input), the dog's reaction (output) is examined by the owner (supervision). A typical method of teaching the dog to sit involves pushing the dog down in the sitting position after the command is made and giving the dog a reward, e.g. verbal praise or a snack. After a number of training cycles, the dog eventually learns to associate the word 'sit' with the reward from sitting down correctly. On the other hand, unsupervised training allows the network to 'learn on its own', so to speak, as there is no correct output to be associated with a given input. Unsupervised training is most commonly used for data clustering – input vectors learn to categorize themselves to specific regions within the output space. However, methods for unsupervised training of an ANN is beyond the scope of this research and with not be discussed further.

In general, humans learn how to perform many tasks by making mistakes. For example, a human infant does not inherently know how to walk at birth, but requires months of 'training' by trial and error. Using the feelings from its hands and feet, among other things, as inputs to neural net of the brain controlling motor skills, the child slowly learns the 'correct' feelings of standing upright. Biologically, the child's brain is continuously restructuring itself; from existing dendritic connections being strengthened or weakened, to entirely new connections being formed and others destroyed. This restructuring is all

in an effort to produce the 'correct' output of the neural net (motor skills controlling the child's muscles) for the given inputs to the neural net (sensory data from the child's appendages). It should be noted that the term *correct* was set in quotes because it is a very subjective term when applied to the precision of how humans learn new things. As will be shown, 'correct' is also very subjective when applied to ANNs for purely numerical tasks.

Many methods of training an ANN are available; however a description of them all would be beyond the scope of this treatise. We will examine a particular type of ANN utilizing a supervised method of learning called *error-backpropagation*, or simply, *backpropagation*.

A.3.1 Backpropogation

Backpropagation [54] is the most widely used ANN training algorithm in the world. This is probably due to the fact that the method is one of the most powerful as well as one of the easiest to understand and implement. It is powerful because any problem that requires associating an input pattern to an output pattern can be successfully solved assuming the network is set up correctly and that there exists some relation between the input pattern and the output pattern. It is easy to understand because the learning procedure is very intuitive since the foundations of neural computing were derived from the way humans learn. In short, backpropagation examines the output of the network, compares the output to a 'correct' example, calculates an error, and then this error is back-propagated into the network where the weights and biases are altered in a way that the network produces an output closer to the 'correct' value for the same input. In other words, the ANN *learns* from its mistakes, similar to how a child learns to walk as mentioned earlier. Note that backpropagation only refers to the way the weights and biases are updated, e.g., by propagating the error information back through the network, and is completely unassociated with the way information is processed through the network. Finally, the only real difference between backpropagating networks is the type of learning algorithm the network uses to update the weights and biases. Though there are numerous learning algorithms to choose from, only a few of the most common will be explained here.

In its simplest form, standard backpropagation refers to a type of steepest decent algorithm that calculates the maximum negative gradient on the error surface (defined by a performance function) relative to the weights, and then adjusts the weights in proportion to the error gradient. The movement along the steepest gradient of the error surface allows the performance function to decrease most rapidly. One of the most common performance functions used is the Mean-Squared-Error (MSE) relation. After each epoch, which is defined as a single pass of all the input data through the network, the error is computed, squared and then averaged. When the value of the MSE reaches an acceptable value, training is stopped and the network is considered 'trained'.

Equation 9 is an example of a basic backpropagation learning rule, where an individual weight $w_{i,j}$ at the $k+1$ epoch is updated in proportion to the gradient of error at the kth

$$w_{i,j}(k+1) = w_{i,j}(k) - \alpha \nabla e(k) \tag{9}$$

epoch . The constant of proportionality α is called the *learning rate* and is typically chosen by the network designer and falls between 0 and 1. In general, if the learning rate is large, the system learns faster but the learning procedure may become unstable where as if the learning rate is small, the gain in stability is paid for by a slower learning time. It should be noted that a priori knowledge of the optimal learning rate before training begins is not practical. As a matter of fact, the optimal learning rate for a specific training function changes during the training procedure [55]. Therefore, training procedures involving a *variable* learning rate have been developed and implemented with great success, such as that in [56]

An extension of the steepest decent technique that is frequently used includes information about how to weights are changing, typically referred to as 'momentum'. Momentum is included by adding a fraction of the weight change from the previous iteration to the current weight change determined by the standard gradient decent method. The inclusion of momentum allows the solution along the error surface to effectively escape local minimums and consequently allows faster convergence times. An excellent reference on the gradient decent method with momentum may be found in [58]

Although the steepest decent algorithm calculates the direction that the performance function decreases most rapidly, it does not always find the path for faster convergence. Therefore many types of algorithms that search for faster convergence times have been developed and are typically referred to as *conjugate* gradient algorithms. Some of these algorithms, such as Fletcher-Reeves [57] and Polak-Ribiere [58], perform a line search along conjugate directions of the steepest gradient. However, the line search methods are computationally expensive since the network response to all inputs must be computed several times for each search [55].

One of the most popular and powerful training algorithms used today, the Levenberg-Marquardt [59] algorithm (LMA), is based upon a basic Newton or Secant algorithm for multiple nonlinear equations. Newton algorithms require the computation of the Hessian matrix (second-partial derivatives) of the performance function $V(\underline{x})$ with respect to the weights and biases. Newton updates to the parameter vector $\underline{x}$ are of the form

$$\Delta \underline{x} = -\left[\nabla^2 V(\underline{x})\right]^{-1} \nabla V(\underline{x}) \tag{10}$$

where $\nabla^2 V(\underline{x})$ and $\nabla V(\underline{x})$ are the Hessian and Jacobian matrices, respectively. If the performance function is a squared-error type such as

$$V(\underline{x}) = \sum_{i=1}^{N} e_i^2(\underline{x})$$

it can be shown[60] that

$$\nabla V(\underline{x}) = J^T(\underline{x})\underline{e}(\underline{x})$$
$$\nabla^2 V(\underline{x}) = J^T(\underline{x})J(\underline{x}) + S(\underline{x})$$

where

$$S(\underline{x}) = \sum_{i=1}^{N} e_i(\underline{x})\nabla^2 e_i(\underline{x})$$

and is assumed to be nearly zero (Gauss-Newton Method). The LMA includes a variable factor μ, similar to a variable learning rate, within the brackets in Equation 10 that is adjusted depending on how $V(\underline{x})$ changes. Therefore, using $S(\underline{x}) \approx 0$ for approximation of the Hessian matrix, the parameter update (Equation 10) is now

$$\Delta\underline{x} = -\left[J^T(\underline{x})J(\underline{x}) + \mu I\right]^{-1} J^T(\underline{x})\underline{e}(\underline{x}).$$

The main strength of the LMA is that it is very efficient when compared to gradient algorithms and therefore the training time is relatively short. The downside of the LMA is the memory requirements are quite large and is really only very efficient for small to medium size networks since the memory requirements increase as the square of the size of the network [60].

A.3.2 Activation Functions

Using the correct type of activation function within each neuron of the network is critical in having the network perform correctly. Early networks, such as those based on the Perceptron [61] network architecture and learning rule, utilized the so-called *hard-limiter* function (see Figure 105a). The hard-limiter function acts on the sum of the weighted inputs, *net*, and produces a binary value of -1 or 1 (or 0 and +1). Research has shown that these types of networks can only perform linear classification, i.e., they can only classify linearly separable patterns (input vectors) and therefore their use was rather limited. Classic examples are the Perceptron's ability to represent the Boolean AND and OR functions, but not the Exclusive-OR (XOR) function [62].

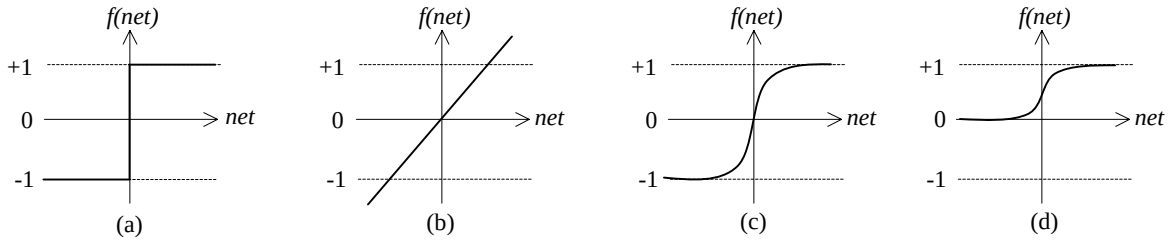


Figure 105: Common activation functions – (a) hard-limiter (b) linear (c) tan-sigmoid (d) log-sigmoid

Linear Networks

Networks based on the ADALINE [63] (ADAptive LINear Element) structure with a single layer of neurons containing *linear* activation functions (see Figure 105b) are typically referred to as 'linear adaptive filters' and are used in many practical applications such as digital signal processing, prediction in control, and noise cancellation. Their strength is that they can approximate any linear functionality between the inputs and outputs when used with a least-squared-error training algorithm such as the Widrow-Hoff Learning Rule [63] (WHLR). The WHLR for updating network weights and biases, shown in Equation 11 for a single neuron with *N* inputs, is a relatively simple version of the generalized backpropagation algorithm. Note that the WHLR as well as the backpropagation algorithm require a continuous activation function in order to calculate a derivative.

$$w_{1,j}(k+1) = w_{1,j}(k) + \alpha \frac{\partial\left(e^2(k)\right)}{\partial w_{1,j}} \qquad j = 1 \ldots N \tag{11}$$

Linear activation functions do not have much of an effect on the output of the network and only act to scale the output. Therefore, the partial derivative of the squared error in Equation 11 is relatively easy to compute (see [55]). For a single linear neuron with *N* inputs,

$$\frac{\partial\left(e^2(k)\right)}{\partial w_{1,j}} = 2e(k)\frac{\partial e(k)}{\partial w_{1,j}} \qquad j = 1 \ldots N$$

With the error defined as the difference between the target value and the output from the neuron,

$$\frac{\partial e(k)}{\partial w_{1,j}} = \frac{\partial\left[t(k) - o(k)\right]}{\partial w_{1,j}} = \frac{\partial}{\partial w_{1,j}}\left[t(k) - \left(\sum_{j=1}^{N} w_{1,j}\, p_j(k) + b\right)\right] = -p_j(k) \qquad j = 1 \ldots N$$

where $p_j(k)$ is the jth input at the kth epoch.

