UNIVERSITY OF SOUTHAMPTON

THE STUDY OF MOTORWAY OPERATION USING A MICROSCOPIC SIMULATION MODEL

by

BESHR SULTAN

A thesis submitted for the degree of
DOCTOR OF PHILOSOPHY

FACULTY OF ENGINEERING AND APPLIED SCIENCE
CIVIL AND ENVIRONMENTAL ENGINEERING

June 2000

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
CIVIL AND ENVIRONMENTAL ENGINEERING

Doctor of Philosophy

THE STUDY OF MOTORWAY OPERATION USING A MICROSCOPIC SIMULATION MODEL

By Beshr Sultan

In general motorways have been built to provide a better level of service for long distance travellers and to separate them from local traffic. During the last two decades, traffic demand on motorways has grown considerably, and a phenomenon termed "flow breakdown" has become more prevalent. Flow breakdown occurs when there is a dramatic reduction in speed in congested condition.

This research is concerned with the study of the behaviour of the vehicle/driver combination on motorways, in order to understand the effects which occur at higher flows and the potential to alleviate the problems. A simulation model was developed on the basis of the action point model, where the drivers' thresholds of perception and action were calibrated and measured by analysing vast sets of microscopic data.

A comprehensively instrumented vehicle was used to collect the data for the project. A filter programme was developed for the raw data to overcome the effects of vibration and noise, which occurs due to many dynamic factors and circumstances in collecting the data. Three main situations were distinguished in developing the simulation model:

1) The Approach Process: This is concerned with the behaviour of a driver when approaching slower vehicle ahead. The research shows drivers are more likely to rely on the time-to-collision in controlling their approach process. However, other factors were measured, such as the perception threshold and the deceleration level.

2) The Following Process: This is concerned with the behaviour in close following situation. Many understandings were obtained from the analysis, probably the main one was that drivers tend to follow with shorter headways at high speeds, and no obvious threshold was found for the relative speed. Other behavioural elements investigated included driver speed control and the deceleration level during the close following process.

3) The Lane Changing Process: This is a very complicated process, which has many parameters and factors that are difficult to model. Relative speed and distance were used to determine thresholds for drivers' decisions to change lane. However, a complicated procedure was necessary to achieve a realistic behaviour in the model. The model was validated by comparing the output with the real traffic data.

The calibrated and validated simulation model was further developed to investigate and assess the effects of two different Automatic-Cruse-Control [ACC] systems on traffic on motorways. The research shows that in general ACC systems work better with longer headway in reducing speed and acceleration variation. The comparison between the two tested systems showed that each has its own positive and negative side. Finally the safety benefits of such systems probably cannot be gained until the percentage of the equipped vehicles are very high.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

بسم الله الرحمن الرحيم

ACKNOWLEDGEMENTS

I would like to thank and express my sincere gratitude to:

My supervisor, Professor M. McDonald for his guidance and continuous encouragement throughout the research period.

Dr. M. Brackstone and other members of Transportation Research Group of University of Southampton for valuable discussion at different stages of this study.

Mr. B. Mould for his technical support in programming and developing the Instrumented Vehicle.

My parents and my wife for their continuous support and encouragement to achieve the success in this life.

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND:

In modern societies, Motorways have been built in order to separate the long distance travel vehicles from the local traffic. During the last two decades the traffic demand on motorways has grown dramatically. The delays in peak hours have become considerable, affecting not only drivers and passengers but also businesses and industry. Reduction in traffic congestion on motorways is very important and necessary.

When a traffic queue starts to build up on a motorway a reduction in speed, and sometimes flow, usually happens suddenly rather than gradually. This phenomenon is called Flow Breakdown. Many Studies have been undertaken to determine the factors which are significant in causing such traffic flow breakdowns, but no definite causes have been found. Simulation models are one of the most useful tools in the study of traffic management problems which exhibit random parameters and variations. Therefore, a good simulation model that is well calibrated and validated will certainly be a major step in studying traffic congestion and searching for solutions to reduce the delay and cost on motorways.

1.2 OBJECTIVE OF THE RESEARCH:

The objectives of this research are:

1- To Contribute to the understanding of the behaviour of drivers on motorways in close following / lane changing situations.

2- To develop a simulation model able to be used in investigating a range of potential solution to motorway traffic problems.

3- To apply the model to develop new understanding of the impacts of Automatic Cruise Control (ACC) system on motorway traffic operations.

1.3 METHOD OF APPROACH:

The approach has been to explore the higher flow motorway situations using a simulation model. There are several stages in this process:

(i) The exploration of drivers' behaviour to set the main characteristics and fundamental specification of the model.
(ii) The development of a model.
(iii) The calibration and validation of the model's parameters.
(iv) The application of the model and interpretation of the results.

There are many inter-relations between the various stages in the approach which are linked by a comprehensive literature review and a wider content of results and applications. A key requirement of the objectives of the project is a sufficiently detailed database of individual vehicle's motion characteristics. It is virtually impossible to collect such a database by the classic way of measuring motorways data i.e. video cameras and speed detectors. Instead, a highly instrumented vehicle was developed to collect this database. The vehicle has a radar, two video cameras, speedometer, accelerometer and other instruments. It can provide information about the instrumented vehicle's speed and acceleration and the relative speed and distance with other vehicles.

The computer simulation model was built to replicate the movement of vehicles on motorways. Two main types of computer simulation model have been traditionally used to describe traffic flow: Macroscopic and Microscopic models. Macroscopic models employ approaches such as a fluid analogy in which traffic flow is seem as a fluid in which shock waves, etc occur. Whereas, microscopic simulations replicate the movement of individual vehicles in traffic stream.

The difficulty in defining the precise capacity for a motorway, because the traffic is influenced by individual behaviour, has made microscopic models more suitable for simulating traffic on motorway, provided the behaviour of individual drivers can be understood sufficiently. Any satisfactory model must contain a car-following logic that can replicate the decision and movement of individual drivers. In particular lane changing logic which is more complicated needs to be potentially well calibrated. This research has been possible because of the availability of obtained data on individual driver behaviour from an instrumented vehicle.

CHAPTER 2

LITERATURE REVIEW

2.1 MOTORWAY CAPACITY AND SPEED-FLOW RELATIONSHIPS:

Speed-flow relationships are fundamental to the understanding of traffic flow phenomena on Motorways. These relationships have been studied for more than (60) years, and it is important to note the significant changes that have taken place over this period.

GREENSHIELDS (GREENSHIELDS 1938)[1] suggested that the relationships between flow and densities as well as between speed-flow had a parabola-shaped. Consequently, speed will be a linear function of density. Thus, when an average flow is given, there are two possible average speeds. The upper half of the parabolic curve gives free flow situations (high speed) and the lower half indicates constrained situations (slow speed), Figure 2.1.

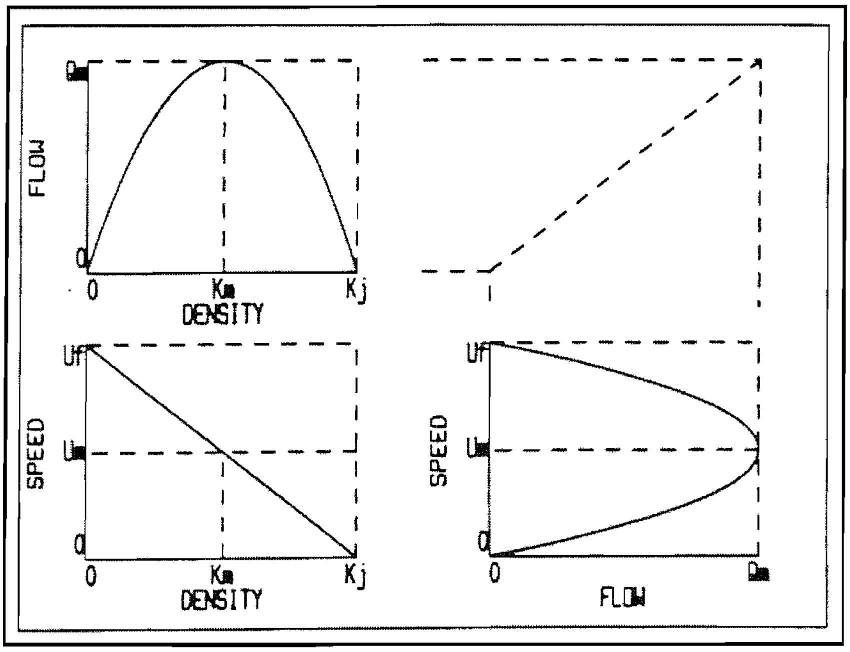


Figure 2.1: The fundamental relationships between speed, density and rate of flow, (GREENSHIELDS-BD 1938)[1]

In 1950 with the publication of the first Highway Capacity Manual (HCM) a major development in the use of speed-flow relationships occurred (MAY 1994)[2]. The HCM gave a procedure for calculating what was termed practical capacity. It proposed that the capacity under ideal conditions is 2000 passenger car per hour per lane (pcphpl), and the practical capacity occurs at an approximate Volume / Capacity (V/C) ratio of 0.75.

In 1965 the HCM (HIGHWAY CAPACITY MANUAL, 1965)[3] introduced the level of service concept based on the V/C ratio. The speed-flow relationships in the 1965 HCM are shown in Figure 2.2 with the horizontal scale normalised as the V/C ratio. The value of the ideal capacity remained at 2000 pcphpl under ideal conditions.

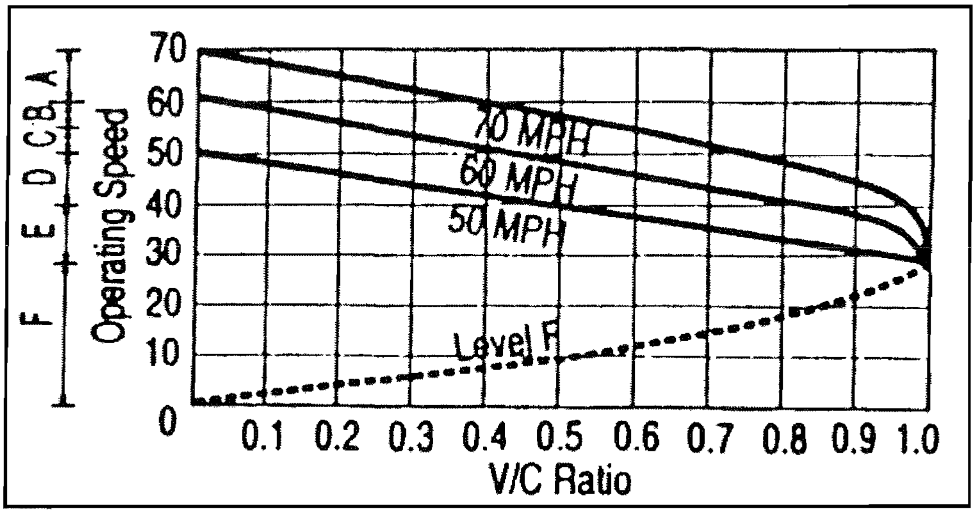


Figure 2.2: Speed-Flow relationships (HIGHWAY CAPACITY MANUAL, 1965)

Later, in 1985 the shape of the higher portion of the speed-flow curve continued to be considered parabolic with a rate of reduction of speed with increasing V/C (HIGHWAY CAPACITY MANUAL, 1985)[4]. Maximum capacity was expected to occur at speeds about 35 mph. Greater emphasis was given to density, and it was used as the traffic parameter on which level of service was determined. Although, the 'ideal' capacity continued to be 2000 pcphpl, many sites were reported to convey more than 2000 pcphpl. The 1985 HCM speed-flow relationships are shown in Figure 2.3.

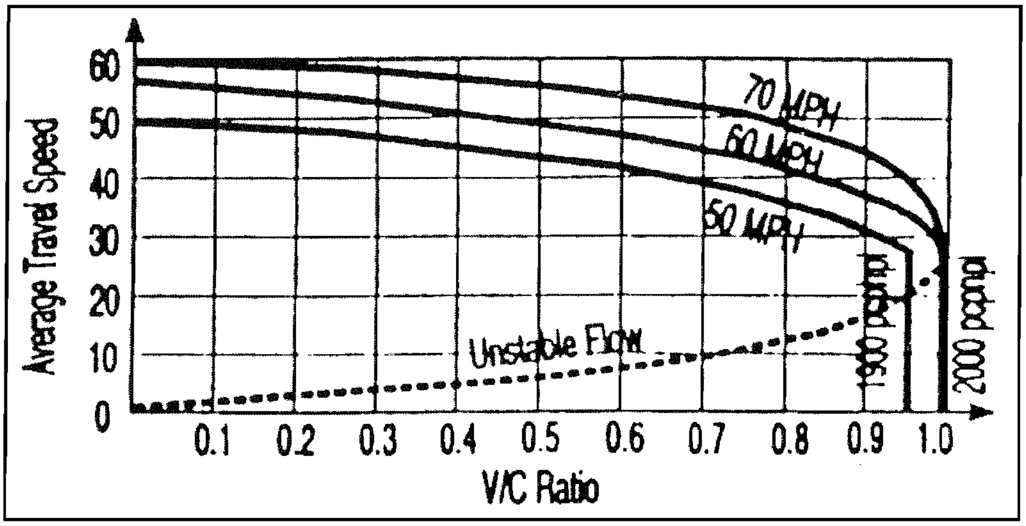


Figure 2.3: The Speed-Flow relationships from the (HIGHWAY CAPACITY MANUAL 1985)[4]

In 1994, the level of service concept was continued, with emphasis on density (HIGHWAY CAPACITY MANUAL, 1994)[5]. Two significant changes were incorporated. First, the capacity under ideal conditions was increased by 10 percent, from 2000 to 2200 pcphpl. Improvements in the vehicle fleet and driver capabilities were considered to be the reason for the increase. The second change was in the shape of the upper portion of the speed-flow relationship as well as the recognition of a multiregime relationship. The speed-flow relationships in the 1994 HCM is shown Figure 2.4.

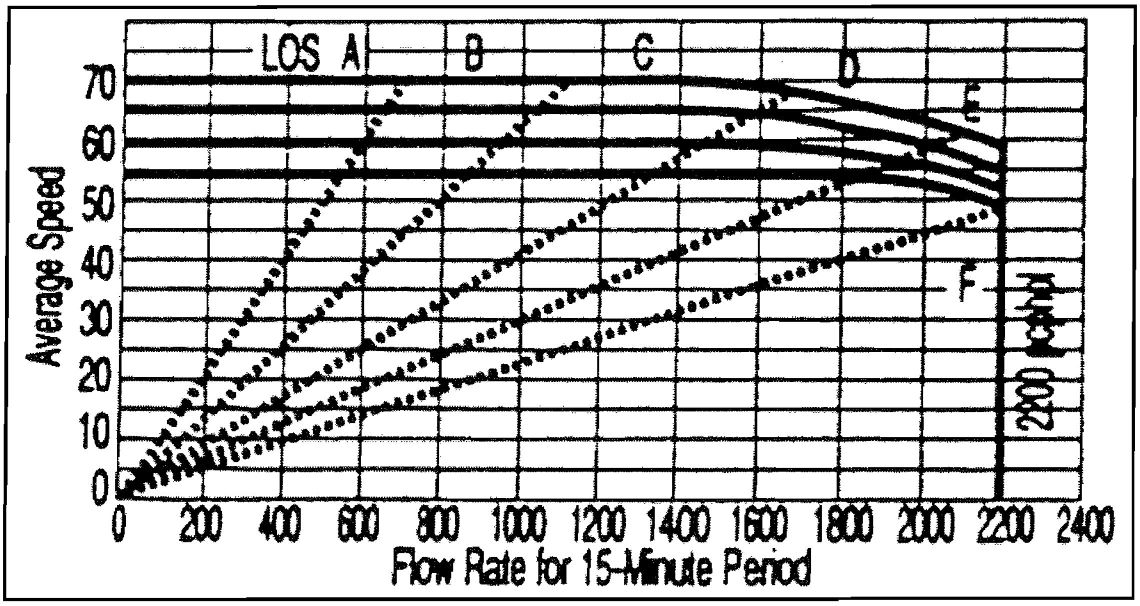


Figure 2.4: The 1994 HCM Speed-Flow Relationship.

Although the relationship between the speed and the flow are shown as a single line, there is considerable variability in practice. Many empirical studies have been undertaken in the recent years to investigate speed-flow and flow occupancy relationships under different circumstances. F. L. Hall and J. H. Banks distinguished between three stages (F. L. HALL AND J. H. BANKS 1992)[6]:

1- Not congested area.
2- Queue discharge.
3- Congested area.

They even reported that no single location could provide the full range of the stages at one time. Indeed the data needed to create even one segment for queue discharge operation must come from a series of locations.

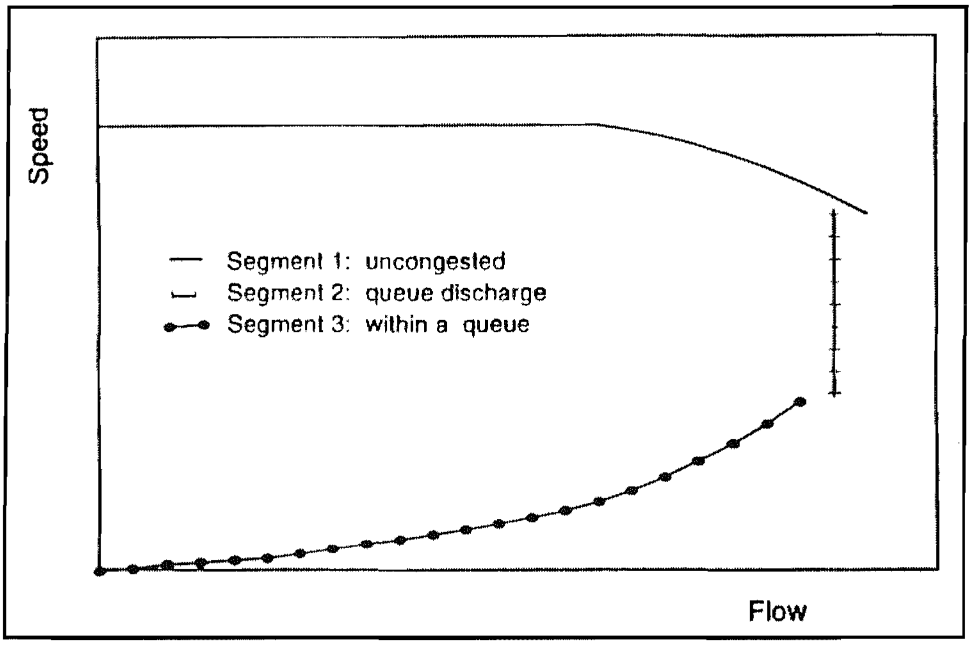


Figure 2.5: Speed-Flow relationship proposed by (HALL & BANKS 1992)[6]

Another comprehensive study by HALL, SMITH and MONTGOMERY concluded that the breakdown flow rate varies with the free-flow speed, while it was far from clear whether or not there was a correlation between the capacity and free-flow speed (HALL F. L & Smith W. S. & Montgomery F. 1994)[7].