An analogous update law may be applied to the bias, *b*, and it is easy to show that

$$\frac{\partial\left(e^2(k)\right)}{\partial b} = 2e(k)\frac{\partial e(k)}{\partial b} = 2e(k)(-1)$$

These above results may be extended to multiple neurons using matrix form and the WHLR for updating weights and biases can be stated as Equations 12 and 13, respectively.

$$W(k+1) = W(k) + 2\alpha e(k)\,p^T(k) \tag{12}$$

$$b(k+1) = b(k) + 2\alpha e(k) \tag{13}$$

Example: Linear Function Approximation

As stated in the previous section, linear adaptive filters can approximate any linear function using a least-squared algorithm such as the WHLR. To demonstrate this fact, suppose a dynamic process that depends on a single input variable can be modeled exactly by the (unknown) linear relation

$$z(t) = 5x(t) + 4 \tag{14}$$

but all we have is the input and output data (*x(t)* and *z(t)*, respectively) at 0.5 second intervals. The mapping from the input to the output function will be approximated (actually it will be solved exactly since the process is exactly linear) using a linear adaptive filter and utilizing the WHLR. Arbitrarily choosing *x(t)* to range from 0 to 10, *z(t)* is found to range from 4 to 54. A plot of *z(t)* versus time is shown in Figure 106 at the 0.5 second intervals and the 1-input, 1-output ANN model for this problem is shown in Figure 107. The equation relating the input to the output for this model is simply,

$$y(i) = Wp(i) + b$$

where *W* and *b* are the weight and bias, respectively, that need to be found through training. Note that the increment *i* ($i = 1\ldots21$) corresponds to the time increment which was arbitrarily chosen to be 0.5 seconds and should not be confused with the epoch number. Recall that the epoch number corresponds to each pass of all the input data through the network and the consequent incremental update for the weight and bias.

Table 11 displays the results of training using the WHLR. Training was stopped when the results were accurate to the fourth decimal place corresponding to 2100 epochs. Since the weight and bias were initially unknown and chosen at random, it makes sense that the first epoch produces the largest error. However, the second epoch reduces the original error by more than 90% and is the single largest decrease in error per epoch. Furthermore, the results converge quite rapidly for the first three epochs, but then slow significantly with more and more training required to decrease the error by relatively

small amounts. This is typical of training using the WHLR. For completeness sake, the network was reset to the first weight and bias value chosen at random using the WHLR and the Levenberg-Marquardt algorithm was employed. The results displayed in Table 12 indicate that only three training epochs were required to approximate the function to within four decimal places. It is clear that the Levenberg-Marquardt training algorithm is much more efficient at training these types of networks.

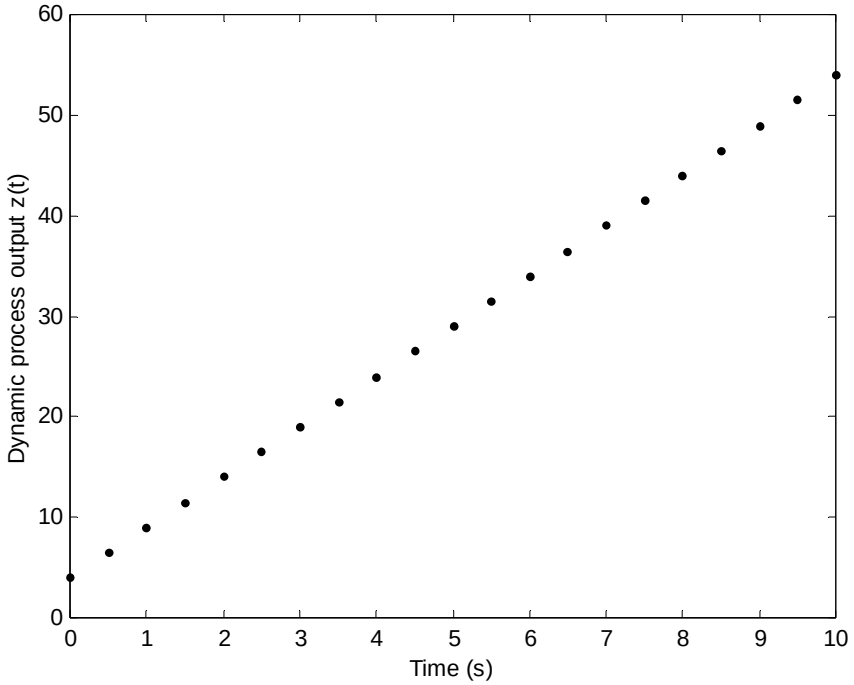


Figure 106: Plot of the linear time-dependent function z(t)=5x(t)+4 at 0.5 step values of x(t) ranging from 0 to 10.

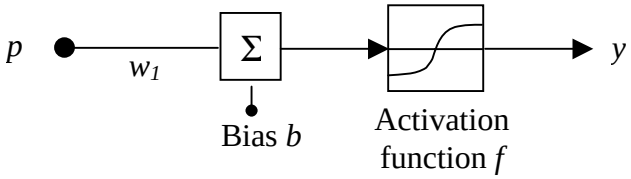


Figure 107: A 1-input, 1-output linear ANN model to map the function z(t)=5x(t)+4

Table 11: Function approximation results using the WHLR.

W	b	MSE	Epoch
0.6433	0.2898	8.239E+02	1
3.9914	0.7997	7.728E+01	2
5.0006	0.9646	9.195E+00	3
5.3037	1.0252	2.966E+00	4
5.3937	1.0543	2.376E+00	5
5.4222	1.1366	2.200E+00	10
5.4006	1.2835	1.980E+00	20
5.3421	1.6805	1.440E+00	50
5.2629	2.2175	8.526E-01	100
5.2020	2.6301	5.035E-01	150
5.0917	3.3783	1.037E-01	300
5.0320	3.7831	1.262E-02	500
5.0086	3.9419	9.067E-04	750
5.0023	3.9844	6.515E-05	1000
5.0002	3.9989	3.363E-07	1500
5.0000	3.9999	1.736E-09	2000
5.0000	4.0000	5.993E-10	2100

Table 12: Function approximation results using the Levenberg-Marquardt Algorithm.

W	b	MSE	Epoch
0.6433	0.2898	8.239E+02	1
5.0001	3.9995	8.103E-08	2
5.0000	4.0000	2.613E-15	3

A more practical use for this type of network would be to find the linear curve fit, in the least-squares sense, for sporadic data that appear to follow a linear relationship. Therefore, random noise has been added to the 'perfect' data obtained from Equation 14 and plotted in Figure 108. Again, the linear network of Figure 107 is initially assigned a random weight and bias, and then trained for 2000 epochs using the WHLR. The results using the WHLR as well as the Levenberg-Marquardt algorithm are presented in Table 13 and Table 14, respectively. Although the results with the WHLR indicate a slightly faster convergence time relative to the training with the WHLR on the 'perfect' linear data of Equation 14 (2000 epochs compared to 2100 epochs), the Levenberg-Marquardt algorithm is clearly the more efficient learning rule for these types of networks solving the problem in just two epochs. A plot of the ANN curve fit overlaying the noisy data is shown in Figure 109. A final note, the results for the weight and bias values match that exactly with the coefficients found using a standard least-squares curve fit.

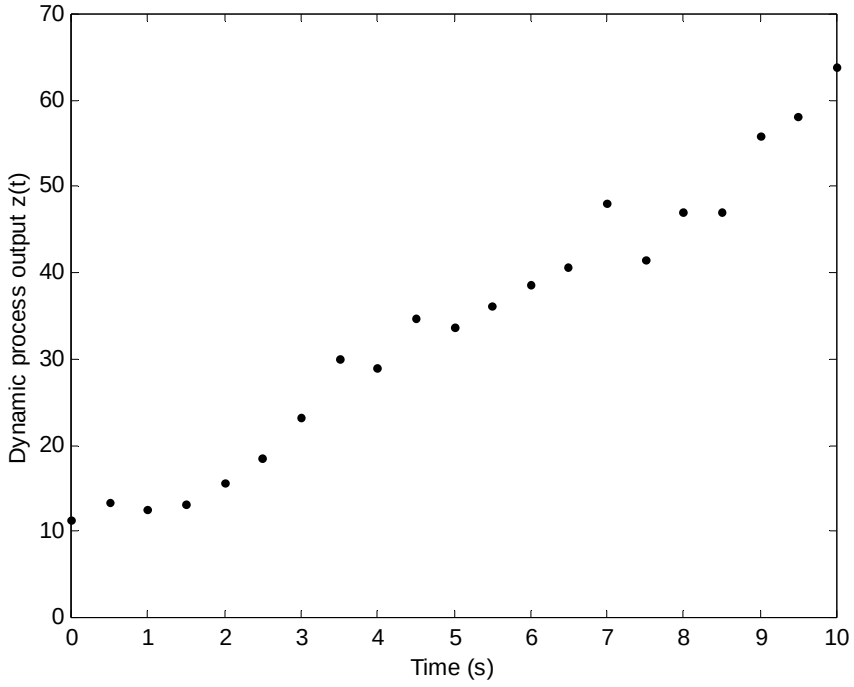


Figure 108: Plot of the linear time-dependent function z(t)=5x(t)+4 at 0.5 step values of x(t) ranging from 0 to 10 with added noise.

Table 13: Function with noisy data approximation results using the WHLR.

W	b	MSE	Epoch
0.5073	0.5877	1146.97	1
4.4254	1.2027	123.002	2
5.6047	1.4135	29.5415	3
5.957	1.5021	20.9027	4
6.0597	1.5538	19.9971	5
6.0804	1.7365	3.49946	10
6.0258	2.1075	18.0136	20
5.8917	3.0165	15.2008	50
5.7104	4.2461	12.1018	100
5.571	5.191	10.2716	150
5.3184	6.904	8.17532	300
5.1817	7.8311	7.69763	500
5.1088	8.3251	7.63147	1500
5.1085	8.3275	7.63146	2000

Table 14: Function with noisy data approximation results using the Levenberg-Marquardt Algorithm.

W	b	MSE	Epoch
0.5073	0.5877	1146.97	1
5.1085	8.3275	7.63146	2

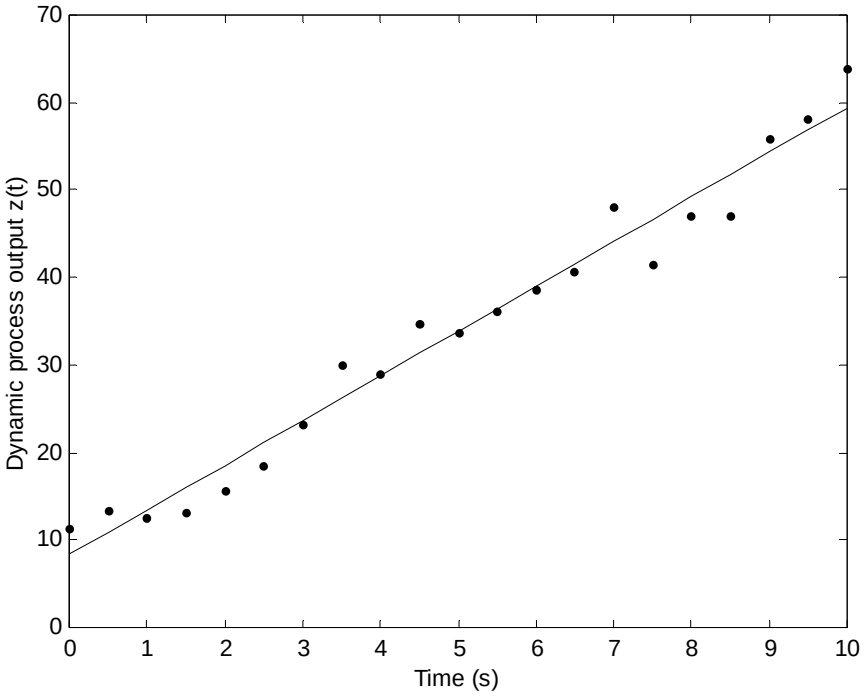


Figure 109: ANN representation of the function z(t)=5x(t)+4 with added noise

Nonlinear Networks

One of the most powerful aspects of artificial neural computing lies in the ease in which they perform *nonlinear* mappings from the input space to the output space. In addition, they have the ability to not only find nonlinear relations between the input and output variables, but many times can give a measure of the *strength* of the relationship between specific input and output variables. In other words, information about how important an input variable is to correctly model the output variables is contained in the weight and bias values and many times may be deduced. This feature may help reduce some unnecessary or redundant input data; thereby saving time and possibly money that would be spent collecting that data.

As with linear networks containing linear activation functions like that in Figure 105b, nonlinear networks must contain nonlinear activation functions such as the tan-sigmoid or log-sigmoid function shown in Figure 105c and Figure 105d, respectively. These

types of functions are commonly referred to as ‘squashing functions’, since they effectively ‘squash’ the output between specified limit points, albeit in a continuous fashion (unlike the hard-limiter function). The continuity of these nonlinear functions, shown mathematically in Equations 15 and 16, allow derivatives to be calculated and used with the various learning rules, such as WHLR and Levenberg-Marquardt. It should be clear that the only difference between the two functions is that the tan-sigmoid allows output values ranging from -1 to +1, while the log-sigmoid only allows output values between 0 and +1.

$$f(net) = \frac{2}{1+e^{-2net}} - 1 \qquad -1 \le f \le 1 \tag{15}$$

$$f(net) = \frac{1}{1+e^{-net}} \qquad 0 \le f \le 1 \tag{16}$$

The inherent capability of networks with nonlinear neurons is evident when the nonlinear functions are expanded into their Taylor series representations. Although the general Taylor representation is quite involved for activation functions like that in Equations 15 and 16, a single input to a log-sigmoid neuron with an arbitrary bias and unity weight can be expanded[64] as shown in Equation 17. It should be clear that this series can approximate any nonlinear function once the correct coefficients have been found (through training).

$$\frac{1}{1+e^{-(x+b)}} = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots \tag{17}$$

It should be noted that the importance of the scalar bias is evident in the Taylor series expansion of the log-sigmoid activation function, like in Equation 17, but without the bias value included in the exponent as shown in Equation 18. The series leaves out the even powers of *x* and therefore cannot approximate some nonlinear functions[64].

$$\frac{1}{1+e^{-x}} = a_0 + a_1 x + a_2 x^3 + a_3 x^5 + \cdots \tag{18}$$

Example: Nonlinear Function Approximation

Similar to the linear example in the preceding section, suppose we wish to map the nonlinear parabolic function

$$z(t) = 5x(t)^2 + 20 \tag{19}$$

which is plotted in Figure 110 at 0.5 second intervals. Using a single-input, single-output network with a single tan-sigmoid neuron on the hidden layer and a linear neuron at the

output, 10 epochs of training resulted in a MSE of 0.000026 utilizing the Levenberg-Marquardt training algorithm. The curve fit performed by the network is shown in Figure 111 with individual values at the 0.5-second intervals tabulated in Table 15 and their respective prediction error. The network clearly performs well, except at points near the origin. This can be improved by adding more nonlinear neurons in the hidden layer which has the mathematical effect of adding more higher-order terms in a series expansion, such as a the classic Fourier series, or increasing the order of the curve fit in a least-squares sense. For example, using two tan-sigmoid neurons in the hidden layer with a single linear neuron in the output layer, the MSE was reduced to 6.1e-8 after 100 epochs of training. These results are also tabulated in Table 15. It should be noted that although the network with two neurons in the hidden layer (Network 2) was allowed to perform 100 epochs of training compared to the 10 epochs performed by the network with only one (Network 1), this was not the cause of the more accurate representation on the function. This is due to fact that Network 1 reached its minimum gradient value (10^{-10}), typically set by the network designer, after just 10 epochs. In essence, this means that the network has learned the function to the best of its ability, i.e., the error gradient is so small that the weights and biases are virtually constant and cannot be trained any further.

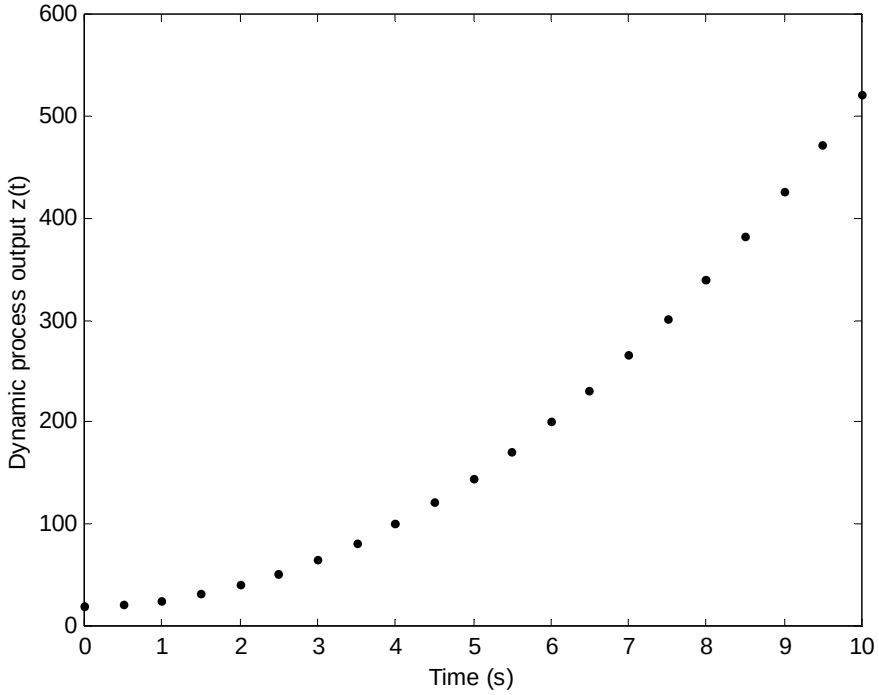


Figure 110: Plot of the nonlinear time-dependent function $z(t)=5x(t)^2+20$ at 0.5 step values of x(t) ranging from 0 to 10.

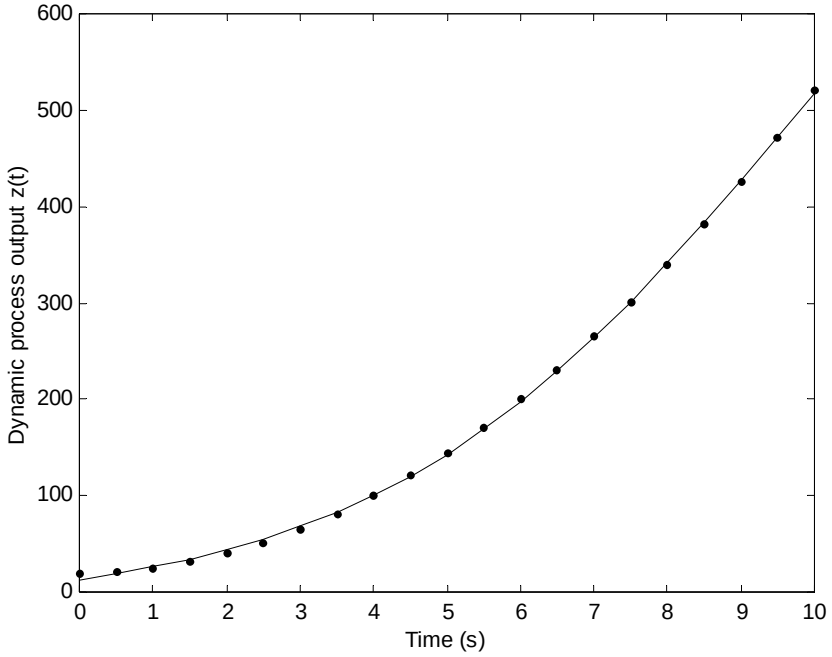


Figure 111: ANN representation of the nonlinear time-dependent function $z(t)=5x(t)^2+20$ at 0.5 step values of x(t) ranging from 0 to 10.

Table 15: ANN prediction of the nonlinear time-dependent function $z(t)=5x(t)^2+20$ at 0.5 step values of x(t) ranging from 0 to 10.

z(t) Actual	z(t) Predicted 1-NL neuron	%Error	z(t) Predicted 2-NL neurons	%Error
20.00	13.18	34.11	20.18	0.91
21.25	19.03	10.43	21.16	0.43
25.00	25.94	3.77	24.83	0.66
31.25	34.08	9.05	31.12	0.40
40.00	43.63	9.07	39.96	0.09
51.25	54.80	6.93	51.31	0.11
65.00	67.82	4.34	65.12	0.19
81.25	82.92	2.05	81.40	0.18
100.00	100.33	0.33	100.13	0.13
121.25	120.30	0.78	121.32	0.06
145.00	143.03	1.36	145.00	0.00
171.25	168.68	1.50	171.17	0.05
200.00	197.38	1.31	199.86	0.07
231.25	229.15	0.91	231.10	0.07
265.00	263.93	0.40	264.87	0.05
301.25	301.53	0.09	301.20	0.02
340.00	341.64	0.48	340.05	0.02
381.25	383.82	0.67	381.40	0.04
425.00	427.50	0.59	425.20	0.05
471.25	472.05	0.17	471.35	0.02
520.00	516.76	0.62	519.76	0.05

As with the linear example in the previous section, random noise can be added to the function in Equation 19 and the nonlinear networks may be used again as a function approximator. Results using Network 1 with the Levenberg-Marquardt algorithm and after 112 epochs of training may be found in Figure 112. The accuracy of the results are similar to those when Network 1 was trained on the 'perfect' function, where values near the origin may be slightly off. Therefore, Network 2 was trained to approximate this function and the results after 67 epochs of training are shown in Figure 113. Clearly this is not the shape we desired because we knew a priori that the function should be parabolic. This is an example of a common problem when using neural networks for function approximation referred to as *over-training*.

Over-training

Over-training can occur for a number of reasons, but most frequently when too many training epochs are performed to fit the data and/or having too many neurons in the hidden layer. As mentioned earlier, the number of neurons in the hidden layer is proportional to the freedom the network has to fit the data. Therefore, the network has the ability to over-*fit* the data and may miss the 'bigger picture' of what the shape is supposed to look like. This did not occur with the 'perfect' data curve-fitting results tabulated in Table 15 because the training data *was* perfect, i.e., the training data lied on a perfect parabolic shape and therefore made it impossible for the network to move outside the data. Over-fitting a batch of data can be exemplified using a network with 10 tan-sigmoid neurons in the hidden layer and the results after 300 epochs of training are plotted in Figure 114. The network now has the ability to conform its way through all the data points and clearly, this is not the fit we desired. Recall, however, that we knew a priori what the shape should look like, which of course is not always the case. For example, the process $z(t)$ may actually follow the path shown in Figure 113, depending on how much we know about the process we are trying to model.