Also, by observing motorway bottlenecks in Texas RINGERT and URBANK reached four main conclusions, (RINGERT J. & URBANK T.1993)[8],They are:

1- Variance in Flow rate decreases after the speed drop to queue discharge.
2- Peak flows for individual lanes occur in free-flow conditions before flow breakdown.
3- Bottleneck configuration may influence the maximum possible flow obtainable during free-flow and possibly queue discharge conditions.
4- Queue discharge appears to be the best estimate for maximum sustainable flow and capacity.

Finally, HOUNSELL and MCDONALD (HOUNSELL N. B. and MCDONALD M. 1994)[9] investigated the factors which are significant in causing traffic flow breakdowns on UK. motorways. They found from the data (600 hours of recording Video tape) that:

1- Speed breakdown initially occurs in the offside lane (the fastest lane). Whether this led to full flow breakdown depended very much on the level of demand and the distribution of traffic flow between lanes.
2- Flow breakdowns usually occur between the end of the merge and approximately 2 Km downstream.
3- Breakdown was characterised by a sudden reduction in speed of at least 20-30 km/h.
4- Traffic operation immediately before a speed breakdown in the offside lane appeared to be particularly unsafe.

In conclusion, studying traffic on congested motorways and the flow breakdown phenomenon empirically enables some understanding of its characteristic. However, the following questions cannot be fully answered using empirical database alone:

- How does it happen?
- Why does it happen?
- When does it happen?

2.2 SIMULATION MODELS:

The remarkable growth in personal-computer power has made simulation models more attractive and accessible. In transportation, simulation models are used to simulate the traffic behaviour on roads to identify solutions for a specific problem. They offer many benefits for researchers, such as:

1. Allowing the study of complex traffic systems in the laboratory rather than in the field.
2. The ability to investigate road and traffic situations which do not currently exist.
3. The capability to describe stochastic process.

There are two main types of traffic simulation models:

2.2.1 Macroscopic Models:

These models are concerned with the movement of vehicles in quantities (flow, average speed and density) rather than individually. Macroscopic simulation has attracted considerable interest since the 1960s, and because of the lack of universally accepted traffic flow theory and varying operational characteristics, each model was developed largely through intuition. (PAPAGEORGIOU M. 1995)[10]. Two samples of the macroscopic models are:

1- **SIMAUT Model**: Proposed by MORIN (MORIN 1985)[11], it is based on the hydrodynamic theory of traffic flow (shock-wave propagation). The model displays a photograph of the different dynamic cells along the motorway in terms of average speed, flow and density at the end of each time slice usually between 2 - 15 min.

2- **META Model**: Developed by PARAGEOGIOU in 1989 (PAPAGEORGIOU M. 1995)[10], it is based on a geometrical subdivision of a given freeway axis into several sections, the length of which may be chosen up to 1000m. META describes the time evolution of three traffic variables: Traffic density, mean speed and traffic volume.

As macroscopic models portray traffic in terms of aggregate parameters such as volume, speed and density, they take much less time to simulate the movement of a large number of vehicles over a large network with limited computational resources. However, the assumption of similarity to a fluid obscures the fact that traffic flow is composed of the movement of individual vehicles. This is a limitation at higher flow levels, as the maximum flow rate at flow breakdown is not stable, even at the same location, and it is dependent on the behaviour of individual drivers (HALL F. L & Smith W. S. & Montgomery F. 1994)[7].

2.2.2 Microscopic Models:

These models consider the movement of vehicle individually. There are two basic movement to any vehicle, the longitudinal movement (Car Following) and the lateral movement (Lane Changing). Car following models represent the behaviour of drivers when they are approaching or following a slower vehicle. Three types of approaches can be distinguished in specifying the driver decision process in microscopic models:

2.2.2.1 Neural-Network Approaches:

Neural network computing attempts to mimic the functionality of the brain in a very fundamental manner. A neural network has a large number of simple, interconnected processing units, which are arranged in a number of layers. The input layer receives the information from the outside world which is then passed through the neural network to generate an output. In order to make the neural network understand the relationship between the input and the output, it needs to be repeatedly exposed to a large amount of input/output pairs during a training process. Although the neural network can establish input/output relationships they provide no explanation for their decisions (LYONS G. 1995)[12]. Moreover, distinguishing between drivers' behaviour becomes more difficult in a neural network approach.

2.2.2.2 Fuzzy Logic Approaches:

This approach is in some way similar to the neural network one, in that it relies on the assumption that driver decisions are determined according to a set of vague rules developed through experience (KIKUCHI & CHAKROBORTY 1993)[13]. The output from a Fuzzy Logic interface would form the drivers' decision. The difficulty with this approach is finding the proper rules and their distributions and how they influence driver decisions.

2.2.2.3 Mathematical Approaches:

Mathematical approaches are widely used to represent drivers' decisions process. Much of the research has related to the acceleration or deceleration process ie. car following. The equations which describe the logic are calibrated using appropriate data for every related situation. The Following two paragraphs will cover the literature for Mathematical Approaches for car following and lane changing models.

2.3 THE MATHEMATICAL CAR-FOLLOWING MODELS:

Car following models are concerned with the behaviour of drivers when they are approaching or following a slower vehicle ahead. In order to represent the characteristic of individual drivers in the traffic flow realistically, this logic may need to be complex. Many car-following models have been proposed, but the basic theory describing the individual models can be classify into three different types:

2.3.1 Stimulus-Based Models:

This car-following model assumes that the acceleration of any vehicle is determined by the driver reactions to speed differences and relative distance:

$$a_n(t) = C * V_n^m \frac{[V_{n-1}(t-T) - V_n(t-T)]}{[X_{n-1}(t-T) - X_n(t-T)]^l} \qquad .....2.1$$

Where:

$V_{n-1}(t)$, $X_{n-1}(t)$: are the speed and the location of the lead vehicle at the moment t.

$V_n(t)$, $X_n(t)$: are the speed and the location of the following vehicle at the moment t.

l, m, C : are parameters should be estimated from empirical data.

This approach was first developed in 1958 (CHANDLER, HERMAN AND MONTROLL 1958)[14] at the General Motors research labs in Detroit. They calibrated the model by using wire-linked vehicles to examine the responses of eight test subjects to a 'realistic' speed profile of a lead vehicle.

Their main results are:

1- ΔX contributed little to influence drivers' acceleration in the following process.

2- The scaling Constant showed a high variation between subjects.

In 1959, the model was improved theoretically (HERMAN &MONOTROLL 1959)[15]. It was suggested that the spontaneous fluctuation in the drivers' acceleration might be the reason of the difference of the r^2 value from unity for ΔV. Many attempts followed to calibrate the parameters in the model, but these parameters did not have any obvious connection with either the driver or the vehicle, which made accurate calibration virtually impossible. Table 2.1 shows a summary of the research that has been done in order to estimate the values of optimal parameters [m & l] for the stimulus-base model since HERMANN and MONTROLL developed this model (BRACKSTONE M. 1996)[16].

Source	M	l	Approach
Chandler, Herman and Montroll (1958)	0	0	Micro.
Gazis, Herman and Potts (1959)	0	1	Macro.
Herman and Potts (1959)	0	1	Micro.
Helly (1960)	1	1	Macro.
Gazis, Herman and Rothery (1961)	0-2	1-2	Macro.
Keller and May (1967)	0.8	2.8	Macro.
Heyes and Ashworth (1972)	-0.8	1.2	Macro.
Hoefs (1972) (Dc no brk / Dc brk / Ac)	1.5 / 0.2 / 0.6	0.9 / 0.9 / 3.2	Micro.
Treiterer and Myers (1974) (Dc / Ac)	0.7 / 0.2	2.5 / 1.6	Micro
Ceder and may (1976) (Single regime)	0.6	2.4	Macro.
Ceder and may (1976) (uncgd. / cgd.)	0/0	3/0-1	Macro.
Aron (1988) (DC / ss / Ac)	2.5 /2.7 / 2.5	0.7 / 0.3 / 0.1	Micro.
Ozaki (1993) (Dc / Ac)	0.9 / -0.2	1 / 0.2	Micro.

Table 2.1: is a summary of optimal parameter combinations for the stimulus-base model (BRACKSTONE M. 1996)[16]

Key: Dc. / Ac. : deceleration / Acceleration
brk. / no brk. : deceleration with and without brakes.
uncgd. / cgd. : uncongested / congested .
ss. : steady state.

2.3.2 Stopping Distance-Based Model:

This model was developed in 1981 (GIPPS 1981)[17]. The main assumption is that the driver of a following vehicle selects his or her speed to insure that he or she can bring the vehicle to a safe stop if the leader stops suddenly. (i.e. taking the following driver's reaction delay into account). The equation for this model is:

$$V_n(t+T) = Dc_n * T + \sqrt{} (Dc_n^2 * T^2 - Dc_n*(2*[X_{n-1}(t) - S_{n-1} - X_n(t)] - V_n(t) *T - V_{n-1}^2(t) / Dc_{n-1}) \quad \ldots 2.2$$

Where:

Dc_n : Is the maximum deceleration that the driver of the following vehicle wished to undertake.

Dc_{n-1} : Is the predicted maximum deceleration by the lead driver.

S_n : Is the minimum safe margin between two successive vehicles plus the length of the lead vehicle.

T : Is the reaction time of the driver.

$V_n(t)$: Is the speed of the following vehicle at the moment (t).

$X_n(t)$: Is the location of the following vehicle at the moment (t).

This model has the advantage over the previous one that all its parameters have an identifiable physical meaning that represents characteristics of the drivers or vehicles. However, the overall behaviour of vehicles generated with this type of model becomes rather too stable when an accurate maximum deceleration b_n is given to each vehicle. Also, this model is not always effective, especially in situations where the lead vehicle applies a maximum deceleration and comes to a complete stop.

2.3.3 The Action Point Model:

This model came from MICHAELS & TODOSIEV (MICHAELS, R. M. 1963)[18] (TODOSIEV, E. P. 1963)[19]. Who suggested that drivers would initially be able to tell that they were approaching a vehicle due to changes in the apparent size of the vehicle, and by perceiving relative velocity through changes on the visual angle subtended by the vehicle ahead. The human threshold of the change in the visual angle (dθ /dt) [which is called then The Optic Flow] is: [3 - 10 * 10^{-4}] with a mean at (6 * 10^{-4}).

$$Tg(\frac{\vartheta}{2}) = \frac{W}{2 * DX} \quad \Rightarrow \quad \vartheta = 2 * arctg(\frac{W}{2 * DX}) \quad \Rightarrow$$

$$\frac{d\vartheta}{dt} = \frac{-4 * W * DV}{4 * DX^2 + W^2} \qquad \text{.....2.3}$$

If a driver remained below the threshold of visual angle, the action will be based on any perceived change in spacing. The change in space should be above the Just Noticeable Difference [JND], which is about 12% of the relative distance (HOFFMANN 1966)[20]. Many empirical studies have been undertaken in order to investigate thresholds of perception and WIEDEMANN (LEUTZBACH W. & WIEDEMANN R. 1986)[21] applied the concept in a simulation model. The main assumption is that driver response to avoid a collision with a slower vehicle ahead would be by adopting a constant deceleration using the equation:

$$Dc_n = -\frac{[V_{n-1}(t-T) - V_n(t-T)]^2}{2[X_{n-1}(t-T) - X_n(t-T) - S]} + Dc_{n-1} \qquad .....2.4$$

Where :

Dc_n , Dc_{n-1} : The deceleration of the following vehicle and the lead vehicle..

V_n , V_{n-1} : The speed of the following vehicle and the lead vehicle.

X_n , X_{n-1} : The position of the following vehicle. and the lead vehicle.

S : The length of the lead vehicle plus the space margin between the two vehicles when they come to stop.

WIEDEMANN's model, ie. MISSION model, was the first action point model used to simulate traffic on motorways. However, many modifications can be made to Mission model to make the output more realistic. A complete review to the Mission model has been made and described in Chapters 7 and 8, to enable an improved simulation model to be developed.

2.3.4 Conclusion:

Although the previous models are carefully designed, they generally cannot simulate congested situations on motorways and the associated breakdown phenomena for several reasons. These vary from model to model:

a- For the Stimulus based Model:

1- It has many parameters which do not relate to any physical assessment of driver behaviour. This makes calibration very difficult, and the output varies according to the situation at which the calibration has been undertaken.

2- The change in the value of the parameters according to the situation on the motorway makes it impossible to realistically simulate how the level of service on motorways changes.

b- For the Safe Distance Model:

1- The output of this model is rather too stable for real traffic on congested motorways.

2- The headway between vehicles generated by the model are longer than on congested motorways.

c- For The Action point model:

1- The assumption that drivers perceive the absolute value of the relative speed and relative distance is unrealistic. Researches have shown that the uncertainty in human perception usually means that drivers under estimate the time needed to catch the vehicle ahead.

2- The Model always propagates the interaction in a platoon of vehicles by adding the acceleration of the lead vehicle to the acceleration of the following one. This generates a high rate of deceleration which is not realistic.

The only calibrated action point model is the Mission model, which has been calibrated for German situations. However, it is likely that the characteristics of drivers vary from one country to another. U.K Drivers may have quite different characteristics.

2.4 THE MATHEMATICAL LANE CHANGING MODEL:

The lane-changing model controls the lateral movement of vehicles on motorways. Due to the complexity in the lane changing decision, it is very difficult to simulate the event accurately. Several major macroscopic investigations have been undertaken into relationships between lane changing rates, flows and even the speed of the differing lanes. eg. (SPARMANN 1979)[22] and (CHANG & KAO 1991)[23]. Other studies have attempted to investigate the lane-changing manoeuvre microscopically. WORRALL and BULLEN demonstrated that lane changing rates rises with the flow initially and then decreases (WORRALL and BULLEN 1970)[24]. They also made three other observations:

1- The lane changing time varies between 2 sec at a flow of 1500 vph/lane and at 65 kph, to 3.5 sec at 300 vph/lane at 125kph.

2- There is no noticeable relationship between the size of gaps accepted in the target lane and the density in that lane.

3- The accepted gaps tended to increase as the speed of the manoeuvre increased. It was believed likely that there may be some dependence on either the lead or the lag gap in an individual lane change, but not both.

Later work by PAHL (PAHL 1972)[25] confirms the general magnitude of the accepted gap sizes with total gaps reducing from 6 to 3.7 sec as flow increases, and lag gaps reducing from 2.7 to 1.8. More recent work using data on over 2000 lane change events found no relationship between headways (lead or lag) and the speed or relative speed of the vehicle involved (MCDONALD & BRACKSTONE 1994)[26].

In view of the above, it is unsurprising that the microscopic models have been based largely on a number of well held beliefs about the decision making process, with little experimental calibration.

2.4.1 The Safety Based Model:

The idea of this approach is that if the driver cannot obtain a speed nearer to his or her desired speed by more than a certain amount, then he or she would try to lane change. The driver's decision will depend on how much the rear vehicle in the new lane will decelerate to avoid an accident. Gipps' model (GIPPS 1986)[27] is an example of this type. The problem with this model is that does not replicate an increase in driver's desire to change lane when he or she is delayed behind a slower vehicle. Moreover it does not answer the question as to when the drivers will change lane.

2.4.2 The Action Point Model:

This approach was primarily developed by SPARMANN (SPARMANN 1978)[22], and subsequently incorporated into the MISSION model (WIEDEMANN, R. & REITER, U. 1992)[28]. The model represents human estimation of distances and speed differences in lane changing decisions as for car following. New thresholds (SDXP) and (SDVP), delimit the area of potential influence of vehicle in neibourhood lanes, are defined as multiples of the thresholds delimiting the actual influence (SDX) and (SDV) which are the maximum acceptable following distance and the relative speed threshold in approaching vehicles respectively:

SDXP = FX * SDX

SDVP = FV * SDV

FX, FV: vary according to the lane and type of lane changing that could be considered.

Although these factors define a range, within which a vehicle will influence the lane changing decision, they do not describe the conditions under which drivers would actually change lane. (eg. the speed difference, or the preferred gap size). It can only be assumed that these choices are sampled from normal/uniform distribution.

2.4.3 The Scoring and Threshold Model:

This method has been implemented in the SISTM model (WOOTTON & JEFFREYS 1990)[29]. It assesses the driver's desire to change lane by the stimulus he or she feels from a range of scaled factors. Two types of stimulus factors have been distinguished: the stimulus factors to move to the right and the stimulus factors to move to the left. Every factor replicates a certain type of stimulus such as the relative speed of the lead vehicle, the speed advantage (The desired speed of the following driver minus the speed of the lead vehicle), the relative distance ... etc.

Once these stimuli have been compiled, they are compared with threshold values and, if exceeded, the driver will move into the new lane, providing any driver closing from the rear in the new lane does not have to decelerate than a pre-set amount. The criticism of this model is that there is no particular way to calibrate the factors. However, drivers must have stimulus factors that influence overtaking decisions, otherwise the manoeuvres would be the same in a range of different circumstances.

2.4.4 Other Studies:

The three models described above were concerned with the driver's decision to overtake when there is a restriction. None considers the driver's decision when there is no restriction to change lane. Yousif investigated the relation between flow rate and the lane change frequency per km on dual carriageway motorway regarding drivers' actions after they overtook slower vehicles (YOUSIF S.1995)[30]. He considered five types of action:

1- stay in current lane.

2- move to slower-speed lane.

3- move to a lane according to the desired speed.

4- move to a lower density lane.

5- Move to faster-speed lane.