Finally, the number of training epochs that the network is allowed to perform has a substantial effect on the results. Since the type of training we are performing uses the MSE as the performance function and requires that all the data pass through the network before any updates are made the weights and biases (defined as one epoch), the initial training epochs result in a more general shape for the curve fit. For example, Figure 115 shows the results of training a network with 10 neurons in the hidden layer on the noisy data of Figure 112 after just 4 epochs. Obviously, this is a better representation of the overall shape of the data. This technique is commonly referred to as 'early-stopping' [55].

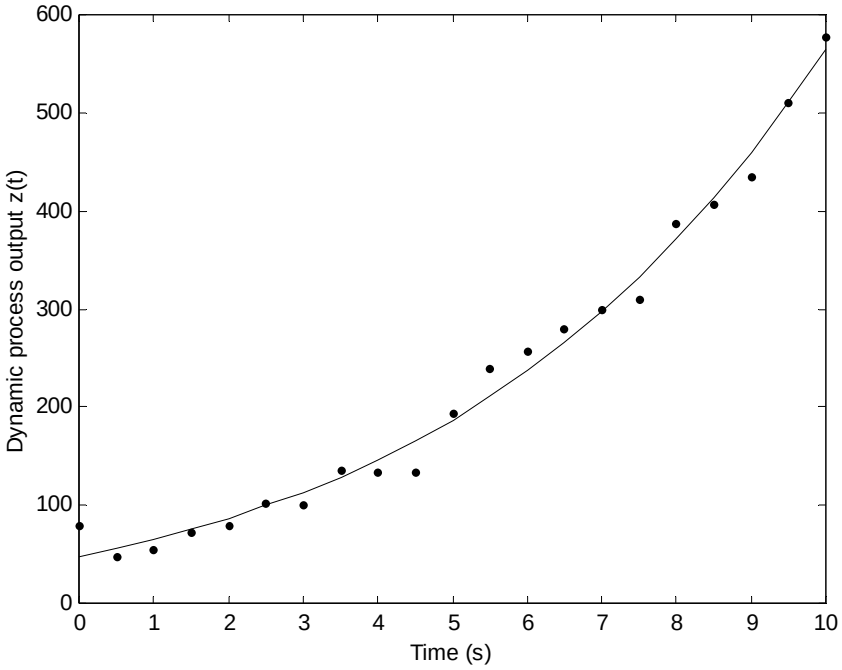


Figure 112: ANN representation of the nonlinear time-dependent function $z(t)=5x(t)^2+20$ with added noise at 0.5 step values of x(t) ranging from 0 to 10.

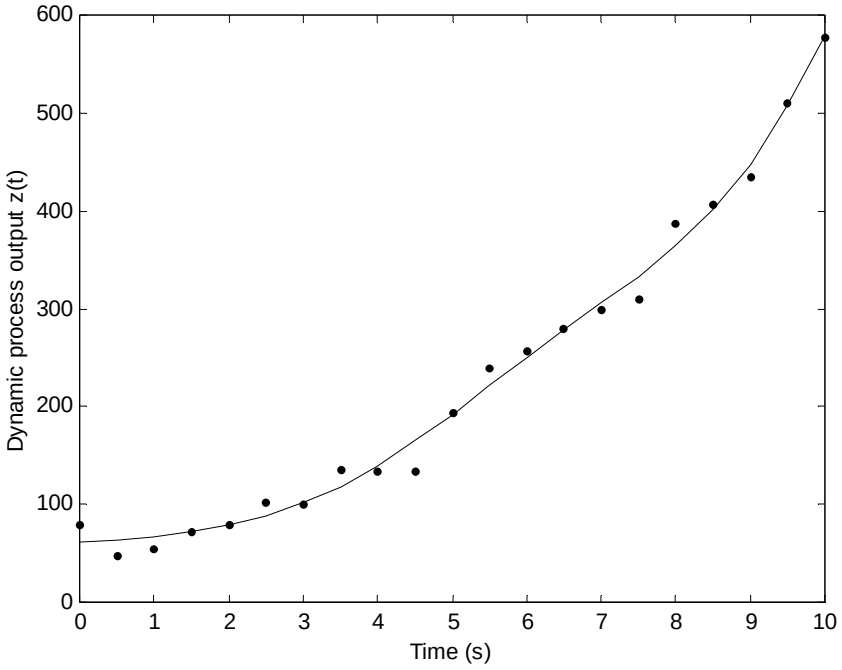


Figure 113: Example of using too many neurons in the hidden layer

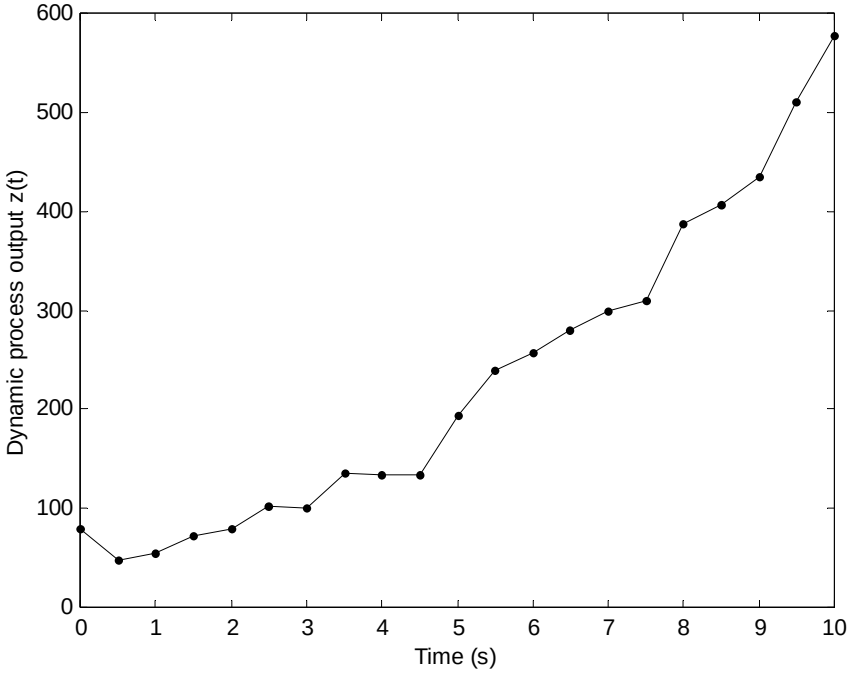


Figure 114: Example of the over-fitting of data performed by an over-trained network.

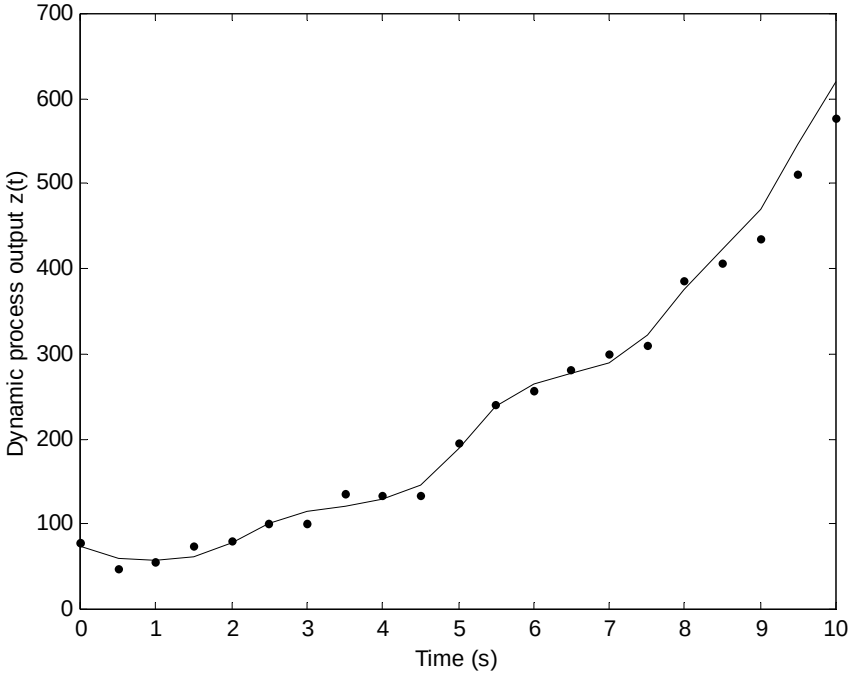


Figure 115: Example of the ‘early-stopping’ technique used to training ANNs using a network with 10 tan-sigmoid neurons in the hidden layer.

Appendix B. Fuzzy Inference Systems

B.1 Introduction

As with an artificial neural network (ANN), a Fuzzy Logic Inference System (FLIS), or simply 'Fuzzy System', is a tool to map an input space to an output space. However, since ANNs are based purely on well-defined mathematical tools, such as least-squares curve fitting or function minimization, this leaves the user with little knowledge of the internal workings of the network, i.e., a trained ANN that maps an input space to an output space is truly a 'black-box'. But the development of a FLIS actually relies on the day-to-day experiences of the designer, or the designer's specific experience with a system they are trying to model. Furthermore, once the FLIS is developed into the black-box that satisfactorily maps the inputs to the outputs, the box is actually quite transparent in that the inner workings are relatively easy to understand. Finally, FLIS's are typically a more convenient solution for many types of problems. As a matter of fact, it is typically regarded that if the development of a FLIS to model a particular system is *not* convenient, then there is probably a better solution using another method.

Probably the most powerful aspect of Fuzzy Systems is that they are developed using *natural* language (linguistics) that ordinary people use on a regular basis[65]. Furthermore, though the foundations of Fuzzy Logic are qualitative in nature, these qualitative descriptions are converted into precise mathematical formulations to give crisp output values. This is actually where the contradiction in terms 'Fuzzy Logic' is generated; *Fuzzy*, implying non-crisp or non-exact, and *Logic*, implying exact, crisp values such as 1 or 0.

A simple example of type of 'fuzzy' reasoning humans use everyday is whether or not to take an umbrella with you when you go outside. Some of the inputs to our reasoning 'system' may be the condition of the sky, whether sunny, cloudy or a mixture of the two and the status of the forecast for the day, whether the chance of rain (COR) is low, medium or high. We can systematically develop some rules for a *Rule Base* that apply to this decision process. For example, assuming it is not raining at the time of our decision, some rules we could develop could be:

1. If the sky is cloudy and COR is high, then take the umbrella
2. If the sky is cloudy and COR is medium, then take the umbrella
3. If the sky is cloudy and COR is low, then do not take the umbrella
4. If the sky is sunny with clouds and COR is high, then take the umbrella
5. If the sky is sunny with clouds and COR is medium, then do not take the umbrella
6. If the sky is sunny with clouds and COR is low, then do not take the umbrella
7. If the sky is sunny and COR is high, then take the umbrella
8. If the sky is sunny and COR is medium, then do not take the umbrella
9. If the sky is sunny and COR is low, then do not take the umbrella

Notice that there is always an *if*-part (antecedent) and a *then*-part (consequent) of each rule. This is a very important part of developing a FLIS as the analysis of the antecedent and consequent compose three of the five steps in Fuzzy Logic computation. The fuzzy-ness of our Fuzzy System will appear when we try to define what exactly 'cloudy' is, or what 'low' is. As we will see, defining these kinds of subjective terms is relatively easy to do using Fuzzy Logic.