Every action was applied alone in a simulation program and the results were acceptable except with action (2). This suggests that it is not realistic to assume that all drivers will move back to slower speed lanes.

2.4.5 Conclusion:

Clearly, it is very difficult to develop an accurate lane-changing model, due to the complexity in the driver decision process and the potentially high number of parameters that influence the decisions. In general, previous models have been developed before the researchers looked at the data. The models were then calibrated and/or validated using available data, very often the data was not adequate to support the calibration and validation processes accurately. The methodology in this project is to develop the model from the data base, therefore, good quality relevant data is crucially important and the better the data, the better the model will be calibrated.

CHAPTER 3

AN INSTRUMENTED VEHICLE DATA BASE

3.1 INRODUCTION:

In order to investigate driver behaviour, a database of individual driving characteristics, which includes variables such as relative speed, relative distance, acceleration and speed need to be compiled. Such data is important as a basis for any credible microscopic simulation modelling in which calibration and validation are separated. This Chapter describe the TRG instrumented vehicle and how it was used to collect microscopic observing information. It also included the development of filtering and smoothing software to prepare the output for this study.

3.2 THE INSTRUMENTED VEHICLE:

The idea of the Instrumented Vehicle (IV) is not new, it goes back to 1958 [Chandler, Herman and Montroll]. Since that time, many researchers have used an IV's to obtain Data, because of the detail of individual vehicle's motion characteristics that can be collected. Moreover, the data can be used directly for other purposes such as Intelligent-Cruse-Control (ICC) algorithms and safety research.

The difficulties in constructing and testing an IV can be summarised in two main points:

1- Questions about the accuracy of the measurement and the flexibility of using the vehicle.
2- How to save and process this huge amount of data to make it accessible for research.

Nowadays, overcoming these restraints is much easier than before, because of the rapid improvement in computer power, the accuracy of the measuring instruments and the ability to save large quantities of data on a CD-ROM.

The current TRG IV was built during the summer of 1996. Figure 3.1 illustrates where the tools and instruments are fitted. The car is equipped with:

1- A MMW (Millimetre Wave) radar which can be fitted either in the front or the rear of the vehicle.
2- An optical Speedometer.
3- An Accelerometer.
4- Two Digital Cameras (one in front and another in the back)
5- Two Videos.
6- Two Microphones
7- A Personal Computer in order to save the Data from the other equipment.
8- Power Supply.

Each will be described in turn below:

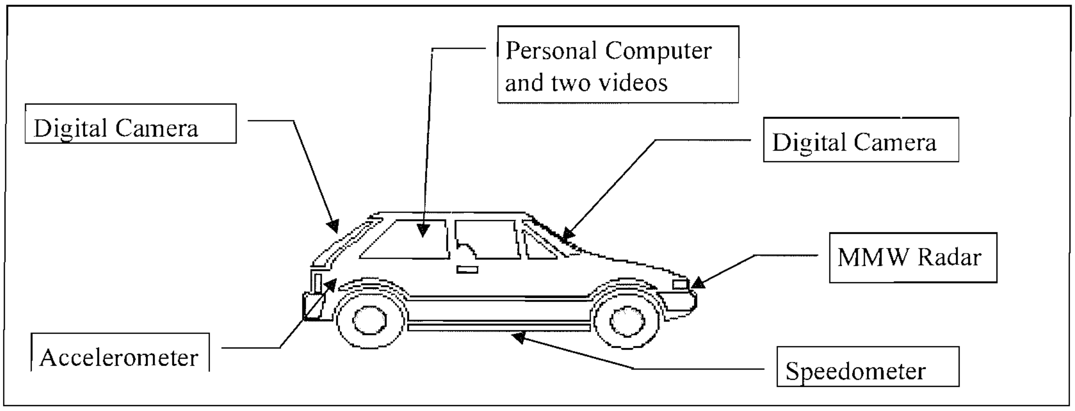


Figure 3.1 The TRG Instrumented Vehicle

3.2.1 The MMW Radar:

This measures the relative speed and distance to others vehicles. It is a research tool developed by *LUCAS Ltd.* The radar's range is about 150m with an angle of sight of around 8 degrees divided into three beams each one covering 3 degrees, Figure 3.2. The output from the radar consists of twelve readings (four from each beam) every 0.1 sec, each of which contains three values:

1- The relative distance of an object ahead of the radar (cm)

2- the relative speed of the same object (cm/sec)

3- The strength of the reflecting signal (integer value between 10 and 200, the higher the better, which is influenced by the relative distance and subject's material).

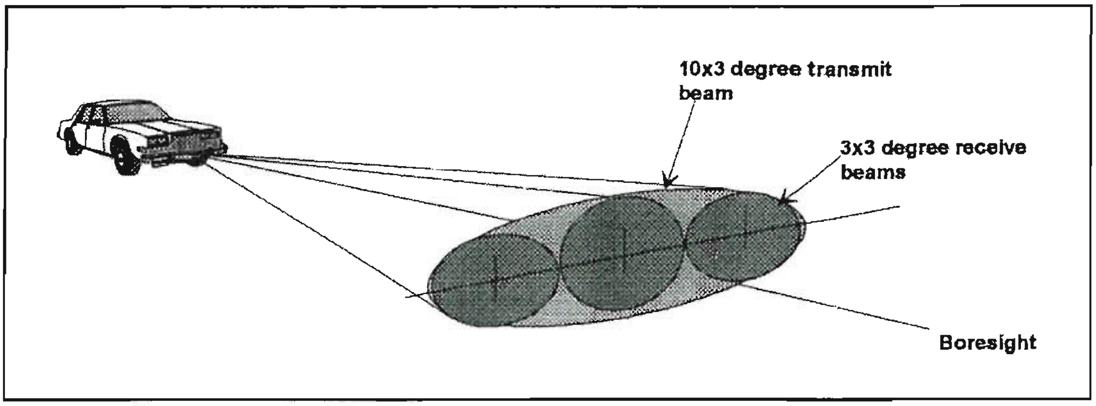


Figure 3.2: The MMW Radar and its beams.

Limitation :

1- The accuracy of the radar is about [± 20 cm per 100 m] for relative distance and [± 0.50 m/sec] for relative speed.

2- The shape and the size of the target affects the strength of the reflecting signal.

3- Because of the 8° angle of sight, the Radar cannot detect any vehicle in the neighbouring lane unless as far from the radar as Dx_a , where:

$Dx_a = 3.75 / (2 * \tan(4^\circ))$ ⇒ $Dx_a = 3.75 * 7.15$ ⇒ $Dx_a = 26.8m$

Additionally, the radar is blind to other vehicles in neighbouring lanes, if there is vehicle closer than a certain Distance Dx_b ,where : $Dx_b = W * 7.15$

where W is the width of the ahead vehicle Assuming W = 2.1 m ⇒ Dx_b = 15m). Figure (3.3) shows the angle sight and its limitations.

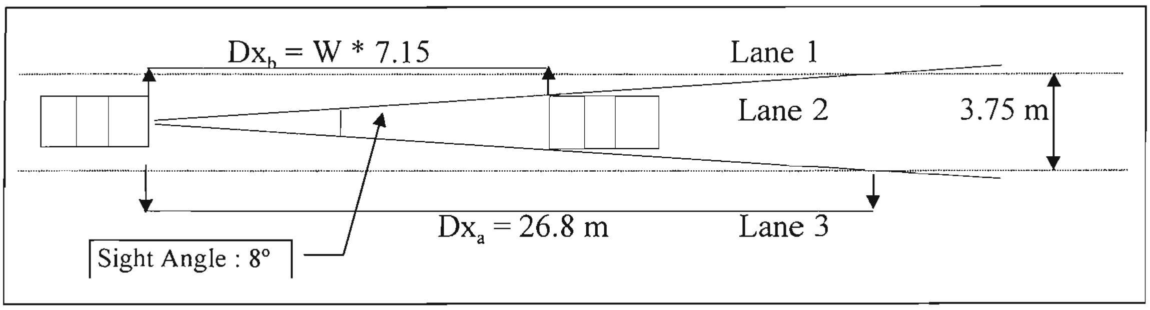


Figure (3.3): The Radar Limitations

3.2.2 The Speedometer:

This is an optical device that measures the ground speed. The manufacturer (DATRON Co.) climes that the speedometer accuracy is about (± 1 km/h), and it gives a reading every tenth of a second with a unit km/h*10. However, a calibration test by TRG showed the same result.

3.2.3 The Accelerometer:

This has been developed by the Transport Research Laboratory TRL to measure the longitudinal and lateral accelerations of a vehicle. This device is located above the centre of the rear axle and fixed on a horizontal surface. The output is a read every 0.1 sec with accuracy about ± 0.1 m/sec^2.

3.2.4 Two Digital Video Cameras:

These are important to capture the video images from the IV front and back. They are very small and hidden in a way other drivers will not be able to notice them. Also, they are digital which allow the IV computer to control them.

3.2.5 Two Videos:

These are important to record the video images from the front and rear video cameras. The videotape is standard SVHS for high quality images recording. The first video records the video images from either the front or the rear camera depending on the radar position (front or rear respectively). The other video records the two images from the video cameras simultaneously by using a video mixer connected to the two cameras.

3.2.6 Two Microphones:

These are useful to allow aural recording into the video data, which help to add remarks to the data in certain circumstances.

3.2.7 The PC Computer:

The computer was configured in the TRG Laboratory. It has a Cyrix [X686] processor, 32 MB RAM and ASCISI controller. Also, it has:

- A VIDEO BLASTER card for capturing video images.
- A SOUND BLASTER card for saving sound.
- A JAZ drive: which is a removable disk with a capacity of one GEGA-BYTE to give the flexibility to transfer the data files to another computer.
- A TFT crystal colour screen: which is useful to save room in the vehicle.

This PC controls all the instruments in the car, and saves the Data to the JAZ drive by running special software which was written using the VISUAL BASIC Professional programming language. The software saves every five minutes of data in a text file with a date and time header.

3.3 DATA REDUCTION:

The raw data collected by the instrumented vehicle has several defects:

1- Because the radar has three separate beams it may collect three reading at the same time from the same object.

2- When the radar does not receive a reflection, it sets a value of (-1) for the relative distance or a zero to the relative speed.

3- The Data has a high fluctuation, which needs smoothing in order to draw any conclusion. Figures 3.4, 3.5 and 3.6 show samples of the raw data from an output file.

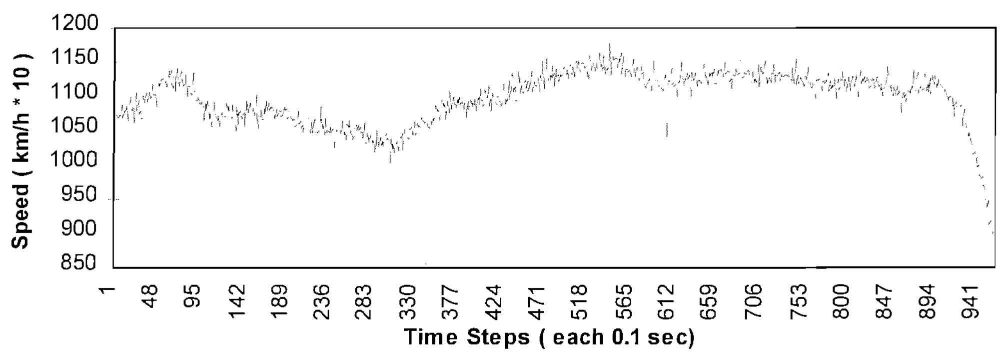


Figure 3.4: Sample of Speed from Speedometer [Raw Data]

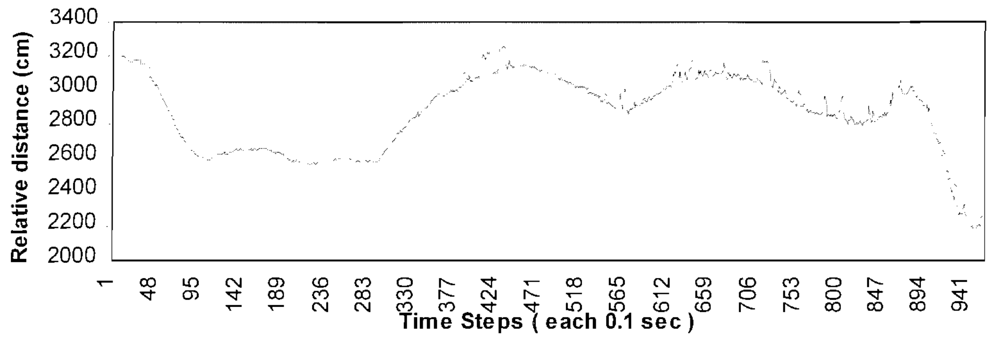


Figure 3.5: Sample of Relative distance from the Radar [Raw Data]

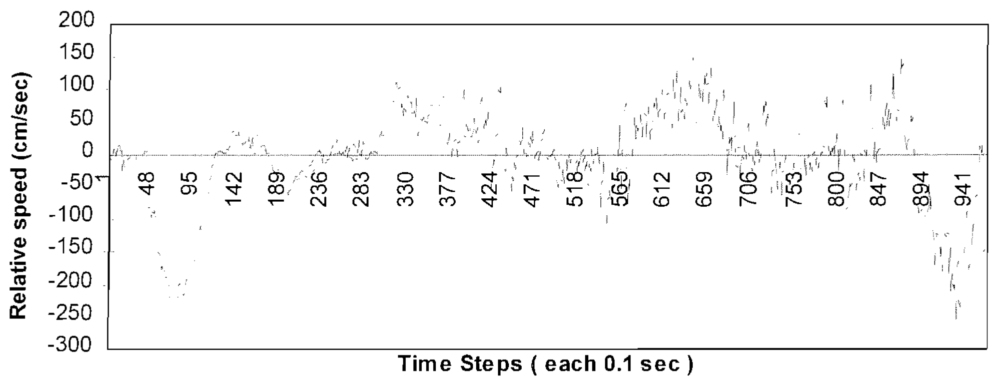


Figure 3.6: Sample of Relative Speed from the Radar [Raw Data]

4- The Radar does not sort the data from each object, so it is possible to find a mix of more than one object in each data stream depending on the strength of the different radar signal returns. Figure 3.7 shows a raw data file from the instrumented vehicle.

These data defects largely result from the small variation in vehicle positions resulting from the road surface, suspension vehicle interactions and the standard interpretation of the radar signal returns. Clearly, it is very important to build a program that can filter and smooth the data.

The same Object

False Data

PC_Time	R_1	V_1	R_2	V_2	R_3	V_3	R_4	V_4	R_5	V_5	R_6	V_6	R_7	V_7	R_8	V_8	R_9	V_9	R_10	V_10	R_11	V_11	R_12	V_12
14:26:25	6576	507	-1	0	-1	0	-1	0	2778	-49	3191	-111	6595	337	7171	-127	2781	-61	7195	-130	6555	186	-1	0
14:26:25	6605	507	-1	0	-1	0	-1	0	2769	-59	3178	-136	6622	336	7167	-105	2754	-92	7189	-130	6571	186	-1	0
14:26:26	6635	507	2733	0	-1	0	-1	0	2734	-86	3165	-146	6634	299	7164	-93	2785	-74	7181	-130	6581	186	-1	0
14:26:26	6652	507	2733	0	10745	0	-1	0	2726	-82	3143	-158	6664	310	7128	-129	2730	-130	7181	-130	6630	222	-1	0
14:26:26	6682	507	2733	0	10745	0	9874	0	2716	-91	3118	-170	7124	-124	6685	309	2716	-139	7181	-130	6643	222	5915	0
14:26:26	6699	507	2733	0	10745	0	9874	0	2704	-90	3131	-126	7121	-106	6709	306	2751	-80	7129	-169	6666	222	-1	0
14:26:26	6728	507	8708	[illegible]	[illegible]	[illegible]	9874	0	2695	-96	3136	-90	7107	-124	6747	319	2720	-114	7111	-178	6679	222	6082	0
14:26:26	6745	507	8708	[illegible]	[illegible]	[illegible]	[illegible]	0	2688	-91	3137	-81	7098	-108	6770	308	2709	-111	7077	-202	6661	0	-1	0
14:26:26	6774	507	8708	0	-1	0	-1	0	2657	-85	3123	-99	7089	-111	6788	317	2705	-95	7078	-165	2979	0	6661	0
14:26:26	6804	507	8384	0	-1	0	-1	0	2652	-76	3128	-74	7084	-106	6834	344	2694	-82	7053	-190	2979	0	2859	0
14:26:26	6838	507	8384	0	8088	0	7886	0	2681	-34	3125	-70	7041	-158	6850	326	2703	-58	7040	-179	-1	0	-1	0
14:26:26	6869	507	-1	0	8088	0	7886	0	2685	-33	3116	-67	6935	-195	6904	361	2675	-87	7040	-158	7596	0	-1	0
14:26:26	6900	507	-1	0	-1	0	7886	0	2694	-14	3113	-65	6901	326	6954	-177	2693	-47	7018	-167	7700	82	5642	0
14:26:27	6928	507	-1	0	-1	0	-1	0	2691	-17	3107	-70	6930	334	6951	-179	2677	-61	7035	-127	7705	82	5387	0
14:26:27	6939	498	2696	0	-1	0	-1	0	2684	-24	3121	-29	6941	333	6933	-165	2660	-76	7022	-117	7712	82	-1	0
14:26:27	6969	497	2696	0	15233	0	-1	0	2672	-55	3099	-55	6985	381	6977	-102	2654	-74	7022	-112	7716	82	8310	0
14:26:27	6986	497	15233	0	2696	0	14979	0	2673	-42	3102	-53	7011	382	6996	-59	2667	-51	7011	-113	-1	0	-1	0
14:26:27	7015	497	15233	0	14271	0	14979	0	2664	-55	3086	-87	7028	364	7022	6	2652	-76	7008	-100	-1	0	-1	0
14:26:27	7027	490	14271	0	15233	0	14979	0	2655	-62	3112	-37	7060	360	7053	48	2645	-78	6996	-92	-1	0	-1	0
14:26:27	7055	490	14271	0	2666	0	13208	0	2647	-49	3113	-18	7105	382	7100	80	2638	-72	6974	-100	-1	0	-1	0
14:26:27	7083	490	14271	0	2666	0	13208	0	2633	-52	3111	-26	7133	382	7114	85	2629	-70	6973	-95	7082	0	-1	0
14:26:27	7112	490	12406	0	12195	0	-1	0	2665	-13	3097	-55	7162	400	7119	85	2648	-29	6984	-61	-1	0	-1	0
14:26:27	7129	490	12406	0	12195	0	11964	0	2710	18	3091	-58	7125	85	7104	391	2699	29	6972	-66	3108	0	-1	0
14:26:28	7156	490	12406	0	12195	0	11964	0	2730	30	3078	-52	7207	350	2641	0	2717	38	6967	-67	2634	0	3108	0
14:26:28	12195	0	12406	0	-1	0	11964	0	2782	85	3049	-101	2641	0	7225	341	2711	26	6957	-77	2634	0	3108	0
14:26:28	-1	0	-1	0	-1	0	-1	0	2736	54	2661	26	3057	-69	7265	343	2655	-34	6947	-74	3108	0	-1	0
14:26:28	-1	0	-1	0	-1	0	-1	0	2745	76	2661	26	3082	1	7292	346	2613	-93	6940	-68	-1	0	-1	0
14:26:28	-1	0	-1	0	-1	0	-1	0	2738	56	2661	26	3068	-28	7318	346	2605	-85	6931	-74	2768	0	-1	0
14:26:28	-1	0	-1	0	-1	0	-1	0	2661	26	2763	92	3070	-30	7391	358	2615	-51	6923	-78	2768	0	5251	0
14:26:28	-1	0	-1	0	-1	0	-1	0	2635	-7	2784	105	3059	-52	7453	383	2689	19	6918	-78	3073	0	-1	0
14:26:28	-1	0	-1	0	-1	0	-1	0	2615	-35	2780	108	3070	-63	7511	403	2708	18	6912	-78	2652	0	3073	0
14:26:29	-1	0	-1	0	-1	0	-1	0	2612	-30	2785	98	3053	-81	7533	391	2680	7	6897	-95	2652	0	-1	0
14:26:29	-1	0	-1	0	-1	0	-1	0	2634	11	2791	103	3030	-69	7535	378	2625	-45	6935	-68	-1	0	-1	0
14:26:29	-1	0	-1	[illegible]	[illegible]	0	-1	0	2621	-11	2724	54	3045	-44	7554	353	2619	-40	6948	-48	-1	0	-1	0
14:26:29	-1	0	-1	[illegible]	[illegible]	0	-1	0	2619	-19	2633	-19	3024	-68	7595	353	2637	-20	6907	-69	-1	0	-1	0
14:26:29	-1	0	-1	0	-1	0	-1	0	2628	-21	2621	-23	3020	-75	7609	345	2603	-55	6882	-95	-1	0	-1	0
14:26:29	-1	0	-1	0	-1	0	-1	0	2618	-32	2611	-36	3020	-41	7611	302	2600	-55	6877	-87	-1	0	-1	0
14:26:29	-1	0	-1	0	-1	0	-1	0	2617	-32	2645	-6	3034	-30	7641	315	2618	-17	6877	-68	2937	0	7040	0
14:26:29	-1	0	-1	0	-1	0	-1	0	2616	-32	2666	25	3021	-49	7670	339	2661	35	6896	-39	2745	0	7040	0
14:26:29	-1	0	-1	0	-1	0	-1	0	2622	-5	3022	-41	7718	360	5259	0	2608	-27	6882	-41	2724	0	-1	0
14:26:29	-1	0	-1	0	-1	0	-1	0	2630	9	2997	-69	7745	349	5259	0	2607	-26	6879	-41	-1	0	-1	0
14:26:29	8665	0	-1	0	-1	0	-1	0	2609	-27	3031	-34	7793	380	6846	0	2609	-22	6862	-56	-1	0	-1	0