B.2 Foundations of Fuzzy Logic

B.2.1 Fuzzy Sets

Fuzzy sets are sets without clear boundaries, i.e., the boundaries do not clearly separate possible members. For example[65], Monday, Thursday and Saturday can be considered members of the classical set 'Days of the Week' as shown in Figure 116a. There is no question that these elements fall crisply and clearly into that category, as opposed to the items shoes, cars and money. However, Figure 116b shows a fuzzy set 'Days of the Weekend'. Though Saturday and Sunday clearly fall into the category with 100% membership, it is not clear whether Friday is considered part of the weekend, or part of the week. If a poll were taken, people would probably disagree as to whether Friday was a weekday or a part of the weekend. Furthermore, they would disagree as to the amount that Friday belongs to either category. Some people might feel that Friday is 70% a day of the week and 30% a day of the weekend. Therefore, we can give Friday a *degree of membership* to the sets 'Days of the Week' and 'Days of the Weekend', of 0.7 and 0.3, respectively. The idea of a degree of membership lays the foundation of Fuzzy Logic.

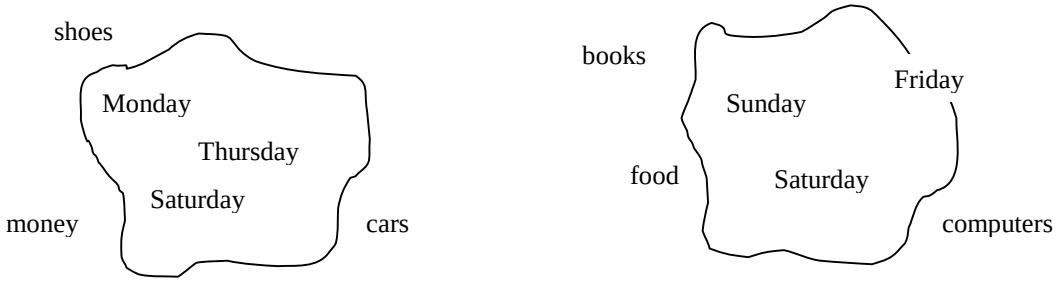


Figure 116: (a) Classical set with clearly defined boundaries (b) Fuzzy set with non-crisp boundaries allowing partial membership

B.2.2 Membership Functions

The degree of membership in Fuzzy Logic is described by a membership function (MF), which is itself defined by the FLIS developer. Fuzzy Logic may be considered a extension or even *superset* of Boolean Logic, where the membership functions output

either 0 or 1. This extension allows partial membership with output values *between* 0 and 1. When Boolean Logic is applied to our Day of the Week or Day of the Weekend controversy, Friday is strictly a Day of the Week as is shown graphically in Figure 117a. However, Figure 117b shows a continuous MF that represents our 'feeling' about Friday and allows it to be partly a Day of the Week and partly a Day of the Weekend. Notice that as Friday starts off (morning and early afternoon), the degree of membership is relatively low since most people do not feel 9:00 AM is really part of the weekend. However, as the afternoon hours and early evening come about, the degree of membership rises exponentially reaching about 70% at dinner-time. Only a few of hours later, membership reaches in excess of 90%. This is in contrast to the strict Boolean example of Figure 117a where Friday at 11:59 PM has a membership value of 0 for Day of the Weekend.

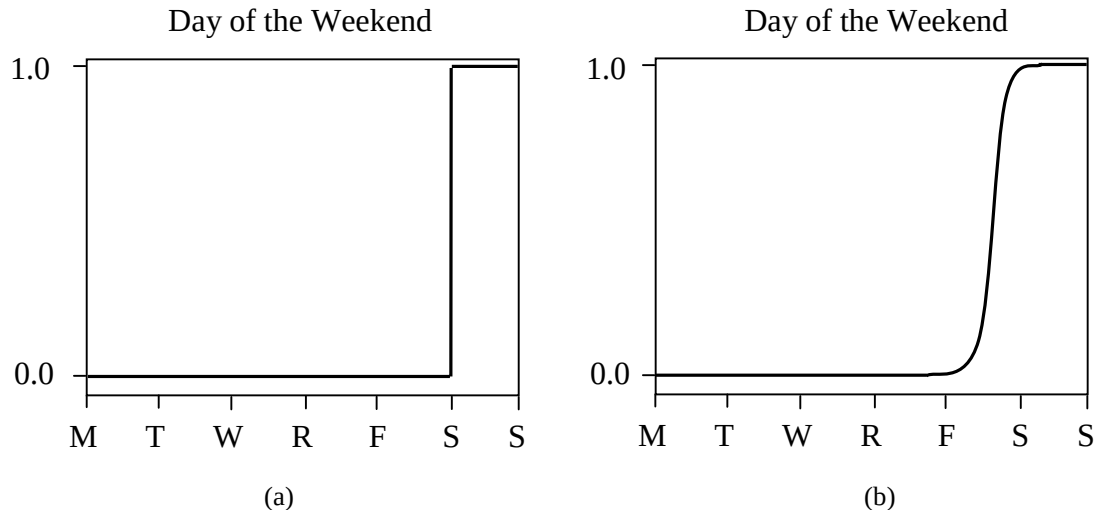


Figure 117: Membership functions for the input Friday to the Day of the Weekend category represented by (a) Boolean Logic and (b) Fuzzy Logic

The actual shape of an MF is up the FLIS designer and is typically based upon their experience with the system they are trying to model. Figure 117b is an example of a sigmoidal MF that is used for many types of applications. Other commonly used MFs are shown in Figure 118.

To show how Fuzzy Logic is a superset, and therefore may represent, the standard binary Boolean Logic operators AND, OR and NOT (the so-called truth-tables, shown in Table 16), the fuzzy equivalents using the *min*, *max* and *complement* functions are shown in Table 17. Note that the fuzzy equivalents do not have to be binary values (0 or 1) since the MFs allow partial membership. Therefore, the fuzzy operators AND, OR and NOT may be considered *multi-valued* logical operators. This can be shown graphically in Figure 119 where the multi-valued logic is shown in contrast to the binary values of Boolean Logic.

B.3 Fuzzy Inference

Fuzzy inference refers to the actual mapping from the input space to the output space using a FLIS. There are two methodologies in frequent use today for fuzzy inference: the Mamdani Method [66] and the Sugeno Method [67]. The Mamdani Method allows MFs like those used for Fuzzification to be used for Implication (see Figure 118) and is by far the most frequently used of the two methods. This is due to the fact that using the fuzzy MFs, like those in Figure 118, is very intuitive and well suited for human decision-making. The Sugeno Method is a little more abstract in that it allows an explicit mathematical function of the inputs; specifically either a linear (first-order) function or a constant (zero-order) function.

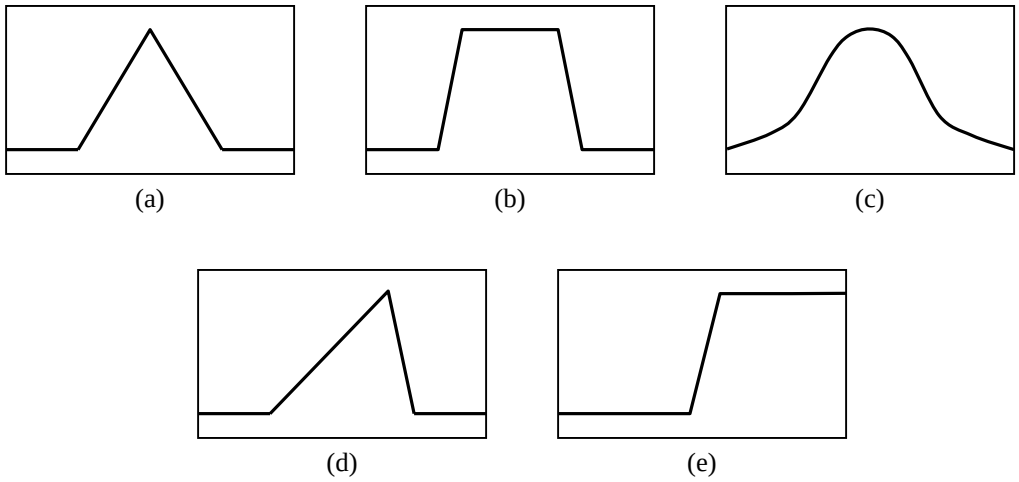


Figure 118: Common membership functions: (a) Triangular (b) Trapezoidal (c) Gaussian (d) Skewed Triangular (e) Open-ended Trapezoidal

Table 16: Boolean Logic truth-tables (a) AND (b) OR (c) NOT

a	**b**	**a AND b**
0	0	0
0	1	0
1	0	0
1	1	1

(a)

a	**b**	**a OR b**
0	0	0
0	1	1
1	0	1
1	1	1

(b)

a	**NOT a**
0	1
1	0

(c)

Table 17: Fuzzy Logic equivalent truth-tables (a) AND (b) OR (c) NOT

a	b	min(a,b)
0	0	0
0	1	0
1	0	0
1	1	1

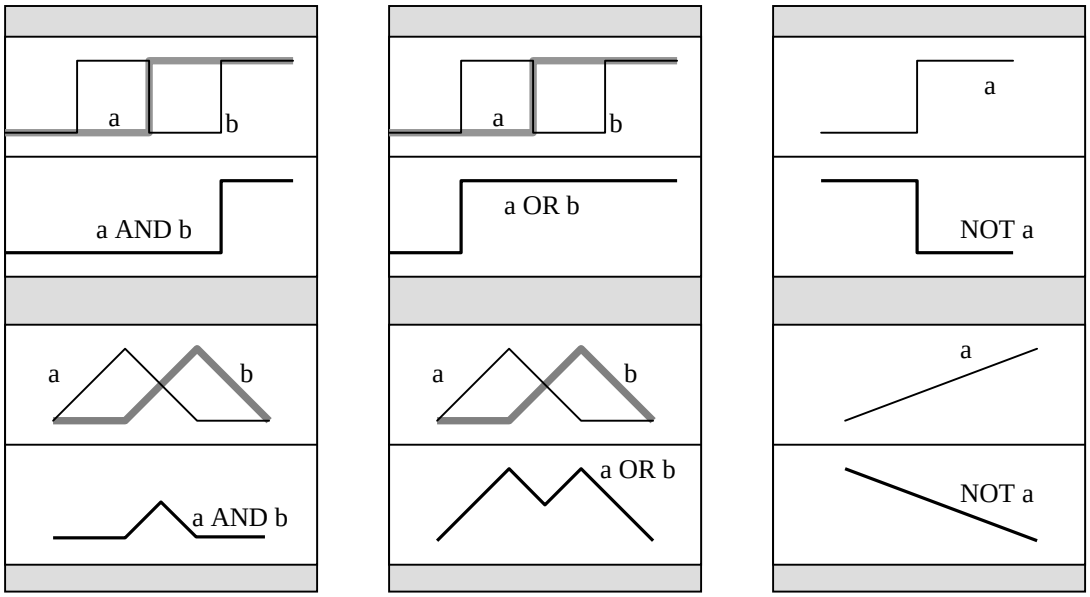
(a)

a	b	max(a,b)
0	0	0
0	1	1
1	0	1
1	1	1

(b)

a	1-a
0	1
1	0

(c)