R_1 : Relative Distance [cm] - V_1 : Relative Speed

Figure 3.7: Sample of a raw data file before applying the filter.

3.3.1 Filtering Data:

The first step in filtering the data was to identify the rules that should be applied to the data to produce acceptable output. After a comprehensive study to the raw data, the following rules were developed for application to raw data files as a first filter to identify clear errors:

1- Any signal with strength less than 40Hz will be discarded: This is because of the unacceptable accuracy of the reading when the strength of the signal is less than 40Hz.

2- Any Relative Distance reading with (-1) will be discarded: This erases all the false readings from the raw data file.

3- The minimum number of a series of reading is five: The filter will erase any series lasting for less than 0.5sec, which is arguable.

4- The maximum difference between two successive relative-speed reading from the same object should not exceed (100 cm/sec) in (0.1) sec. This threshold will enable the filter to distinguish between different objects ahead of the radar, and it was suggested because of two accumulative factors, they are:

   a- The maximum expected relative deceleration with other vehicles ($0.5g = 5m/sec^2$): This factor can yield a relative speed difference of (50 cm/sec) in (0.1sec) time interval.

   b- The error margin of the radar itself as a value of (50 cm/sec) for the relative speed.

5- The maximum difference between two successive relative-distance reading from the same object (after adding the relative-speed's influence) should not exceed (150cm) in (0.1sec). This is because the radar sometime gives a relative distance reading which is in conflict with the relative-speed reading in the previous time step (ie. an increase in the relative-distance while the relative-speed has a negative value, see Figure 3.8).

The last two rules (4 and 5) are broad assumptions tested by applying the filter program on several files and the results were satisfactory. However, any error because of such rule can be recovered by using the viewing data program later in the process.

File Name	data line	Frame No	Time (sec)	Speed (km*10/h)	DX_6 cm	DV_6 cm/sec	P_6 Hz	DX_7 cm	DV_7 cm/sec	P_6 Hz
10141750	41	3151	304.349	1062	**5889**	**-99**	**52**	*9315*	*26*	*28*
10141750	42	3152	304.416	1088	**5887**	**-114**	**52**	*9366*	*82*	*36*
10141750	43	3153	304.51	1095	**5885**	**-95**	**52**	*9413*	*108*	*40*
10141750	44	3154	304.599	1068	**5732**	**-140**	**56**	*9420*	*108*	*40*
10141750	45	3155	304.687	1081	**5663**	**-179**	**56**	*9428*	*108*	*40*
10141750	46	3156	304.749	1049	**5447**	**-249**	**52**	*9072*	*193*	*48*
10141750	47	3157	304.839	1095	**5349**	**-262**	**52**	*9101*	*203*	*52*
10141750	48	3158	304.957	1091	*9764*	*255*	*56*	**5457**	**-223**	**52**
10141750	49	3159	305.047	1104	**5408**	**-223**	**56**	*9221*	*291*	*56*
10141750	50	3160	305.168	1057	*9266*	*301*	*60*	**5252**	**-272**	**56**
10141750	51	3161	305.291	1052	*9326*	*317*	*60*	**5113**	**-286**	**56**
10141750	52	3162	305.382	1069	*9386*	*325*	*64*	**5157**	**-299**	**56**
10141750	53	3163	305.476	1099	*9442*	*354*	*68*	**5163**	**-286**	**56**
10141750	54	3164	305.596	1025	*9486*	*392*	*80*	**5246**	**-273**	**56**
10141750	55	3165	305.69	1050	*9544*	*419*	*76*	**5152**	**-286**	**56**
10141750	56	3166	305.784	1065	*9578*	*434*	*72*	**5036**	**-299**	**56**
10141750	57	3167	305.88	1072	*9613*	*426*	*68*	**5010**	**-299**	**56**
10141750	58	3168	305.975	1055	*9674*	*476*	*72*	**5034**	**-274**	**56**

Figure 3.8: A raw data sample that shows the confusion in the relative-distance reading.

The Filter program will apply the previous rules in several stages. Each stage consists of several reading to the data file. The filter program has a sub-program, which enables the rules to be changed very easy. In order to filter several files at the same time, the program has been written in VISUAL BASIC computing language. See Appendix (1). Also, the program gives every separate series of data a number, which is very important to the VIEW DATA program. Figure 3.9 shows the output from the filter program applied to the same section of data as in Figure 3.7. It is clear that the data is now much more straightforward to analyse and understand

PC_Time	R_1	V_1	N1	R_2	V_2	N2	R_3	V_3	N3	R_4	V_4	N4	R_5	V_5	N5	R_6	V_6	N6	R_7	V_7	N7	R_8	V_8	N8	R_9	V_9	N9	R_10	V_10	N10	R_11	V_11	N11	R_12	V_12	N12
14:26:25	2781	-61	28	3191	-111	29	0	0	0	6576	507	16	7195	-130	30	6555	186	34	0	0	0	7171	-127	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:25	2754	-92	28	3178	-136	29	0	0	0	6605	507	16	7189	-130	30	6571	186	34	0	0	0	7167	-105	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:26	2785	-74	28	3165	-146	29	0	0	0	6635	507	16	7181	-130	30	6581	186	34	0	0	0	7164	-93	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:26	2730	-130	28	3143	-158	29	0	0	0	6652	507	16	7181	-130	30	6630	222	34	0	0	0	7128	-129	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:26	2716	-139	28	3118	-170	29	0	0	0	6682	507	16	7181	-130	30	6643	222	34	0	0	0	7124	-124	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:26	2751	-80	28	3131	-126	29	0	0	0	6699	507	16	7129	-169	30	6666	222	34	0	0	0	7121	-106	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:26	2720	-114	28	3136	-90	29	0	0	0	6728	507	16	7111	-178	30	6679	222	34	0	0	0	7107	-124	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:26	2709	-111	28	3137	-81	29	0	0	0	6745	507	16	7077	-202	30	0	0	0	0	0	0	7098	-108	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:26	2705	-95	28	3123	-99	29	0	0	0	6774	507	16	7078	-165	30	0	0	0	0	0	0	7089	-111	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:26	2694	-82	28	3128	-74	29	0	0	0	6804	507	16	7053	-190	30	0	0	0	0	0	0	7084	-106	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:26	2703	-58	28	3125	-70	29	0	0	0	6838	507	16	7040	-179	30	0	0	0	0	0	0	7041	-158	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:26	2675	-87	28	3116	-67	29	0	0	0	6869	507	16	7040	-158	30	0	0	0	0	0	0	6935	-195	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:26	2693	-47	28	3113	-65	29	0	0	0	6900	507	16	7018	-167	30	0	0	0	0	0	0	6954	-177	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:27	2677	-61	28	[illegible]	[illegible]	29	0	0	0	6928	507	16	7035	-127	30	0	0	0	0	0	0	6951	-179	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:27	2660	-76	28	[illegible]	[illegible]	[illegible]	0	0	0	6939	498	16	7022	-117	30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	6933	-165	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:27	2654	-74	28	[illegible]	[illegible]	29	0	0	0	6969	497	16	7022	-112	30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	6977	-102	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:27	2667	-51	28	3102	-53	29	0	0	0	6986	497	16	7011	-113	30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	6996	-59	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:27	2652	-76	28	3086	-87	29	0	0	0	7015	497	16	7008	-100	30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	7022	6	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:27	2645	-78	28	3112	-37	29	0	0	0	7027	490	16	6996	-92	30	0	0	0	0	0	0	7053	48	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:27	[illegible]	[illegible]	28	3113	-18	29	0	0	0	[illegible]	[illegible]	16	6974	-100	30	0	0	0	0	0	0	7100	80	33	0	0	0	0	0	0	0	0	0	0	0	0
14:26:27	[illegible]	[illegible]	28	3111	-26	29	0	0	0	[illegible]	[illegible]	[illegible]	6973	-95	30	0	0	0	0	0	0	7114	85	33	0	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0
14:26:27	2648	-29	28	3097	-55	29	0	0	0	7112	490	16	6984	-61	30	0	0	0	0	0	0	7119	85	33	0	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0
14:26:27	2699	29	28	3091	-58	29	0	0	0	7129	490	16	6972	-66	30	0	0	0	0	0	0	7125	85	33	0	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0
14:26:28	2717	38	28	3078	-52	29	0	0	0	7156	490	16	6967	-67	30	0	0	0	0	0	0	0	0	0	0	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0
14:26:28	2711	26	28	3049	-101	29	0	0	0	7225	341	16	6957	-77	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:28	2655	-34	28	3057	-69	29	0	0	0	7265	343	16	6947	-74	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:28	2613	-93	28	3082	1	29	0	0	0	7292	346	16	[illegible]	[illegible]	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:28	2605	-85	28	3068	-28	29	0	0	0	7318	346	16	[illegible]	[illegible]	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:28	2615	-51	28	3070	-30	29	0	0	0	7391	358	16	[illegible]	[illegible]	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:28	2689	19	28	3059	-52	29	0	0	0	7453	383	16	6918	-78	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:28	2708	18	28	3070	-63	29	0	0	0	7511	403	16	6912	-78	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:29	2680	7	28	3053	-81	29	0	0	0	7533	391	16	6897	-95	30	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:29	2625	-45	28	3030	-69	29	0	0	0	7535	378	16	6935	-68	30	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:29	2619	-40	28	3045	-44	29	0	0	0	7554	353	16	6948	-48	30	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:29	2637	-20	28	3024	-68	29	0	0	0	7595	353	16	6907	-69	30	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:29	2603	-55	28	3020	-75	29	0	0	0	7609	345	16	6882	-95	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:29	2600	-55	28	3020	-41	29	0	0	0	7611	302	16	6877	-87	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:29	2618	-17	28	3034	-30	29	0	0	0	7641	315	16	6877	-68	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:29	2661	35	28	3021	-49	29	0	0	0	7670	339	16	6896	-39	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:29	2608	-27	28	3022	-41	29	0	0	0	7718	360	16	6882	-41	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:29	2607	-26	28	2997	-69	29	0	0	0	7745	349	16	6879	-41	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14:26:29	2609	-22	28	3031	-34	29	0	0	0	7793	380	16	6862	-56	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Vehicle No 29

Vehicle No 28

Vehicle No 16

Vehicle No 30

Blank Areas NO Objects

Blank Areas NO Objects

Blank Areas NO Objects

R_1 : The relative distance with target number N_1[cm] - V_1 : The relative Speed with target number N1[cm/sec]

Figure 3.9: A sample of the output from the filter program

3.3.2 Smoothing The Data:

Within the context of the small time units of movement, the data points exhibit fluctuations which are functions of the data collection process rather than inherent description of the underlying following events. It is very important to find a method to smooth the data to better represent the true situation. The data without smoothing is shown in figure 3.10 from which the difficulties of interpretation may be seen. The smoothing data is not a new concept, but how to find the suitable method is an important question.

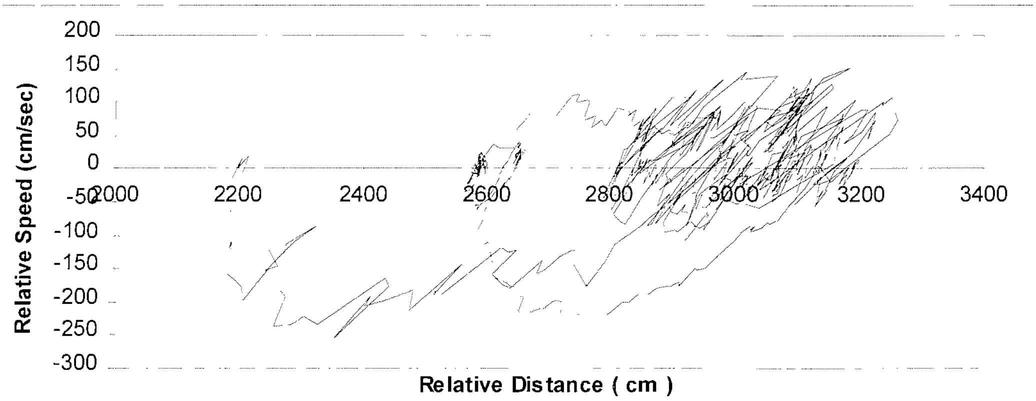


Figure 3.10: The relative speed & distance diagram, a sample of the unsmoothed data

There are three methods that can be used:

3.3.2.1 Moving Average Method:

This method assumes that the estimated value is the average value of the variable over a specific number of preceding periods. This method can be criticised in two points:

1- The estimated value is only influenced by the preceding value, so it is dealing with forecasting more than smoothing.

2- The amount of influence is uniform over the chosen period, but as the reading becomes farther from the estimating point the influence should be less.

3.3.2.2 Exponential Smoothing Method:

This method predicts a value based on the forecast for the prior period, adjusted for the error in that prior forecast. It uses a smoothing constant [a], the magnitude of which determines how strongly forecasts respond to errors in the prior forecast. This method is good but has the same defects as for smoothing. It estimates the value according to the previous values.

3.3.2.3 The Triangular Weighted Moving Average Method (TWMA):

This method would give the greatest weight to the point being estimated and zero weight to the two end points. So, it is taking into account the values before and after the estimated point and assessing the influence according to how close the value to the estimated one. An average over 15 readings before and 15 readings after was found to be adequate for smoothing the data after applying different values of the number readings that should be assessed. However, before applying the TWMA, the original series would first have to have "correction" applied to each end. We can summarise the TWMA procedure in three steps:

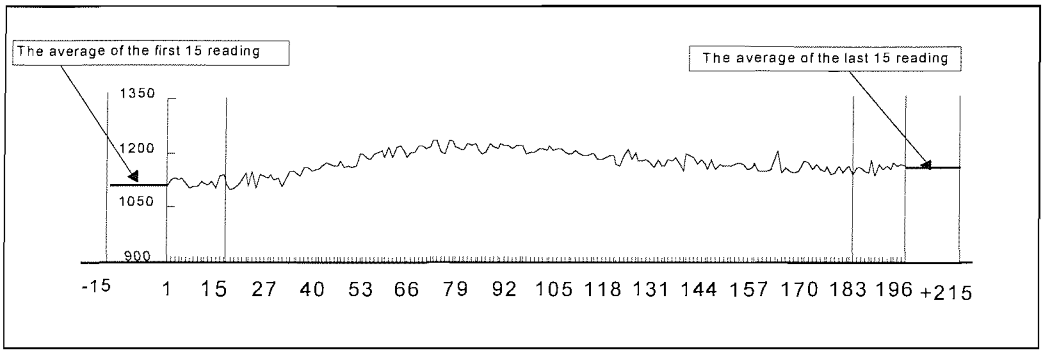


Figure 3.11: Adding the average of the first and last 15 reading to the data series.