(a) (b) (c)

Figure 119: Comparison of binary-valued (Boolean) logic to multi-valued (Fuzzy) logic for the logical (a) AND (b) OR (c) NOT

As mentioned previously, there a five steps that any FLIS must accomplish to obtain a result. The author believes the addition of another step, which refers to assigning the input variables actual values within a user-defined range should be included and will (hopefully) help clarify the discussion that follows. However, since this additional step must occur before the standard five steps and since some may feel it is obvious or too elementary, the additional step will be referred to as Step 0. The six steps are as follows:

0. Assign input values to the input variables within the user-defined range
1. Fuzzification of the inputs
2. Apply the fuzzy operator(s)
3. Apply the implication method
4. Aggregate all the rule-based outputs
5. Defuzzification of the aggregate from Step 4

Step 0

Step 0 refers to actually assigning numeric values within a user-defined range to our linguistic input variables. Although Step 0 may be obvious for some applications, such as using the exhaust gas temperature from a jet engine, which would have a predefined range, the range for many applications are completely arbitrary. For example, what should the numeric range be for the input linguistic variable *sky* from the five rules developed in the Umbrella FLIS? When the input range is arbitrary, using 0 to 10 is a typically a good place to start. Therefore, if 0 corresponds to the worst possible *sky* value (hurricane) and 10 corresponds to the best possible *sky* value (sunshine with no clouds), an input value of *sky*=8 would refer to a pretty nice day (probably sunny with some scattered clouds). To summarize, Step 0 refers to defining the mathematical range that the input variables must lie within (e.g., 0 to 10) and also assigning specific values to these linguistic inputs (e.g., *sky*=8).

Step 1

Step 1 refers to the process in which our linguistic inputs are assigned a 'fuzzified' numerical value depending on the shapes of the various MFs we have defined for the respective sets (categories). For example, if there are three sets or categories that all possible values of *sky* can be a partial member of, say 'Cloudy', 'Sunny with Clouds' and 'Sunny', we can *qualitatively* assert that *sky*=8 would partially belong to categories 'Sunny with Clouds' and 'Sunny' (and possibly 'Cloudy'). The qualitative nature of a FLIS is important because that is exactly how it are designed from the start. This stems from the fact that humans think more qualitatively than quantitatively and a FLIS is designed to reflect that. Therefore we *want* the value *sky*=8 to have a partial membership in 'Sunny with Clouds' and 'Sunny' because it makes sense. Recall Figure 117b where we determined Friday should be a partial member of the category 'Days of the Weekend'. It makes sense that Friday at 10:00 PM be 90% a Day of the Weekend. Therefore, based upon the desire to have *sky*=8 be a partial member of 'Sunny with Clouds' and 'Sunny',

we would design our MFs to reflect that fact. The *quantitative* degree of membership with be determined by the position and shape of our MFs that mathematically describe the three categories.

For this example, there are three possible categories each described by an individual MF and each producing a 'fuzzy' output value. We have chosen to use an Open-ended Trapezoidal MF for categories 'Cloudy' and 'Sunny' and a Skewed Triangular MF for the category 'Sunny with Clouds' which are shown overlaid onto one another in Figure 120. The Open-ended Trapezoidal shape for 'Cloudy' was adjusted to create a cut-off effect for the lower values of *sky*. Although this will be made clearer in the description of steps that follow, the Open-ended Trapezoidal effectively gives more 'power' to the argument one should take their umbrella with them when the sky looks bad (e.g., *sky*=0, 1 or 2) regardless of the *forecast* value. A similar reasoning is behind the Skewed Triangular MF for the category 'Sunny with Clouds'.

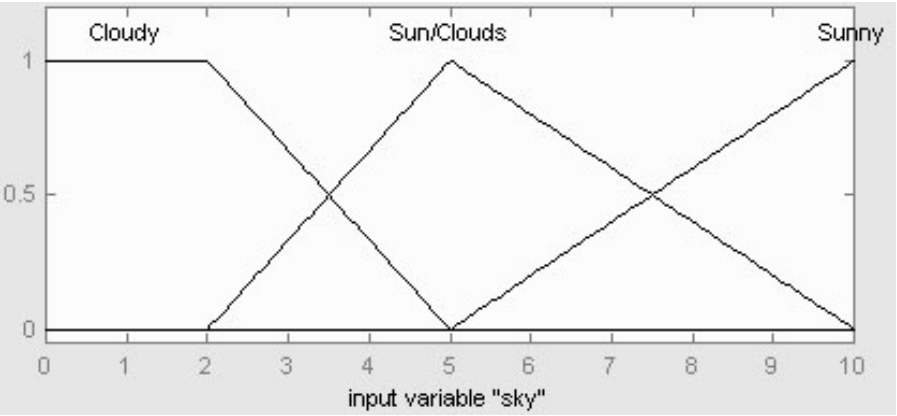


Figure 120: Overlaid independent MFs describing the input variable *sky*

Applying our input *sky*=8 to the three MFs we defined, results in the fuzzified values shown in Figure 121. The three fuzzified outputs, 0.6, 0.4, and 0.0 correspond to the degree of membership in the categories 'Cloudy', 'Sunny with Clouds' and 'Sunny', respectively. In other words, our system tells us that *sky*=8 refers to 60% 'Sunny', 40% 'Sunny with Clouds' and is not 'Cloudy' to any degree. Note that the outputs do not *have* to add to 100%, since the MFs are completely independent, though in this case they do.

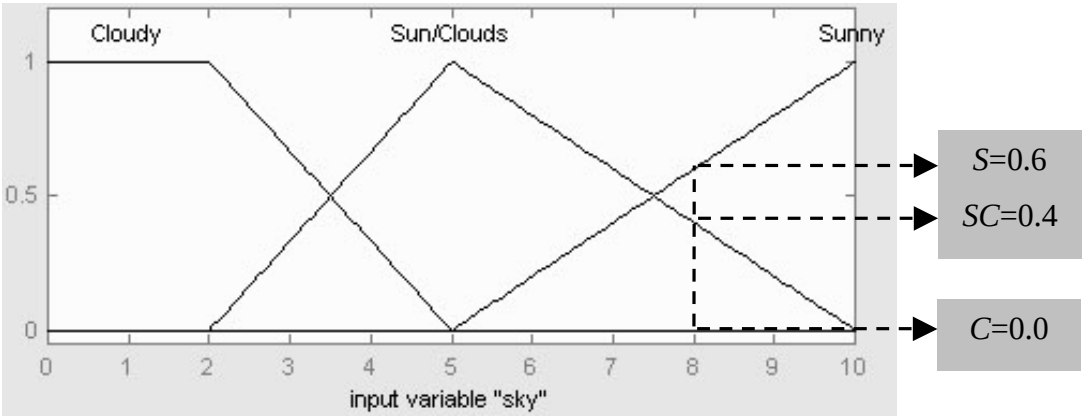


Figure 121: Fuzzification of the input variable *sky*

A similar approach may be applied to the *COR* variable where possible MF's for 'low', 'medium' and 'high' could just reflect the percentage chance of rain, say 60%, given in the forecast (note that the range for input in this case is 0 to 100). This is shown graphically in Figure 122 with the linear Open-ended Trapezoidal MFs for 'low' and 'high' and a Triangular MF for 'medium'. In this case, 'low' and 'high' were chosen to be mutually exclusive (no overlap) ranging from 0 to 50 and 50 to 100, respectively. To implement the fuzziness to the *COR* input space, the 'medium' MF was chosen to overlap both and range from 25 to 75. Fuzzification of *COR*=60% results in a membership value for 0.2, 0.6 and 0.0 for categories 'high', 'medium' and 'low', respectively.

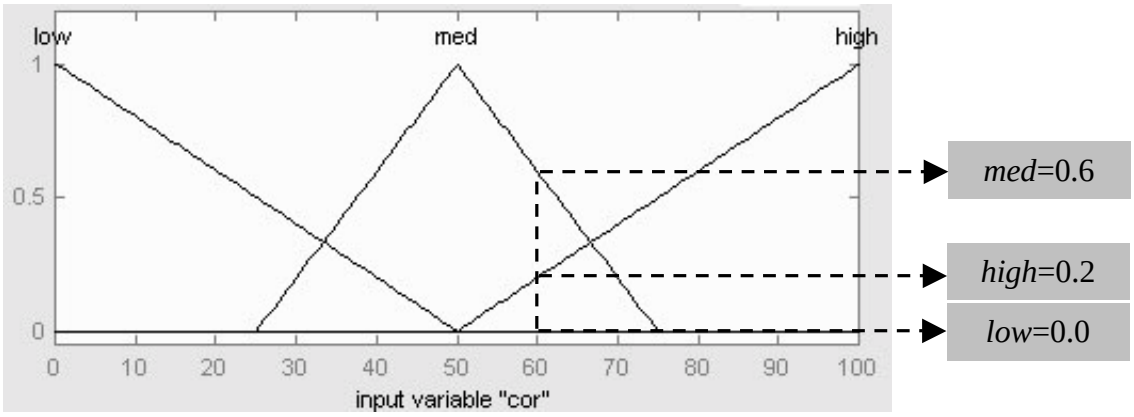


Figure 122: Fuzzification of the input variable *COR*

Step 2

In Step 2, the fuzzy operators within the antecedents of each rule we developed are processed in parallel. For example, the antecedent of Rule 1 proclaimed, 'If the sky is cloudy and COR is high'. Using *sky*=8 and *COR*=60 we found their degrees of

membership to ‘Cloudy’ and ‘high’ to be 0.0 and 0.2, respectively (see Figure 121 and Figure 122). The only fuzzy operator within the antecedent of this rule is AND, which refers to the *min* function in Fuzzy Logic. Therefore, the antecedent in Rule 1 may be transformed into its mathematical equivalent,

$$\min(\textit{cloudy}, \textit{high}) = \min(0.0,0.2) = 0.0.$$

This makes sense since, although the *COR* is above average, the day is quite nice (*sky*=8) and therefore the effect this output has on the decision to take the umbrella or not, is to help shift it into the *not* category. The influence that the state of the sky has on our decision was specified in Rule 1. However, a more careful person might choose Rule 1 to have the opposite effect, i.e., to take the umbrella if the sky is nice but the forecast is for a high chance of rain. To conclude, the antecedent of each rule of our Rule Base may be processed similarly with the results shown in Table 18.

Step 3

The Implication Step determines how the individual antecedent outputs from Table 18 are mapped to the consequent of the respective rules. The output of the Implication Step is a fuzzy *set* that, in essence, displays the importance or weight of that rule in the overall decision. For example, the output of the Implication Step using the Mamdani Inference Method would be the area under a curve defined by the shape of a MF. On the other hand, the result using the Sugeno Inference Method would be a singleton spike of a specific height.

Table 18: Outputs for antecedents for each rule within the rule base using inputs *sky*=8 and *COR*=60.