1- Obtain the average over the first 15 readings of the series (X_S)(suppose the number of reading in the series is N > 30 readings). Determine the average over the last 15 readings (X_L).

2- The average (X_S) is added as a constant value to the left end of the series, while the average X_L is added as a constant value to the right end of the series. The new series has (N + 30) readings, Figure 3.11.

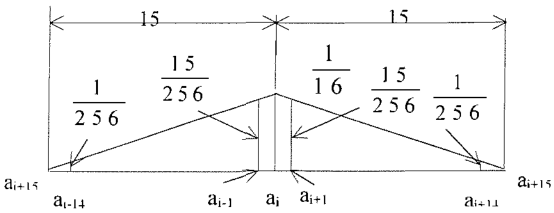
3- Apply on the new series the Triangular Weighting Function, which is :

$$\hat{a}_i = \frac{1}{16}\left(\frac{1}{16}a_{i-15} + \ldots\frac{15}{16}a_{i-1} + a_i + \frac{15}{16}a_{i+1} + \ldots + \frac{1}{16}a_{i+15}\right)$$

where is :

$\hat{a}_i$: Is the estimated value.

a_i : Is the original value.



Weighting Function

The output from the Triangular Weighting Function is the smoothed data series. Figures 3.12, 3.13, 3.14 and 3.15 show how much the data has been improved by this method. It is clear that the filter program is important to any research using the data from the IV

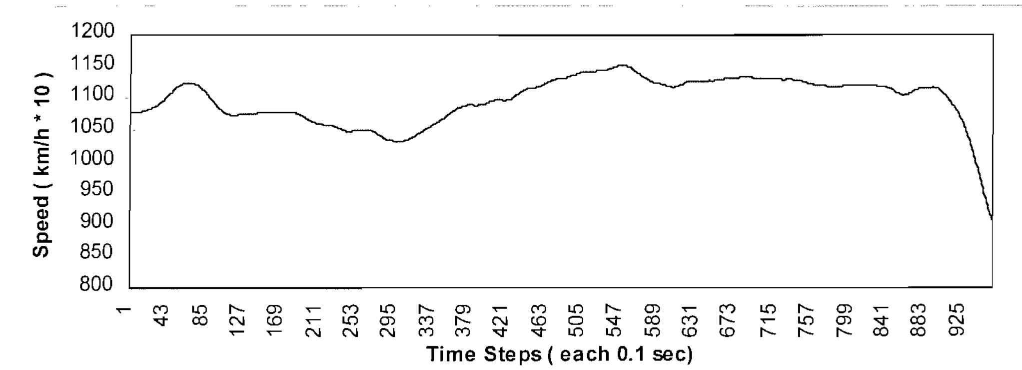


Figure 3.12: The smoothed ground Speed

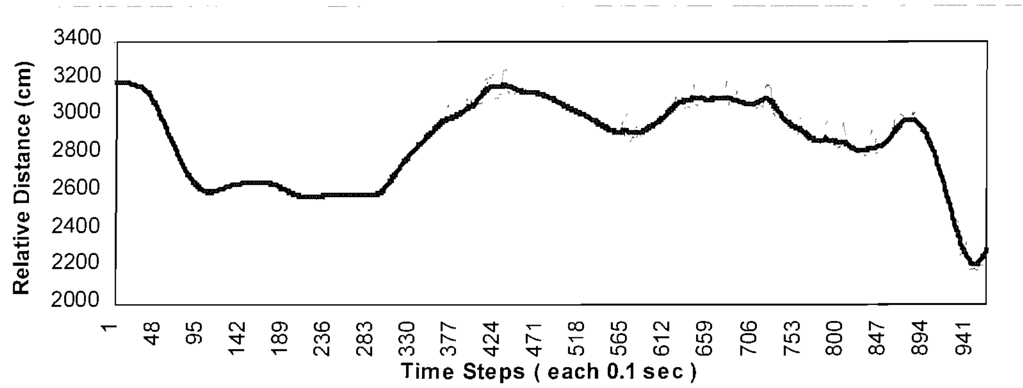


Figure 3.13: The raw and smoothed Relative distance

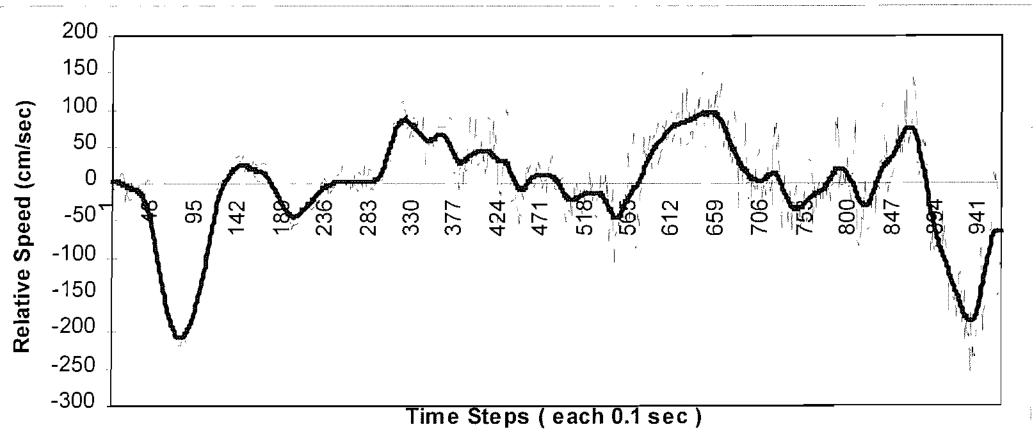


Figure 3.14: The raw and smoothed Relative Speed

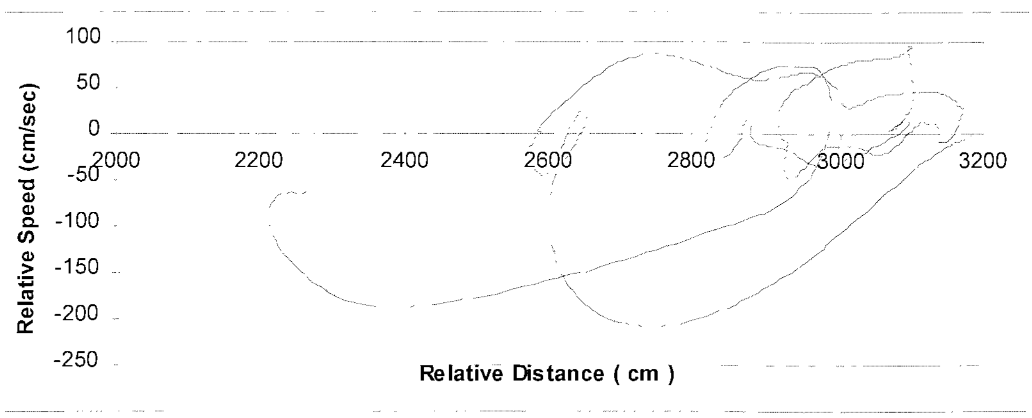


Figure 3.15: The Relative Distance & The Relative Speed after Smoothing

3.4 DATA VIEWING PROGRAME:

A computer program has been written to act as an inter-active environment between the user, data and the video screen. The advantage of this program is that the user can be able to look at the video screen and observe the data at the same time. This shortens the time and makes the data accessible. The program also has the ability to isolate any specific series of data and save it in a separate file such as events and average value readings. The program was written by BRAIN MOULD (TRG technician) in VISUAL BASIC professional programming language under the Windows Operation System.

3.5 SUMMARY:

The procedure to transfer raw data into an accessible and ready to analyse has the following steps (Figure 3.16):

1- Filtering and smoothing the raw data files by using the Filter Program.

2- Viewing the filter data combined with the video recording by using the data-viewing program.

3- Saving the data series that related to the experiment (ie.: the data series that represent the vehicle following the IV [radar in the rear]) in the TDF files format (a text file that can be opened by MS Excel Spread Sheet program).

4- By using MS Excel (a spread sheet software) the data series are matched to produce one data series for every subject (i.e. the driver of the following vehicle).

The final outcome will be a set of data files which can be opened by a spread sheet program and in which each file represents one subject (a driver) who followed the IV for a period of time. The data file has the measurement every 0.1 of a sec for these parameters:

1- Time in seconds with accuracy of one thousand of a second.

2- Speed of IV in km/h.

3- Acceleration of IV in m/s^2.

4- Relative distance in meter between the IV and the following vehicle [radar in the rear].

5- Relative speed in meter per second between the IV and the following vehicle [radar in the rear]. The relative speed would have a negative singe if the situation was closing and positive if it is opening.

By this stage the data is ready to be analysed. However, other subroutines were written in order to isolate specific events or action points from the data series. These will be described as part of the analyses for every experiment in the project.

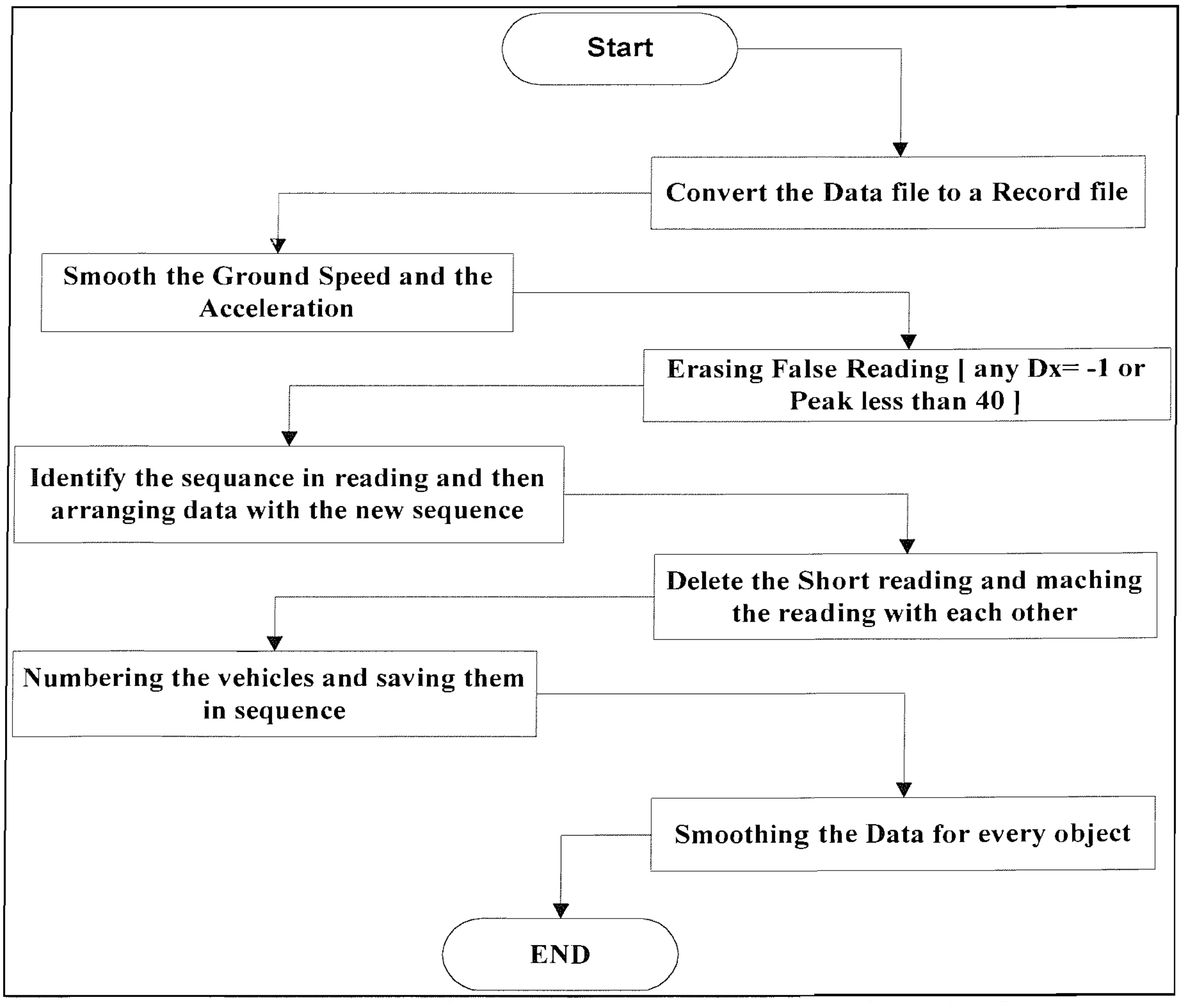


Figure 3.16: The flowchart of the main stages in the Filter & Smoother Program.

CHAPTER 4

THE PROGRAMMING STRUCTURE OF THE SIMULATION MODEL

4.1 INTRODUCTION:

This section describes the system structure of the simulation program, the type of user interface and the vehicle list processing used. The careful organised framework of interrelated subroutines enables high-speed performance, with good memory management, and a user-friendly interface.

4.2 THE SYSTEM STRUCTURE:

The model consists of eight major subroutines, each with a specific main function. These subroutines are:

4.2.1 Input Data Subroutine

In this subroutine the model reads the initialised values such as level of flow, speed parameters, vehicles' types...etc. There are two ways of entering data:

a- Direct Data Input: This method is used for the inter-active environment between the user and the computer. It usually takes a considerable time to operate, as the computer will requires every parameter to be entered individually. However, it can be useful for unfamiliar users to appreciate the requirements and variabilities and is simpler in use. Otherwise, the indirect method is better.

b- Indirect Data Input: This method is used when the user is familiar with the program and has the experience in dealing with data and the way in which the program accepts it. It is more complicated but much quicker.

4.2.2 Vehicle Generation Subroutine:

This subroutine is used to generate vehicles according to the distributions of headways, speeds and vehicle types. It gives every vehicle a number and identifies the vehicle and driver characteristics.

4.2.3 Car Following Subroutine:

This determines acceleration level for the next time increment for the vehicles inside the simulator. It is the core of the simulation program and plays the main part in the whole project. The logic was developed from the results of the experiments described later in the thesis. Three type of car following can be distinguished

1- Free Driving: Where the driver is not influenced by a lead vehicle.

2- Approaching slower vehicle: where the driver controls the approaching process.

3- Following a close vehicle: Car following behaviour.

4.2.4 Lane Changing Subroutine:

This subroutine is concerned with the driver's lane changing decision. The model depends on gap acceptance and the relative speeds with other vehicles. Logically, there are two main questions that should be answered before a driver changes lane, they are:

1- Is there a need to change lane.

2- Is there an ability to change lane.

Until, the answer is 'yes' from these two questions, the driver will not undertake the manoeuvre. Development of the lane-changing model will be discussed later in the thesis.

4.2.5 Saving Data Subroutine:

This subroutine is essential in order to analyse the output from the simulation. The format and the type of the output data are varied between experiments, which means that the subroutine will be modified for every experiment. Mainly, the project is concerned with the flow rate, average speed and the density over a section of three lanes motorway. It is wise to have text format output file that enable the user to undertake the analyses in a spread sheet program such as Excel.

4.2.6 Update Vehicle Position & Speed:

This subroutine calculates the position and the speed for every vehicle inside the model. The calculation is made according to the assumption that every driver will have a constant deceleration or acceleration over the time increment.

4.2.7 Graphics Interface Subroutine:

This views vehicle movements on the computer screen. Because of the graphics the simulation speed will drop. Therefore, an option was developed to enable the user to control the graphics viewing as either ON or OFF.

4.2.8 Erasing Vehicle Subroutine:

This subroutine clears the memory reserved for the vehicles which passed the end line of the simulated section. The subroutine is very important in managing the memory of the computer in order to enable the program runs to be as long as possible.

Other subroutines, which contribute to the model, include generating random numbers with a specific distribution. Such subroutines are described later. The flowchart of the Simulation Model is shown in Figure 4.1.

4.3 THE USER INTERFACE:

In recent years, the Microsoft Windows Operation System has become the 'de facto' standard for Personal computers. This is largely because of its user-friendly interface (i.e. multi-program operation and visual aid interface). It was decided that the simulation program should be developed under the Windows environment. The DELPHY programming developer was used to build the simulation program. It has many advantages such as:

1- It uses Turbo Pascal code, which is relatively easy to understand.

2- It uses the drag and drop method to generate the program interface.

3- It generates one executive file for the program, which makes installing the simulation simple.

4- It is able to access the dynamic memory of the computer.

There are many other detailed facilities that make DELPHY the right choice for Developing the simulation program.

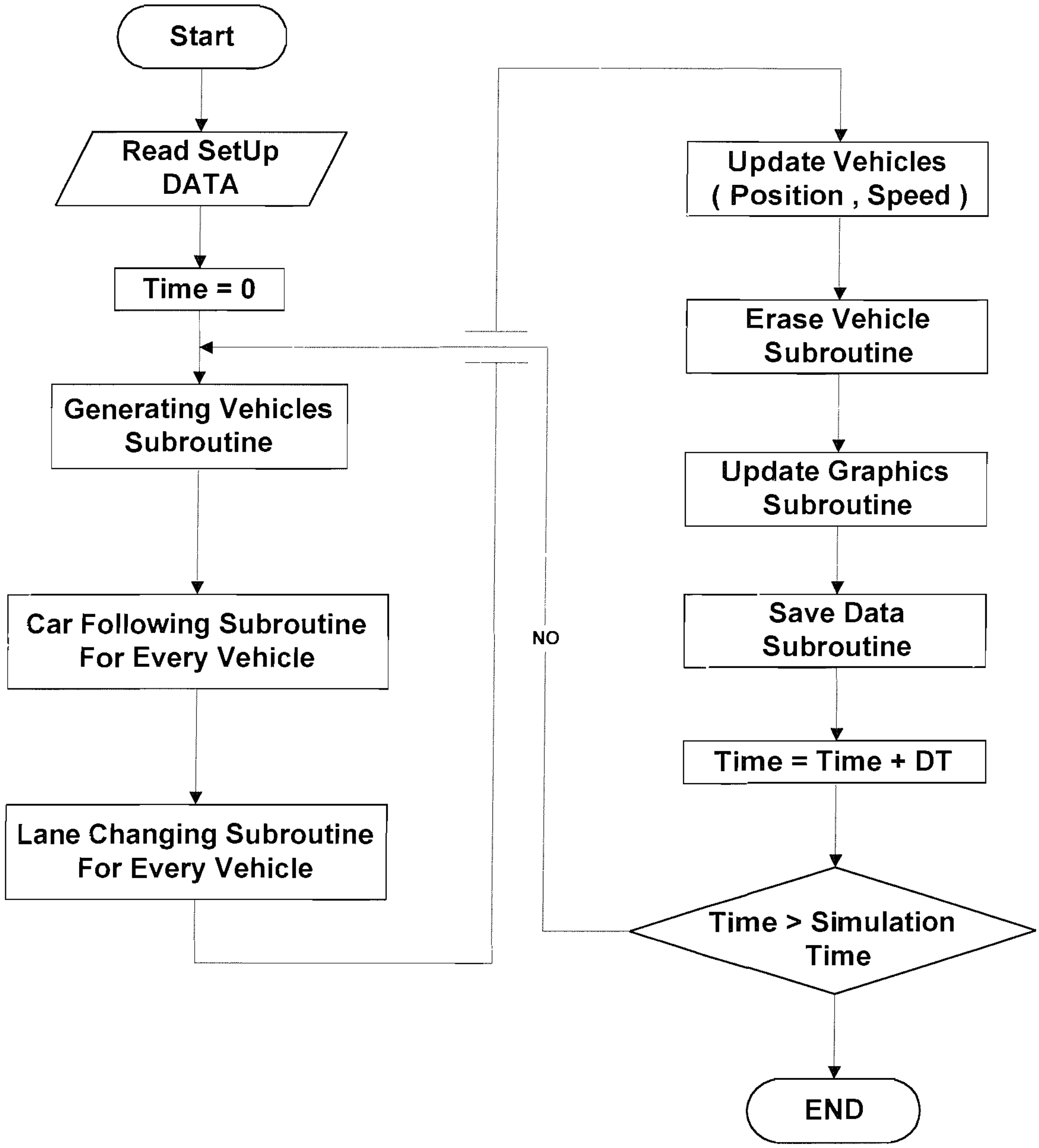


Figure 4.1: The Programming structure of the simulation model

4.4 VEHICLE LIST PROCESSING:

The objective of the vehicle list processing is to organise and manage the computer memory so that the model is efficient and has the ability to deal with a large number of vehicles. A specific number is given to each vehicle when it enters the system. This identifies the vehicle with a record which consists of all the variables that specify the characteristics of the vehicle and its driver and the addresses of the adjacent vehicles. Figure 4.2 shows the adjacent vehicles that may influence driver behaviour.

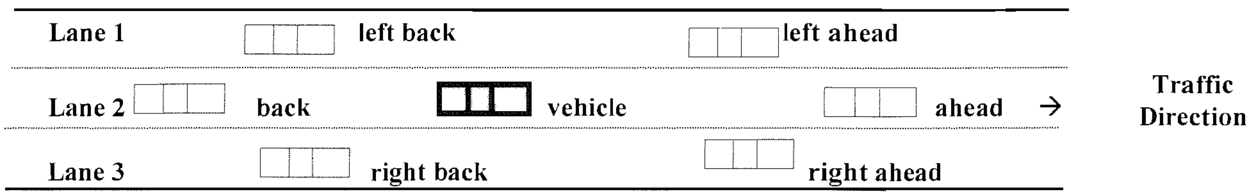


Figure 4.2 the adjacent vehicles that interact with the assessed vehicle.

Because a large number of vehicles need to be simulated, the computer memory accessible by the programming language should also be large. There are two types of memory that are used by programming language Static Memory and Dynamic Memory. Dynamic memory is much larger than Static memory and depends on the amount of the computer's memory itself, while Static memory is restricted to 64 KB of memory. DELPHY is able to access the dynamic memory by using variables called pointers, which can save the address of any record of memory. Thus, each vehicle's record will have pointers to adjacent vehicles, enabling the necessary links between the vehicles.

CHAPTER 5

VEHICLE CHARACTERISTICS AND ARRIVAL

5.1 INTRODUCTION:

The variables that replicate driver and vehicle characteristics are very important in microscopic models which deal with every vehicle. Also, it is necessary to release new vehicles at the entry point of the simulation with appropriately distributed headways so as to achieve a good flow representation in a reasonable time. This Chapter is concerned with vehicle characteristics and vehicle generation.

5.2 VEHICLE CHARACTERISTIC VARIABLES:

These variables relate to both drivers and vehicles.

5.2.1 Driver Variables:

These specify driver characteristics. The complicated inter-relationship between all the characteristics, which contribute to driver behaviour, makes calibrating every individual parameter very difficult. In this model, driver variables were determined after investigating driver behaviour in the car following and lane changing process.

5.2.2 Vehicle Variables:

These parameters are related to the vehicle, independently of driver behaviour. There are two types of vehicle parameters:

a- <u>Constant Parameter</u>: which are:

1- Vehicle identification number.

2- Vehicle type: Passenger car, HGV, Coach and Van.

3- Vehicle's length: This varies due to the vehicle type and to the trends in motor industry. In England VAN AS has done field measurements and found that the normal distribution can describe the variation of vehicle length, see Table 5.1 (VAN AS 1979)[31].

Vehicle Type	Mean (m)	Std (m)	Minimum (m)	Maximum (m)
Car	3.74	0.51	2.40	5.00
Van	3.95	0.53	2.80	5.20
Light Lorry	7.62	1.92	4.30	12.00

Table 5.1: Measured vehicle length (VAN AS 1979)[31]

Also, by using a data from M4 motorway, BRANSTON estimated the average length for each vehicle type, see Table 5.2 (BRANSTON D. 1977)[85]. Later Skabardonis has suggested a normal distribution with moderate parameters between the two previous studies for his simulation model, see Table 5.3 (SKABARDONIS A. 1983)[35]. As the vehicle length distributions reported by SKABARDONIS address the same mixed traffic field area as the field area of this study they are used in the proposed model.

Vehicle Type	Near-side Lane	Off-side Lane
Car	3.90 m	4.00 m
Van	4.30 m	4.20 m
Light Lorry	7.20 m	6.80 m
HGV	12.30 m	9.10 m

Table 5.2: Measured vehicle length average (BRANSTON 1977)[85]

Vehicle Type	Mean (m)	Std (m)
Car	4.00	0.60
HGV	11.00	2.40

Table (5.3): Vehicle length parameters for the model (SKABARDONIS A. 1982)[35]

4- Vehicle's width: This, also, depends on the vehicle type, however, it is less important than the vehicle length. HOQUE has used normal distribution to present this parameter in his simulation study, see Table 5.4 (HOQUE. M.S. 1994)[86]. These values will be adopted for the proposed model.

Vehicle type	Mean (m)	Std. Dev. (m)	Min. (m)	Max. (m)
Car	1.88	0.15	1.46	2.05
HGV	2.41	0.15	2.22	2.68

Table 5.4: Vehicle width distribution for the model (HOQUE. M.S. 1994)[86]

b- Changeable Parameters:

1- The Position of vehicle at the previous three epochs.

2- The Speed of vehicle at the previous three epochs.

3- The present lane number.

4- Journey time.

5- The Maximum Acceleration: Depends on the type of vehicle and the currant speed.

6- The Maximum Deceleration: Depends on the type of vehicle and the currant speed.

7- The front vehicle number in the same lane.

8- The rear vehicle number of in the same lane.

9- The front vehicle number in the left lane.

10- The front vehicle number in the right lane.

11- The rear vehicle number of in the left lane.

12- The rear vehicle number of in the right lane.

Other variables used for programming purposes only have not been mentioned.

5.3. TIME HEADWAY DISTRIBUTIONS:

Vehicle generation deals with the timing and introduction of each individual vehicle into the model. Therefore, determining the proper headway distribution is the best way to describe the vehicle arrival process in a microscopic simulation model (CHIN. H. C 1983)[32]. The assumption of random arrivals may be adequate when traffic flows are low, but it is less effective when traffic flows build up toward a congested situation.

There are two factors which characterise the headway distribution in traffic flows:

a- Vehicles have finite lengths and each headway therefore contains a real minimum time in which a vehicle must travel its own length.

b- A lack of passing opportunities causes queuing or bunching of vehicles. Successive headways are no longer independent and the random hypothesis breaks down.

Thus, two conditions of traffic stream should be considered in generating or measuring headway distribution: vehicles in following processes (constrained) and vehicles in non-following processes (free). Non-following vehicles are those which are unimpeded by preceding vehicles, while following vehicles are those which are prevented by the vehicles ahead from reaching the desired speeds. The specific description of 'following' is detailed later.

5.3.1 Random Headways (Non-following Vehicles):

A series of events is defined as random when (Adams 1936)[33]:

- **a-** Each event is independent of any other event.
- **b-** Equal intervals of time are equally likely to contain equal numbers of events.

Under these conditions, traffic is distributed according to the Negative Exponential Distribution (ASHTON W. D. 1966)[34]:

$$P(H_t < t) = 1 - e^{-Qt} \qquad \text{....5.1}$$

Where $P(H_t < t)$ is the probability of a headway H_t (sec) being less than t , in a stream of traffic with average flow rate of Q (veh./sec). The probability density function of the exponential distribution is :

$$f(t) = P' (H_t < t) = Q \cdot e^{-Q.t} \qquad \text{....5.2}$$

The average headway T and standard deviation σ would be :

$$T = \sigma = 1/Q$$

5.3.2 Non-Random Headways (following Vehicles):

Generally, non-random headway distributions are classified into single distribution models and mixed distribution models.

5.3.2.1 Single Headway Distributions:

a- Shifted Negative Exponential Distribution

Because vehicle length cannot be ignored, the negative exponential distribution can be corrected by shifting the distribution away from the origin by a quantity (τ), which is the minimum headway:

$$f(t) = \frac{1}{T - \tau} * e^{-(t - \tau) / (T - \tau)} \qquad t > \tau \qquad \text{....5.3.a}$$

$$f(t) = 0 \qquad t \leq \tau \qquad \text{....5.3.b}$$

Where T is the average headway (1/Q). The standard deviation would be $\sigma = T - \tau$.

This distribution is very easy to apply in any programming language. Unfortunately, observations have found that this distribution had given a close agreement to motorway headway data only for lane volumes up to 750 veh/h (SKABARDONIS A. 1982)[35].

b- Pearson Type III Distribution

TOLLE reported that this distribution is amongst the possible formulations for headway probabilities (TOLLE 1976)[36]. Its density function is:

$$f(t) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} * (t-\tau)^{\alpha-1} * e^{-\beta.(t-\tau)} \qquad \text{....5.4}$$

Where: (α, β, τ) are parameters of the distribution and $\Gamma(\alpha)$ the gamma function. The distribution can be simplified by making the parameter (τ) zero and the result would be the gamma distribution:

$$f(t) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} * t^{\alpha-1} * e^{-\beta.t} \qquad \text{....5.5}$$

The gamma distribution was simplified by assuming α is integer. This led to the ERLANG distribution (CLEVELAND 1964)[37]:

$$f(t) = \frac{\beta^{\alpha}}{(\alpha-1)!} * t^{\alpha-1} * e^{-\beta.t} \qquad \text{....5.6}$$

The average and the standard deviation of the Erlang distribution is given by:

$$T = \alpha / \beta \qquad \sigma = \sqrt{\alpha} / \beta$$

As can be seen, the negative exponential and shifted negative exponential distributions are particular cases of the ERLANG distribution where $\alpha = 1$

This distribution has been shown to give a satisfactory fit to empirical data especially at large and intermediate headway values but a poor fit at short headways. (BUCKLEY D.J. 1962)[38].

c- Log-normal Distribution

If the logarithm of a variant follows the normal distribution, the distribution would be the log-normal distribution, with a probability function of:

$$f(t) = \frac{1}{\sigma * t * \sqrt{2 * \pi}} * e^{-(Ln(t) - \mu)^2 / 2\sigma^2} \qquad ....5.7$$

Where μ and σ are the average and standard deviation of the distribution. The mean headway T and standard deviation S in terms of natural values are given by:

$$T = e^{\mu + 0.5\sigma^2} \qquad ....5.8$$

$$S^2 = T^2 * (e^{\sigma^2} - 1) \qquad ....5.9$$

TOLLE found that the log-normal distribution provides a good fit at high traffic flows (TOLLE 1976)[36].

5.3.2.2 Mixed Headway Distributions:

Mixed headway distributions attempt to model a distribution f(t) of all headways in terms of the distribution of following headways g(t) and the distribution of non-following headways h(t). The common density function is:

$$f(t) = \Phi * g(t) + (1 - \Phi) * h(t) \qquad ....5.10$$

Φ is the proportion of restrained vehicles (ie. these with following headway). Because these relationships consist of more than one distribution, they will be described as models to distinguish between them from the single distribution models. Depending on the way the two distributions are specified, there could be two types of mixed Models which are described below:

a- Composite Models: there are two examples of these distributions:

(i) Double Exponential Model: This consists of two exponential distributions, one for the following and another for the non-following situations (SCHULL 1955)[39]. The probability function would be:

$$P(H \geq t) = \phi * e^{-(t-\tau)/(T_1-\tau)} + (1-\phi) * e^{-t/T_2} \qquad ...5.11$$

Where :

T_1 : mean headway between restrained vehicles.
T_2 : mean headway between unrestrained vehicles.
τ : minimum headway.

It was found that this distribution is acceptable for low and medium flows but does not seem to give a good fit at high flow rates.

(ii) Hyperlang Model: This was proposed by DAWSON and CHIMINI, and it is a linear combination of ERLANG and negative exponential distributions (DAWSON & CHIMINI 1968)[40]. The probability function for it is:

$$P(H \geq t) = \phi * e^{-(t-\tau_1)/(T_1-\tau_1)} + (1-\phi) * e^{-K*\frac{t-\tau_2}{T_2-\tau_2}} \sum_{X=0}^{K-1} \left(K \frac{t-\tau_2}{T_2-\tau_2} \right) * \frac{1}{X!} \qquad ....5.12$$

Where :

T_1 , τ_1 : mean and minimum headway for free vehicles.
T_2 , τ_2 : mean and minimum headway for restrained vehicles.
K : an index denoting the degree of non-randomness in the distribution of restrained headways

As can be seen, it is a general model, and while it was reported that the results from this model are very good, a disadvantage is the number of parameters in the distribution.

b- Travelling Queue Models:

MILLER proposed that traffic stream forms a process comprised of random bunches and gaps (MILLER 1961)[41]. The change of the random arrival process by the moving queue technique is similar to that imposed by the classical queuing system with a single server. The distribution of following headways on a motorway resembles the service time distribution in the queuing system. Each non-following headway t is the sum of a following headway x drawn from g(x) and a gap (t - x) which is exponentially distributed.

$$h(t) = \int_0^t g(x) \lambda e^{-\lambda(t-x)} dx \qquad ....5.13$$

The normal, gamma and log-normal distributions were proposed to represent the g(t) distribution. It was found that the log-normal distribution gave the best fit for a wide range of flows (BRANSTON 1976)[42].

5.3.3 Conclusion:

Although the shifted negative exponential distribution is very easy to use, it is not suitable for the distribution of headways at higher rates of traffic flow. The generalised queuing model with log-normal distribution of following headways g(x) seems to be the best model to replicate the headway distribution on motorways for simulation task, ie a good model that can be generated by simple computing process. The mean and standard deviation of the distribution g(x) assumed to remain constant over the flow range with a value (BRANSTON 1976)[42]:

m = 1.6 sec   Std =0.40   for the slow lane

m = 1.3 sec   Std =0.40   for the fast lane

Also, WASIELEWSKI found that the mean and the standard deviation of the g(x) distribution have the values 1.32 sec and 0.52 sec respectively, and they are independent of the flow rate (WASIELEWSKI 1979)[43]. SKABARDONIS (SKABARDONIS A. 1982)[35] suggested the mean inter-bunch gap 1/λ and the proportion ϕ of the following vehicles should be calculated as:

$$\lambda = Q - 0.5 \cdot Q^{1.5} \qquad \text{.... 5.14}$$

$$\phi = \rho - 0.5(\rho - 1) \cdot Q^{0.5} \qquad \text{.... 5.15}$$

where :

Q : flow rate.
ρ : traffic intensity $\rho = m \cdot Q$

5.4 GENERATING VEHICLES IN THE MODEL:

In the simulation model a following headway TF is randomly generated according to the log-normal distribution with the parameters :

mean = 1.6 sec   Std =0.40   for the off-side lane

mean = 1.3 sec Std =0.40 for the middle and near-side lane

The parameters [λ and ϕ] are calculated from the rate of flow [Q] (equations 5.14 and 5.15), and a uniform random number R is generated and compared with ϕ (equation 5.15), if ($R \leq \phi$) the next arrival would be a follower, otherwise a random number with a negative exponential distribution is generated TE [the distribution parameter would be λ]. The latter (TE) will be added to the following headway (TF) to get a non-following headway. The flow-chart for the vehicle arrival subroutine is shown in Figure 5.1. The following sections explain how the random numbers were generated with desired distributions.