Rule	**Antecedent Output**
1	min(*cloudy*, *high*) = min(0.0,0.2) = 0.0
2	min(*cloudy*, *medium*) = min(0.0,0.6) = 0.0
3	min(*cloudy*, *low*) = min(0.0,0.0) = 0.0
4	min(*sunny w/clouds*, *high*) = min(0.4,0.2) = 0.2
5	min(*sunny w/clouds*, *medium*) = min(0.4,0.6) = 0.4
6	min(*sunny w/clouds*, *low*) = min(0.4,0.0) = 0.0
7	min(*sunny*, *high*) = min(0.6,0.2) = 0.2
8	min(*sunny*, *medium*) = min(0.6,0.6) = 0.6
9	min(*sunny*, *low*) = min(0.6,0.0) = 0.0

Since the answer to our original question (whether or not to take the umbrella) is Yes or No, we can arbitrarily define a mathematical equivalent to Yes > 0 and No < 0. Using the Mamdani Method, the output categories are MFs similar to those used for fuzzification, such as Triangular, Trapezoidal, etc. Typically, the output MFs using Mamdani Inference are overlapping, although in this case our Yes/No scenario needs to be quite exclusive since we cannot have a decision that is partially Yes or No. This can be represented using two mutually exclusive Open-ended Rectangular MFs like that in Figure 123. However, one can imagine other systems requiring overlapping MFs and with a non-arbitrary range in the output space where we may want an answer to have partial membership in many different categories. For example, we may want the voltage input to a motor to be somewhere in the middle of its maximum and minimum values, which is typically the case. Furthermore, the output range is not arbitrary in this case since we desire an input voltage to the motor that we can actually supply.

In a similar manner, using the Sugeno Inference Method would require specific functions to be defined. The simplest possibility is by defining Yes as $+c$ and No as $-c$, where c is some arbitrary constant, say 1 (a zero-order function) as shown in Figure 124. Note that this automatically fixes the middle point at zero, regardless of what the value of c is. Therefore, if the result after Defuzzification (Step 5) ends up greater than or less than zero, the answer is Yes or No, respectively (let us ignore the possibility that the result ends up *exactly* zero as we will leave the decision in this case to the user and whether or not they want to live conservatively or dangerously).

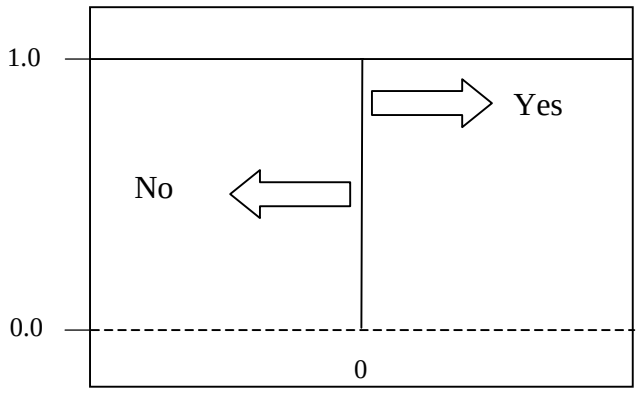


Figure 123: Implication (output) membership functions using the Mamdani Inference Method

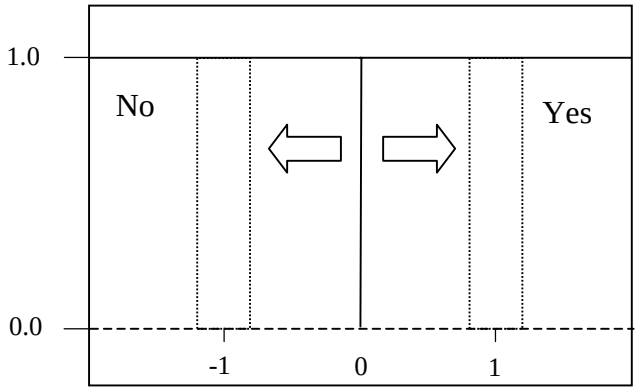


Figure 124: Implication (output) membership functions using the Sugeno Inference Method

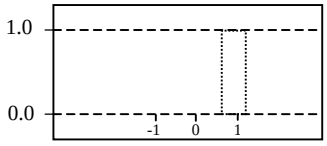
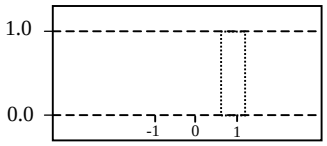
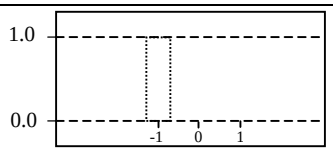
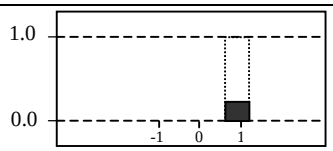
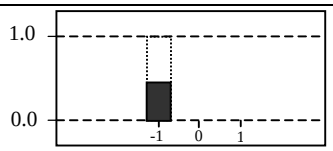
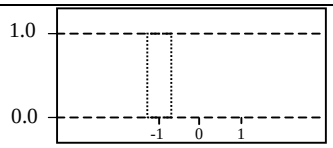
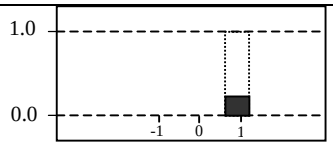
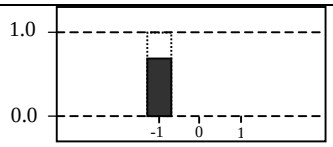
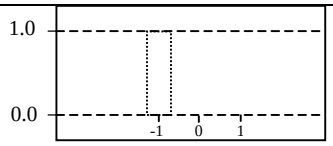
Although there are many types of implication methods, by far, the two most commonly used are the *minimum* and *product* functions. The *minimum* function acts to curtail the individual consequent fuzzy sets (*Yes* and *No* in this case) producing a truncated area-under-the-curve or a truncated spike, depending on the type of inference used. On the other hand, the *product* function acts to scale the height of the area or spike. In both cases the resulting height may be considered the strength or weight of that particular antecedent-to-consequent mapping that will be used in the aggregation of Step 4. For example, the antecedent output of Rule 5 was determined to be 0.4. Using the *minimum* function for implication means that the consequent MF is truncated at a height of 0.4.

For simplicity, the rest of the example will be described assuming the Sugeno Inference Method has been chosen. Therefore, the results of the consequents using the Sugeno Inference and the *minimum* of each rule are shown in Table 19. Note that although the truncated MFs (consequent outputs) in Table 19 are shown with having a width, this is just for illustrative purposes as the spikes actually have a width of zero. Therefore, only their height is important when using the Sugeno Inference Method.

Step 4

Now that we have the output for each rule within our Rule Base (truncated spike heights of the consequent MFs), we need to somehow combine these individual truncated spikes into a single result that keeps the respective weights of each result intact. Typically, aggregation when using the Sugeno Inference Method is to simply use the maximum height for each spike in a given category. For example, the maximum height of an individual spike for the Category 'No' (-1) is 0.6. Likewise, the maximum height of an individual spike for Category 'Yes' (+1) is 0.2. The results of aggregation for this example are shown in Figure 125.

Table 19: Results of antecedents-to-consequents of the Umbrella Decision Rule Base

Rule	Antecedent Output		Consequent Output	
1	0.0	⇨	0% Yes	
2	0.0	⇨	0% Yes	
3	0.0	⇨	0% No	
4	0.2	⇨	20% Yes	
5	0.4	⇨	40% No	
6	0.0	⇨	0% No	
7	0.2	⇨	20% Yes	
8	0.6	⇨	60% No	
9	0.0	⇨	0% No	

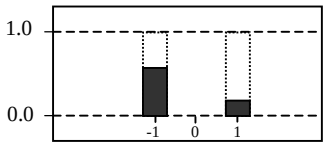


Figure 125: Aggregation of individual Rule Base outputs

Step 5

The most common method for the defuzzification of the aggregate output in Step 4 for the Sugeno Method is the Weighted Average Method. This is probably due to the fact that this method is intuitively correct for many applications. It is equivalent to finding the 'balance-point' of the overall aggregate like that in Figure 126, taking into account all outputs from the Rule Base. For this example, the exact output value of defuzzification, which happens to be -0.5, is unimportant as we are only interested in whether or not it is positive or negative. In this case, it is obvious the result will be negative and therefore the solution to our dilemma of whether or not to take out umbrella is 'No'. It may be of interest that the threshold of *COR* for us to take our umbrella when the *sky*=8 turns out to be 64%.

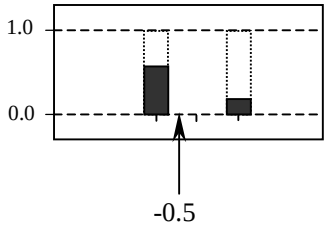


Figure 126: Defuzzification of aggregate area

References

[1] Hinch, J., Shadle, S., Klein, T.M., 1992, "NHTSA's Rollover Rulemaking Program – Results of Testing and Analysis", SAE paper No. 920581

[2] US DOT, 2000, "Comprehensive Truck Size and Weight Study", FHWA Publication number FHWA-PL-00-029

[3] Ervin, R. D., Mallikarjunarao, C., Gillespie, T. D., 1980, "Future Configuration of Tank Vehicles Hauling Flammable Liquids in Michigan – Final Report", Highway Safety Research Institute, Report No. UM-HSRI-80-73

[4] Ervin, R. D., 1986, "The Dependence of Truck Stability on Size and Weight Variables", Int. J. of Vehicle Design, Special Issue on Vehicle Safety, pp. 192-208

[5] Bernard, J., Shannan J., Vanderploeg, M., 1989, "Vehicle Rollover on Smooth Surfaces", SAE paper No. 891991

[6] Gillespie, T. D., 1986, Fundamentals of Vehicle Dynamics, Published by SAE, Warrendale, PA

[7] Blue, D. W. and Kulakowski, B. T., 1991, "Effects of Horizontal-Curve Transition Design on Truck Roll Stability", J. of Tran. Engineering, Vol. 117, No. 1, January/February, pp. 91-102

[8] Allen, R. W., Szostak, H. T., Rosenthal, T. J., Klyde, D. H., 1990, "Field Testing and Computer Simulation Analysis of Ground Vehicle Dynamic Stability", SAE paper No. 900127

[9] Allen, R.W., Szostak, H. T., Rosenthal, T. J., Klyde, D. H., Owens, K.J., 1991, "Characteristics Influencing Ground Vehicle Lateral/Directional Stability", SAE paper No. 910234

[10] Garrott, W. R., Heydinger, G. J., 1992, "Investigation, via Simulation, of Vehicle Characteristics that Contribute to Steering Maneuver Induced Rollover". SAE paper No. 920585

[11] Nalecz, A. G., 1989, "Intermediate Maneuver Induced Rollover Simulation (IMIRS) and Sensitivity Analysis", TSC-U.S. DOT Contract No. DTRS 57-88-P-82668, Final Report

[12] Nalecz, A. G., 1989, "Influence of Vehicle and Roadway Factors on the Dynamics of Tripped Rollover", Int. J. of Vehicle Design, Vol. 10, No. 3, pp. 321-346