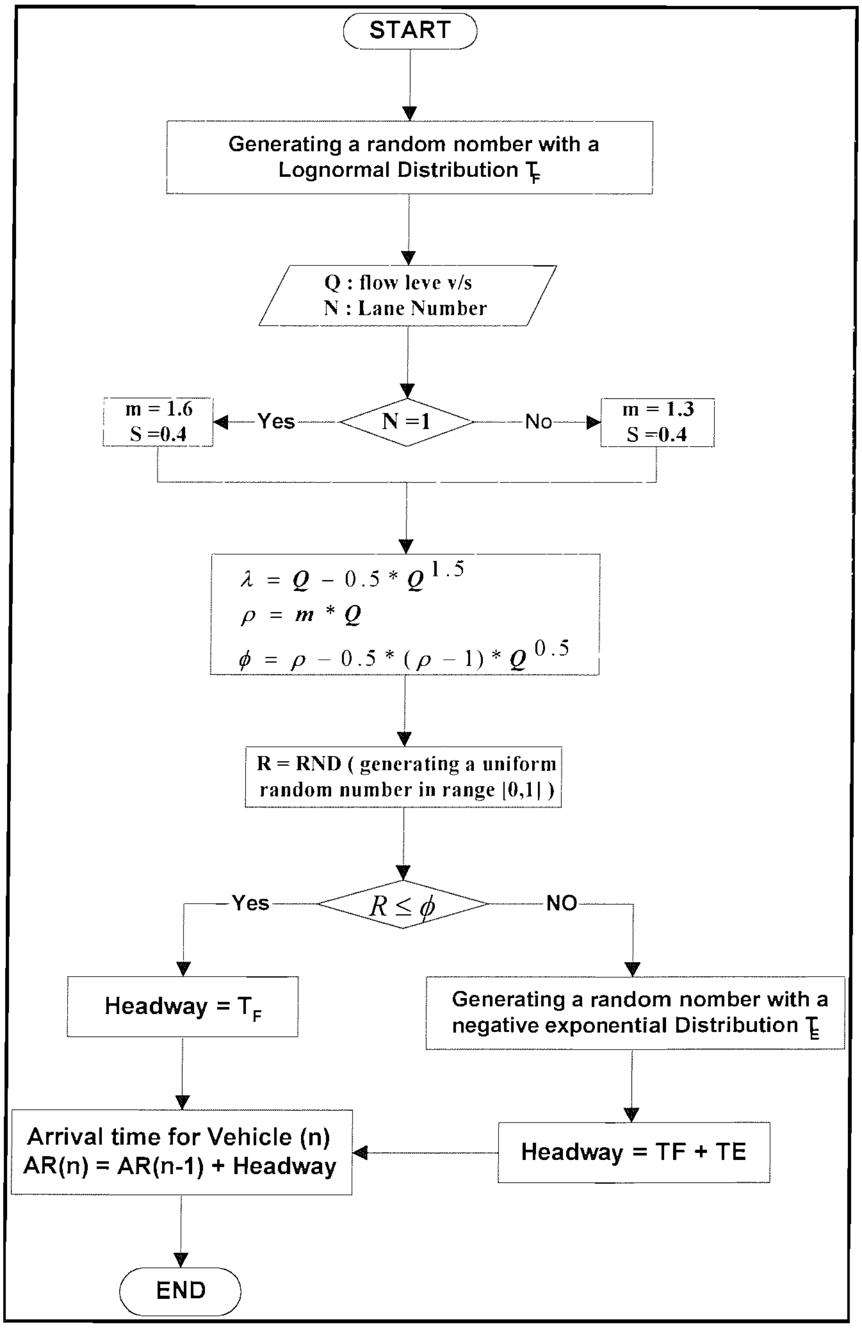


Figure 5.1: The flowchart for the vehicle arrival subroutine.

5.4.1. Generating Random Numbers With A Negative Exponential Distribution:

The corresponding cumulative distribution function for the negative exponential distribution is:

$$P(H_t < t) = 1 - e^{-\lambda . t}$$

Applying a uniform random number (R) [within the range (0,1)] to the left side of the previous equation, would produce a number (t) that has an exponential distribution, as below:

$$R = 1 - e^{-\lambda . t} \qquad \Rightarrow \qquad t = - Ln (1 - R) / \lambda$$

Generating R is straight forward using a set function for random number generation.

5.4.2 Generating Random Numbers With A Normal Distribution:

There are two ways random numbers can be generated with a normal distribution, These are:

a- Approximate Method:

The idea of this method is that the total of any uniformed random numbers (R_i), (i = 1,2 ....n) can be converted into another number which is approximately normally distributed (NAYLOR, T.H. 1966)[44] by the equation:

$$Z = \frac{\sum_{i=1}^{n} R_i - \frac{n}{2}}{\sqrt{\frac{n}{12}}} \qquad .....5.16$$

The accuracy of the method increases with the increase of the uniformed random numbers. SKABARDONIS recommended the use of value n=12 in consideration of the computation time and accuracy (SKABARDONIS 1982)[35].

b- Box And Miller's Method:

BOX and MILLER (BOX and MILLER 1958)[45] converted two uniformly distributed random numbers U_1 and U_2 into new numbers X_1 and X_2 which are independent and normally distributed by using the following two equations:

$$X_1 = \sqrt{(-2LnU_1)} * Cos(2\pi U_2) \qquad ....5.17$$

$$X_2 = \sqrt{(-2LnU_1)} * Sin(2\pi U_2) \qquad ....5.18$$

In order to obtain random numbers that have a normal distribution with mean μ and standard deviation σ, the following equations are applied:

$$X_1 = \mu + \sqrt{(-2\sigma^2 LnU_1)} * Cos(2\pi U_2) \qquad ....5.19$$

$$X_2 = \mu + \sqrt{(-2\sigma^2 LnU_1)} * Sin(2\pi U_2) \qquad ....5.20$$

5.4.3. Generating Random Numbers With a Lognormal Distribution:

In order to have a random number with a lognormal distribution, the exponential of a normal distributed random number is calculated. (e.g. R is a random number with a normal distribution then $R1 = e^R$ would be a random number with a log-normal distribution).

If m and S are the mean and standard deviation of the log-normal distribution respectively, then:

$\mu = Ln(m) - \sigma^2/2$ & $\sigma^2 = Ln(S^2 / m^2 + 1)$

μ and σ would be the mean and standard deviation of the normal distribution.

5.5 DETERMINING FREE SPEED DISTRIBUTIONS:

Free speeds (also called the desired speeds) can be defined as the speeds that drivers would adopt if they were not influenced by the presence of other vehicles. The time headway is usually used to determine whether or not a driver is impeded. There are different ways to determine the critical headway value Ht_{cr} over which the situation would be regarded as free driving. BRANSTON found that Ht_{cr} varies between 3 sec to 4.5 sec depending on the method of approach (BRANSTON, D. 1979)[46]. The HCM (Highway Capacity Manual 1994)[5] recommended a value of 5 sec as the critical headway for following.

5.5.1 Data Collection And Analysis:

A database was collected by video recording flow on a three lane motorway (M3) northeast of Southampton. The time was between 9:00 to 11:00 am, the weather was dry, and the visibility was good. The speed and headway for every vehicle were measured. Also the vehicle type and the lane number were recorded.

Then, any vehicle with a headway less than the critical value (5 sec) was dismissed. In the analysis two type of vehicle was distinguished:

1- Light vehicles [passenger car and van].

2- Heavy vehicles [HGV and Coach].

As expected the normal distribution offered the best fit for the speed and table 5.5 shows the statistical summary and Figures 5.2 and 5.3 show the normal distribution fits:

Veh. Type	Lane	Mean (m/s)	Median (m/s)	Std. Dev. (m/s)	Min (m/s)	Max (m/s)	K-S test
Light	1	27.84	27.78	3.50	19.95	37.04	0.256
Light	2	31.59	31.25	3.07	21.74	39.22	0.063
Light	3	36.50	37.04	3.04	27.78	44.44	0.088
Light	1, 2, 3	31.35	31.25	4.62	19.95	44.44	0.066
Heavy	1 & 2	23.26	22.73	1.43	19.24	27.03	0.143

Table 5.5: Free Speed statistical summary

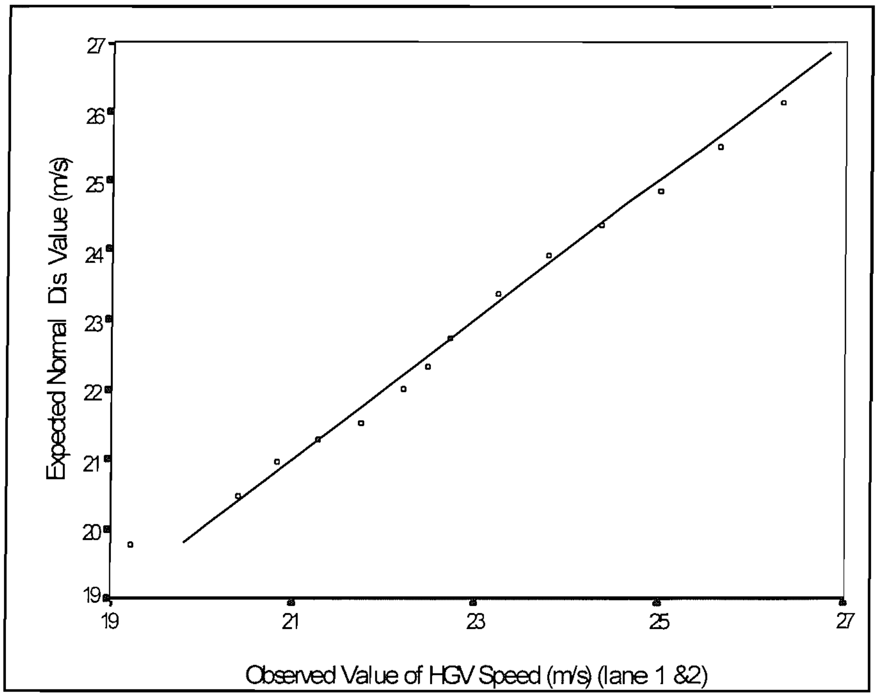
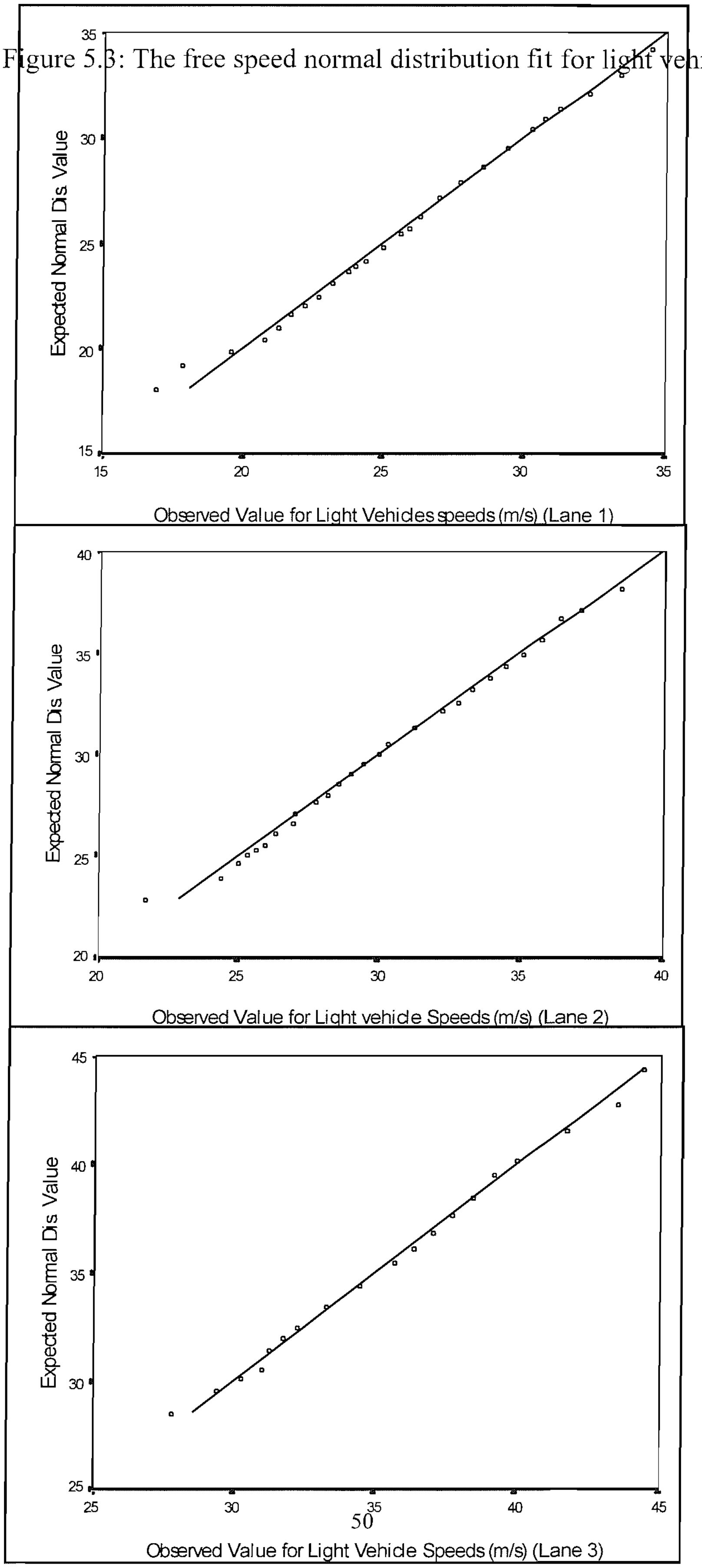


Figure 5.2: The free speed normal distribution fit for HGV (lane 1 and 2).

Figure 5.3: The free speed normal distribution fit for light vehicles



BURROWS found that the free speed normal distribution of any lane can be derived from the free speed normal distribution of the whole traffic stream for a three lane motorway, for the same type of vehicles (BURROWS 1974)[47].

Assuming N(μ , σ) is the free speed of the normal distribution which represents the whole traffic stream, (μ : is the mean & σ : is the variance). The distribution parameters for the three lanes would be:

Lane	Mean	Variance
1	$\mu - \sigma$	$2*\sigma/3$
2	μ	$2*\sigma/3$
3	$\mu + \sigma$	$2*\sigma/3$

The data analysis supports this theory, and Table 5.6 shows the measured and calculated means and variances for the free speed distributions for each lane:

LANE	Calculated Parameters		Measured Parameters	
	Mean	Variance	Mean	Variance
1	26.725	3.082	27.84	3.50
2	31.348	3.082	31.59	3.07
3	35.971	3.082	36.50	3.04

Table 5.6 shows the measured and calculated parameters for the free speed distribution for every lane [Light Vehicles]

Because the model generates the same level of flow for each lane, the free speed distribution will be for the whole traffic. It has parameters:

Light Vehicles : mean (μ) = 31.35 m/sec & Std (σ) = 4.623 m/sec.

Heavy vehicles : mean (μ) = 23.32 m/sec & Std (σ) = 1.43 m/sec.

The interaction between vehicles in the simulation over the warm up section will redistribute the vehicles between lanes. However, Table 5.6 will be used to determine the favourite lane for the drivers of the light vehicles according to their free speed and original lane. The favourite lane concept is useful to simulate the lane-hogging phenomenon by some drivers. Some drivers prefer to stay on their favourite lane as long as other vehicle can pass them, this concept was introduced first by the SISTM model (WOOTTON JEFFREYS 1990)[29] and was called the 'lane-hogging' phenomenon. Because the HGV is slow vehicles, they most likely to be in lane one and the presence of any HGV on lane two is probably due to overtaking slower vehicles. Therefore, it was suggested to use lane one as a favourite lane for HGV. Figure 5.4 shows the flowchart diagram of how to determine the favourite lane for a driver at the moment of his or her arrival.

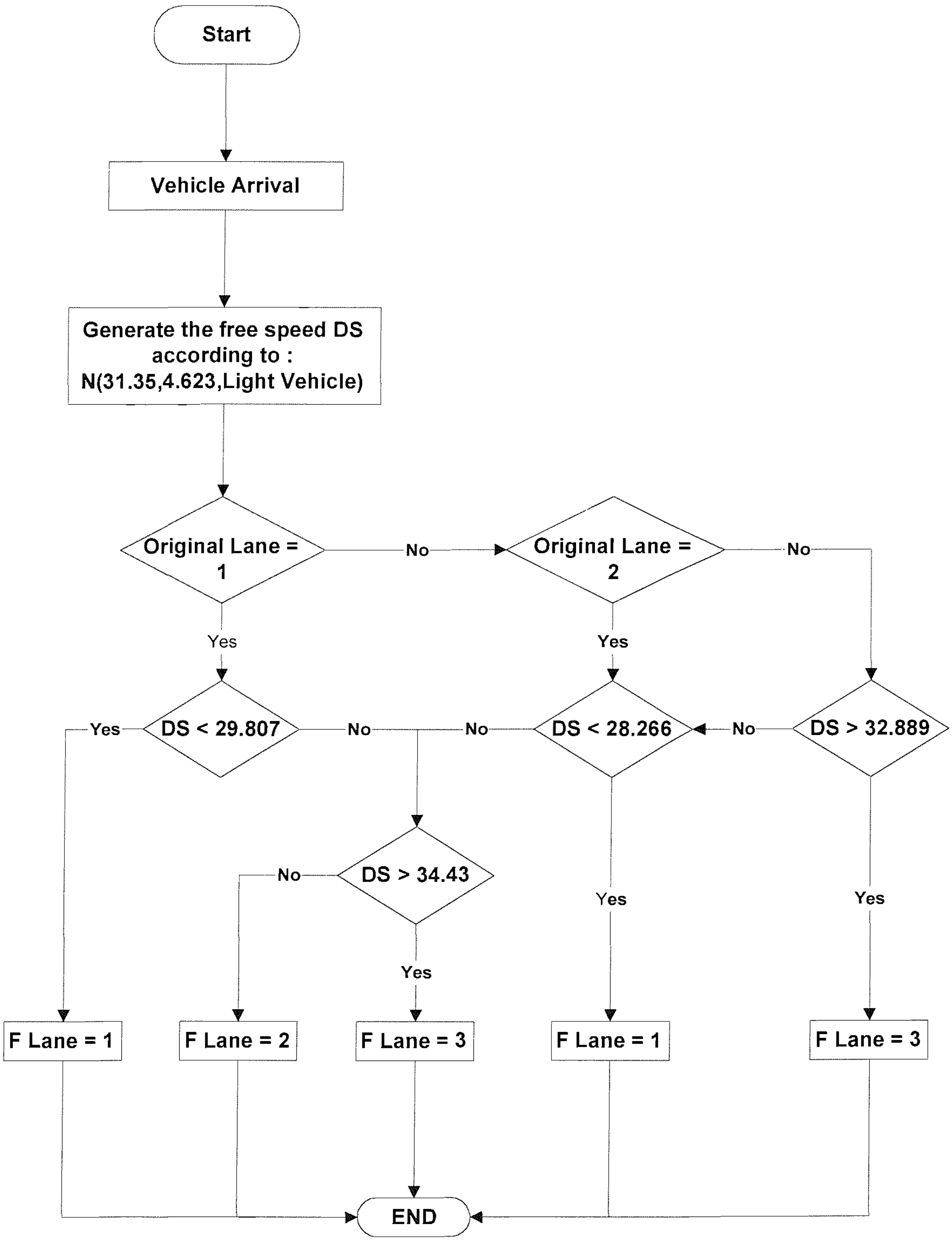


Figure 5.4: The Determination of the Favourite Lane

CHAPTER 6

TIME MANAGEMENT

6.1 INTRODUCTION:

Time management is an essential element in any simulation model. The efficiency of any model depends on its time increment (Epoch) during which the model recalculates the speed and position of all the simulated vehicles. The shorter the Epoch the slower the model. Also, human's reaction time influences the Epoch, and clearly the Epoch should be less than or equal to the reaction time. This chapter is concerned with human reaction times and the time increment in the model.