[13] Goldman, R.W., El-Gindy, M., Kulakowski, B.T., 2001, "Rollover Dynamics of Road Vehicles: Literature Survey", *Int. J. of Heavy Vehicle Systems*, Vol. 8, No. 1

[14] Nalecz, A. G.,Lu, Z., d'Entremont K. L., 1993, "An Investigation into Dynamic Measures of Vehicle Rollover Propensity", SAE paper No. 930831

[15] Ervin, R. D. et. al, 1979, "The Yaw Stability of Tractor-Semitrailers During Cornering", Final Report Contract No. DOT-HS-7-01602, Highway Safety Research Institute, University of Michigan, Report No. UM-HSRI-79-21

[16] MacAdam, C. C., 1982, "A Computer-Based Study of the Yaw/Roll Stability of Heavy Trucks Characterized by High Centers of Gravity", SAE paper No. 821260

[17] Wong, J. Y., El-Gindy, M., 1985, "Computer Simulation of Heavy Vehicle Dynamics Behaviour – User Guide to UMTRI Models", Vehicle Weights and Dimensions Study, Road and Transport Association of Canada, Report No. 3, 1985

[18] Mallikarjunarao, C., 1981, "Road Tanker Design: Its Influence on the Risk and Economic Aspects of Transporting Gasoline in Michigan", Ph. D. Dissertation, The University of Michigan

[19] El-Gindy, M., Hosamel-deen, Y.H., 1989, "Sensitivity Parametric Analysis of UMTRI Static Roll Model", Int. J. of Vehicle Design, Vol. 10, No. 2, pp. 187-209

[20] Ervin, R. D., 1983, "The Influence of Size and Weight Variables on the Roll Stability of Heavy Duty Trucks", SAE paper No. 831163

[21] Preston-Thomas, J., El-Gindy, M., 1992, "Static Rollover Thresholds of Heavy Vehicles", CSME Forum, Symposium on Transportation Technology, Montreal, June 1-5

[22] Preston-Thomas, J., 1991, "Measured Rollover Thresholds of Three-Axle and Four-Axle Cement Mixers and Dump Trucks", NRC Report IME-GTT-CAT-001, National Resource Council Canada

[23] Woodrooffe, J. H., 1990, "Accident Reconstruction Report Involving a Fuel Tanker and a Bus at Merivale and Baseline Roads, Ottawa, Ontario, 21 September 1989", NRC Report TR-VDL-001. NRC No. 31199, National Resource Council Canada

[24] MacAdam, C. C., Fancher, P. S., Hu, G. T., Gillespie, T. D., 1980, "A Computerized Dynamics of Trucks, Tractor Semi-Trailers, Doubles, and Triples Combinations: User's Manual – Phase 4", Highway Safety Research Institute, University of Michigan, Report No. UM-HSRI-80-58

[25] Meyer, C. F., Gibson, D. W., 1980, Route Surveying and Design, 5th Edition, Harper and Row, New York, NY

[26] El-Gindy, M., 1995, "Overview of Performance Measures for Heavy Commercial Vehicles in North America", Int. J. of Vehicle Design, Vol. 16, No. 4-5, pg. 441

[27] Fancher, P. S., Ervin, R. D., Winkler, C. B., Gillespie, T. D., 1986, A Factbook of the Mechanical Properties of the Components for Single-Unit and Articulated Heavy Trucks, The University of Michigan Transportation Research Institute, Ann Arbor, Michigan

[28] Lin, R. C., Cebon, D., Cole, D. J., 1996, "Active Roll Control of Articulated Vehicles", Vehicle System Dynamics, Vol. 26, pp. 17-43

[29] Segel, L., 1956-57, "Theoretical Prediction and Experimental Substantiation of the Response of the Automobile to Steering Control", Proceedings of the Institution of Mechanical Engineers – Automobile Division, pp. 310-330

[30] Lin, R. C., Cebon, D., Cole, D. J., 1996, "Optimal Control of a Single-Unit Lorry", Proceedings of the Institution of Mechanical Engineers – Automobile Division, Vol. 210, pp. 45-55

[31] Palkovics, L., El-Gindy, M., 1995, "Design of an Active Unilateral Brake Control System for Five-Axle Tractor Semi-Trailer Based on Sensitivity Analysis", Vehicle System Dynamics, Vol. 24, No. 10, pg. 725-758

[32] Ellis, L. R., 1960, Vehicle Dynamics, Business Books Ltd., London

[33] Ranganathan, R., Rakheja, S., Sankar, S., 1989, “Steady Turning Stability of Partially Filled Tank Vehicles with Arbitrary Tank Geometry”, J. of Dynamic Systems, Measurement, and Control, Vol. 111, pp. 481-489

[34] Ranganathan, R., Rakheja, S., Sankar, S., 1989, “Kineto-Static Roll Plane Analysis of Articulated Tank Vehicles with Arbitrary Tank Geometry”, Int. J. of Vehicle Design, Vol. 10, No. 1, pp. 89-111

[35] Bauer, H. F., 1981, “Dynamic Behaviour of an Elastic Separating Wall in Vehicle Containers – Part 1”, Int. J. of Vehicle Design, Vol. 2, No. 1, pp. 44-77

[36] Rakheja, S., Piché, A., 1990, “Development of Directional Stability Criteria for an Early Warning Safety Device”, SAE paper No. 902265

[37] Miller, D. W. G., Barter, N. F., 1973, “roll-over of Articulated Vehicles”, Vehicle Safety Legislation – Its Engineering and Social Implications, Proceedings of the Institution of Mechanical Engineers, paper C203/73

[38] George, R. M., 1992, “Behaviour of Articulated Vehicles on Curves”, Heavy Vehicles and Roads – technology, safety and policy”, Proceedings of the 3rd International Symposium on Heavy Vehicle Weights and Dimensions, June 28-July 2,Cambridge, UK, pp. 331-337

[39] Liu, P. J., Rakheja, S., Ahmed, A. K. W., 1997, “Detection of Dynamic Roll Instability of Heavy Vehicles for Open-Loop Rollover Control”, SAE paper No. 973263

[40] Ervin, R. D., Guy, Y., 1986, “Volume 1: The Influence of Weights and Dimensions on the Stability and Control of Heavy Trucks in Canada – Part 1”, The University of Michigan Transportation Research Institute, Ann Arbor, Michigan

[41] Preston-Thomas, J., Woodrooffe, J. H. F., 1990, “A Feasibility Study of a Rollover Warning Device for Heavy Trucks”, NRC Report TR-VDL-005, National Resource Council Canada

[42] Sparks, G. A., Berthelot, C., 1989, “The Cost/Benefit Analysis of a Rollover Warning Device for Large Trucks”, Clayton, Sparks and Associates Ltd., Saskatoon, Saskatchewan, Canada

[43] Barnett, J. D., West, R., 1983, “A New Load Sensor for Truck Self-Weighing Systems”, SAE paper No. 830103

[44] Stability Dynamics Ltd, 1998, “Lateral Acceleration Warning Device Testing”, submitted for review to the Department of Public Safety, DFW International Airport.

[45] TNO Road-Vehicles Research Institute, *Heavy Vehicle Rollover Stability & Safety*, brochure, published by TNO, Netherlands.

[46] Anonymous, 2000, TruckSim User Manual, Mechanical Simulation Corporation, version 4.6, 709 W. Huron, AnnArbor, MI

[47] Anonymous, 1998, CarSim Reference Manual, Mechanical Simulation Corporation, 709 W. Huron, AnnArbor, MI.

[48] Pacejka, H., Sharp, R.S.,1991, “Shear force development by pneumatic tyres in steady state conditions: a review of modelling aspects”, *Vehicle System Dynamic*s, vol. 20, no. 3-4, pp. 121-176.

[49] Chrstos, J.P, and Heydinger, G.J., 1997, "Evaluation of VDANL and VDM road for predicting the vehicle dynamics of a 1994 Ford Taurus", *Research into Vehicle Dynamics and Simulation*, SAE paper No. 970566, pp. 119-133

[50] El-Gindy, M., Mikulcik, E.C., 1993, "Sensitivity of a vehicle's yaw rate response: application to a three-axle truck", *International Journal of Vehicle Design*, vol. 14, no. 4, pp. 325-352.

[51] Russell, S., 1913, "A practical device to simulate the working of nervous discharges", *J. of Animal Behaviour*, vol. 3

[52] Dayhoff, J., 1990, Neural Network Architectures – An Introduction, Van Nostrand Reinhold Publ., New York, NY

[53] McCulloch, W.S., Pitts, W.H., 1943,, "A logical calculus of ideas immanent in nervous activity", Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133

[54] Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986, "Learning representations by back-propagating errors", *Nature*, vol. 323, pp 533-536

[55] Demuth, H., Beale, M., Neural Network Toolbox User's Guide, The Math Works, Inc., 3rd version

[56] Jacobs, R.A., 1988, "Increased rates of convergence through learning rate adaptation", *Neural Networks*, vol. 1, no. 4, pp. 295-308

[57] Fletcher, R., Reeves C., 1964, "Function minimization by conjugate gradients", *Computer Journal*, vol. 7, pp. 149-154

[58] Hagan, M.T., Demuth, H.B., Beale, M.H.,1996, Neural Network Design, PWS Publishing, Boston, MA

[59] Marquardt, D., 1963, "An algorithm for least squares estimation of non-linear parameters", J. Soc. Ind. Appl. Math, pp. 431-441

[60] Hagan, M.T., Menhaj, M.B., 1994, "Training feedforward networks with the Marquardt Algorithm", IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989-993

[61] Rosenblatt, F., 1957, "The Perceptron: A perceiving and recognizing automation (project PARA), Cornell Aeronautical Laboratory Report, 85-460-1

[62] Munakata, Toshinori, 1998, Fundamentals of the New Artificial Intelligence – Beyond Traditional Paradigms, Springer Publishing, New York, NY

[63] Widrow, B., Hoff, M., 1960, "Adaptive switching circuits", 1960 WESCON Convention Record: Part 4, pp. 96-104

[64] Peel, C., Willis, M.J., Tham, M.T., 1993, "Enhancing feedforward neural network training", Application of Neural Networks to Modelling and Control, Chapman & Hall Publ., London, pp 35-52.

[65] Anonymous, 1998, Fuzzy Logic Toolbox User's Guide, The Math Works, Inc., version 2, Third printing

[66] Mamdani, E.H., S. Assilian, 1975, "An experiment in linguistic synthesis with a fuzzy logic controller", International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13.

[67] Sugeno, M., 1985, Industrial Applications of Fuzzy Control, Elsevier Science Publishing Co.

Vita

Robert Goldman was born in Lubbock, Texas in 1971 and spent most of his childhood and teen years in Delaware and Pennsylvania. After graduating from Unionville High School in Unionville, Pennsylvania, Robert attended Penn State University for Mechanical Engineering and graduated with a B.S. in 1995. Robert then pursued an advanced degree at Penn State, graduating with an M.S. in Mechanical Engineering in 1997. For the past four years, he has been a Ph.D. candidate in the Department of Mechanical Engineering and since 1997 has been an advisor to undergraduate students in the Department and has taught courses in Mechanical Design. Recently, Robert has accepted a full-time position at the Naval Undersea Warfare Center in Newport, RI.