6.2 HUMAN REACTION TIME:

Reaction time is used to describe the period between the occurrence of a 'signal' (eg. brake light) and the driver's physical reaction to it. Obviously, reaction time increases with the increase in decision complexity and information content. GARBER AND HOEL considered reaction time to consist of four elements (GARBER and HOEL 1988)[48]:

1- Perception time: The time of perceiving information by the driver's eyes.

2- Identification time: The time of understanding the stimulus.

3- Emotion time: The time of taking the decision.

4- Volition time: The time of the driver's physical motion to perform the decision.

Understanding the previous concept opens the way to better design to the reaction time in simulation models. For example, if a driver perceives a front vehicle braking, the first reaction would be likely to take a long processing time. However, subsequent reaction to the perceived information from the front vehicle will become shorter because the following driver assesses the changes quicker than before. Thus, a model should have the mechanism to simulate the change in reaction time between the start of any event and the subsequent following processes.

For any events the measurable data is the point of action and that includes the reaction time. Therefore, it is difficult to measure reaction time unless other types of experiment and data measurement are conducted. JOHANNSON AND RUMER studied drivers' response times to anticipated braking and found that one second reaction time covered about (88%) of drivers (JOHANNSON & RUMER 1971)[49]. Figure 6.1 shows the distribution of the reaction time conducted, and the percentage of the drivers according to their reaction times. However, it can be argued that other drivers had a longer reaction time due to the condition that they were under the perception threshold.

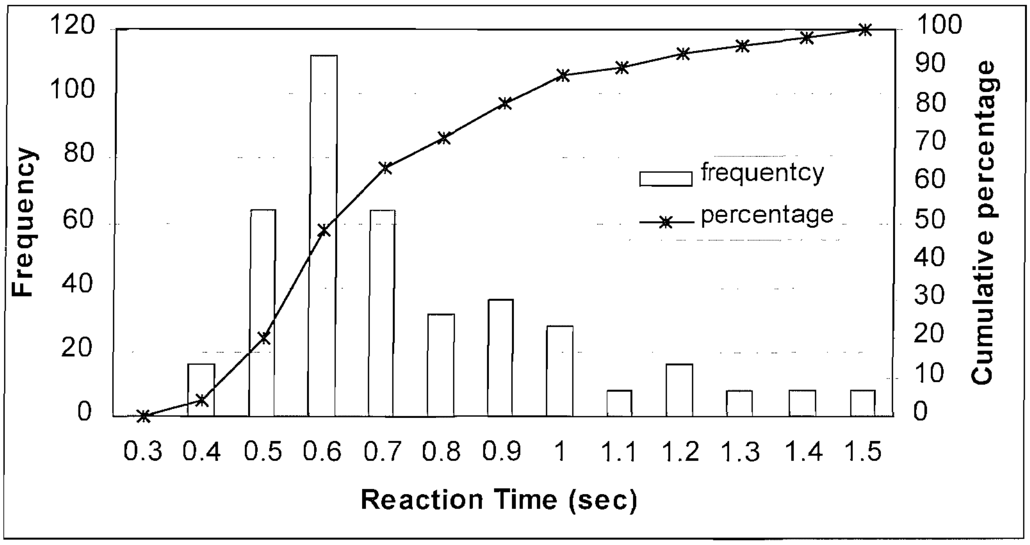


Figure 6.1: The frequency and cumulative percentage of the brake reaction times, (JOHANNSON & RUMER)[49]

BOFF & LINCOLN distinguished three types of reaction time (RT) (BOFF & LINCOLN 1988)[50] :

1- **Simple RT**: When there is one stimulus and the subject makes only one response.

2- **Disjunctive RT**: When also there is only one response to a single stimulus, but other stimuli may be presented as distracters.

3- **Choice RT**: When there are multiple stimuli and multiple response.

Obviously, our aim is the choice RT which replicates the driver situation better than the others.

A- *The Effect of the Number of alternatives on Choice reaction time:*

WOODWORTH found that the RT increases as the number of stimuli and responses increases (WOODWORTH, R.S. 1938)[51]. Also, it has been found that the RT increases at fairly a constant rate as the number of alternatives (N) doubles, see Table 6.1. This indicates that RT is a logarithmic function of the number of alternative stimulus-response pairs:

$$RT = a + b \, Log_2 N$$

However, this result does not work well when the condition is not ideal.

Number of Alternatives S-R pairs (N)	**Reaction Time (msec)**	**Reaction Time increment (msec)**	**RT increment per Doubling of N (msec)**
1	187	187	1 to 2 : 129
2	316	129	
3	364	48	2 to 4 : 178
4	434	70	
5	487	53	3 to 6 : 168
6	532	45	
7	570	38	4 to 8 : 169
8	603	33	
9	619	16	5 to 10 : 135
10	622	3	

Table 6.1: The Reaction time as a function of number of alternative (WOODWORTH. 1938)[51].

B- *The Effect of The Probability Of Alternatives On The Choice RT:*

Condition	Probability	RT (msec)
Equal probability	0.11	390
High probability	0.94	285
	0.75	320
	0.24	375
Low Probability	0.095	405
	0.03	425
	0.01	440

Table 6.2: The effect of Probability of Alternatives

Because of the increase of the number of the alternative stimulus-response pairs, the probability of occurrence of any individual stimulus-response pair will decrease. Table 6.2 presents the result from experiments by FITTS & PETERSON when they investigated the effect of different probability on the reaction time (FITTS & PETERSON 1963)[52]. Obviously, when the probability decreases the RT increases.

C- *The Effect of Warning Interval on Choice reaction time and Error:*

POSNER AND KLEIN found that RT has a U-shaped function with warning interval, and it reached the lowest value when the interval was around 200 msec (POSNER AND KLEIN 1973)[53]. However, the percentage error has an inverted U relationship with the warning interval, with the greatest percentage error at 100 msec. This relationship in which RT decreases and error rate increases is known as the speed-accuracy trade off. However, a compatible stimulus and response give shorter overall RT and fewer errors than an incompatible stimulus and response, Figure 6.2.

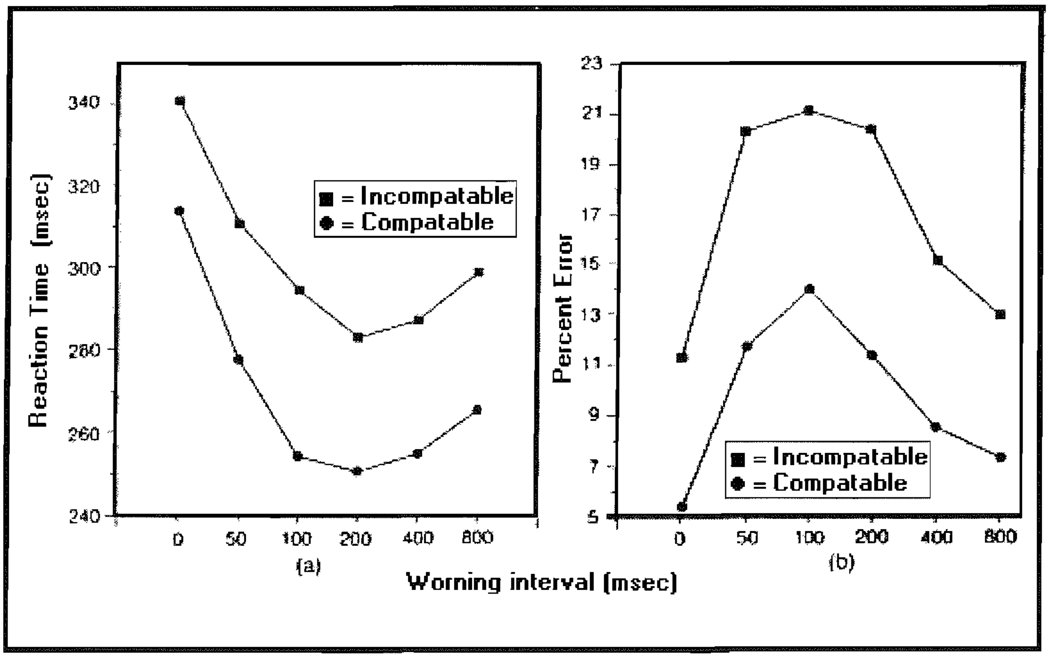


Figure 6.2: Function relating (a) RT for correct responses and (b) errors to duration of warning-signal interval for compatible and incompatible pairings of stimuli and responses (POSNER)[53].

6.2.1 Conclusion:

Due to the complexity of the reaction time process, it is very difficult to develop a model which replicates all the circumstances and conditions that affect RT. By combining the concept of GARBERT, the results of JOHANNSON and other researchers, the assumptions that the Model will rely on are:

1- The driver will perceive the stimulus of DV and DX in (0.5sec) delay. (A reasonable value according to the data from BOFF and LINCOLEN).

2- The first perception of braking from the lead vehicle will have a delay of (1 sec), unless there is a brake light then the delay would be 0.75 sec.

3- The driver will not perceive any information if he or she is under the threshold of perception.

6.3 THE EPOCH:

The Epoch is the time increment in the any simulation model, during which the program will repeat the calculation to assess the new position and speed for every vehicle in the model. Therefore, during the Epoch every driver is assumed to have a constant acceleration or deceleration, which is applied to the motion equations in the model. The importance of the Epoch varies from model to model depending on the assumptions. Thus, in Gipps's model, the Epoch is the reaction time of the driver. In the action point Model, the Epoch length is related to the amount of error that happens when the driver crosses a certain threshold. For example, suppose that DV_0 and DX_0 is the relative speed and the relative distance respectively at the moment [t] between two successive vehicles. Also, suppose that DV_0 and DX_0 were the values which are just under the threshold of the follower driver. At the moment (t + Epoch) DV_1 and DX_1 becomes over the threshold if the relative acceleration lead to a higher DV.

This situation is illustrated in Figure 6.3. The error in evaluating the threshold will increase with Epoch length. In order to simulate the variation of the reaction time, we have to use a short Epoch, which consequently will minimise the error in evaluating the thresholds. An Epoch value of [0.25sec] would enable us to simulate a reaction time of (0.5, 0.75, 1sec ...etc.).

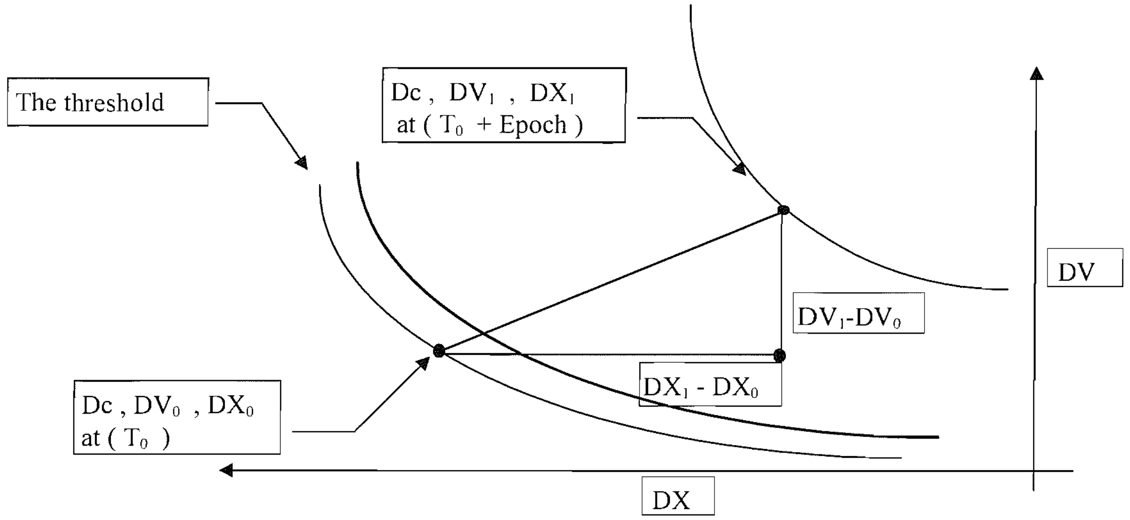


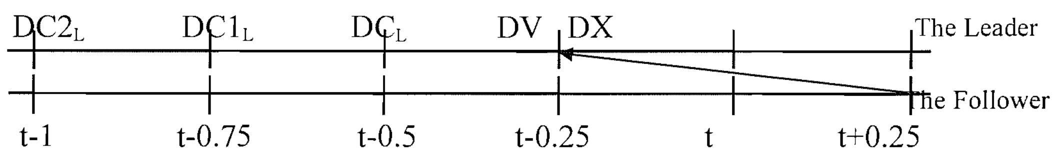
Figure 6.3: The error in the threshold according to the Epoch.

6.4 THE TIME SEQUENCE IN THE MODEL:

Due to the assumption that the minimum reaction time in the model is 0.5 sec, The driver response at the moment (t+0.25) would vary according to the level of relative speed and distance at the moment (t-0.25):

$DV(t\text{-}0.25) = V_L(t\text{-}0.25) - V_F(t\text{-}0.25)$ & $DX(t\text{-}0.25) = X_L(t\text{-}0.25) - X_F(t\text{-}0.25)$

At the same time the driver will perceive the deceleration/Acceleration in 0.75 sec delay.



Where :

$DC2_L$: Acceleration / Deceleration that would be perceived when there is change in the situation of the lead vehicle and there is no indicator such as brake light.

$DC1_L$: Acceleration / Deceleration that would be perceived when there is change in the situation of the lead vehicle and there is indicator such as brake light.

DC_L : Acceleration / Deceleration that would be perceived when there is no change in the situation of the lead vehicle.

Figure 6.4: The Time sequence in the model

The driver will add another delay (0.25sec) to the perception of the deceleration or acceleration, when there is a change in the situation of the lead vehicle (i.e.: constant speed to Accelerating /Decelerating). Also another delay would be added (0.25sec) if the brake lights were not active. Figure 6.4 illustrates the time sequence in the model.

CHAPTER 7

MODELLING THE APPROACH PROCESS

7.1 INTODUCTION:

Although a substantial amount of research has been undertaken or is underway in the examination of the 'Close Following Process', comparatively little has been published concerning the 'Approach Process'. This process describes driver behaviour when approaching a slower vehicle. There are two main alternative theories regarding the behavioural basis underlying this process, based on Time-To-Collision and Optic Flow. However, there is very little data to against which the relative merit of the models can be assessed. This Chapter examines these theories, and explores the two key features of the process: the circumstances under which driver deceleration is instigated, and the process governing the control of the deceleration itself. Finally, a model was developed to simulate the Approach Process based on the database collected.

7.2. REVIEW:

7.2.1 The Optic Flow Theory:

The Optic Flow ($d\theta$ / dt) is defined as the change in the visual angle over time. It is logical that deceleration could not begin until it is possible for the driver to actually perceive his relative motion, with ($d\theta$ / dt) exceeding a threshold. Such perceptual studies are well known and have been in place in driver behavioural models for several decades, (MICHAELS 1961)[18] (TODOSIEV 1963)[19], and has been adopted by WIEDEMANN and others as the basis for the MISSION simulation models (LEUTZBACH & WIEDEMANN 1986)[21].

The MISSION model assumes that the closing vehicle will start its Approach process after passing a relative speed threshold called SDV. Empirical DATA was used to explore the shape of the SDV threshold. That led to the following equation:

$$SDV = \left(\frac{DX - AX}{CX}\right)^2 \qquad \text{....7.1}$$

$$CX = CX_{const} * (CX_{add} + CX_{mult} * (RND1(i) + RND2(i))) \qquad \text{...7.2}$$

Where :

RND1(i) : A random number which has a value between [0-1] with a normal distribution N(0.5, 0.15).

RND2(i) : Relates to the driver's estimation ability, It is normally distributed N(0.5, 0.15) and takes values [0..1].as RND2 become near to 1 the driver's estimation is better.

CX : Its value should represent the range between 25 and (75 or 50), CX_{const} , CX_{add} and CX_{mult} are parameters that define the range of SDV threshold

DX : Relative distance (m).

AX : Stopping distance (m)

After drivers cross this threshold they then decelerate at a rate:

$$B(i) = 0.5 * \frac{DV^2}{ABX - DX} + B(i-1) \qquad \text{.....7.3}$$

Where:

(i) is a number related to the order of the vehicle : (i : follower , i-1 : leader).

ABX : The desired minimum following distance at low speed difference.

$$ABX = AX + BX \qquad \text{......7.4}$$

Where:

$$BX = (BX_{add} + BX_{mult} * RND1(i)) * \sqrt{V} \qquad \text{......7.5}$$

BX_{add} & BX_{mult} : calibration parameters defining the rang of variation.

[Closing $\Rightarrow$ V(i-1) - Opening $\Rightarrow$ V(i)].

AX : The desired separation distance for stationary vehicles (front to front distance). It consists of two parts: The length of the lead vehicle and the desired front-to-rear distance of the driver in the following vehicle (i) . The later is normally distributed depending on the safety need of the driver:

$$AX = L + AX_{add} + RND1(i) * AX_{mult} \qquad \text{.....7.6}$$

where

AX_{add} & AX_{mult}: calibration parameters.

L : is the length of the lead vehicle.

The above model does not consider some important points:

(I) The response to any deceleration by the vehicle ahead should depend on the relative distance and the level of the lead vehicle's deceleration.

(II) The Mission model assumes that drivers perceive the real relative speed and the relative distance value, whatever the spacing distance. Recent research has shown that drivers underestimate the Time-To-Collision, i.e. drivers do not perceive the real value of relative speed and distance (this will be discussed in the next paragraph).

(III) The method of approach and the validity of the data used to calibrate the MISSION model are unclear.

7.2.2 The Time To Collision Theory:

In 1976 LEE put forward a theory of visual control of braking, based on information about the time to collision (LEE D. N. 1976)[54]. This theory assumes that the drivers judge their decisions to start braking on the basis of the Time-To-Collision and the headway with the vehicle ahead. Also, it assumes that the level of braking during the closing process stage, depends on the reduction in the Time-To-Collision which the driver wants.

Lee said that *the time to collision is specified by the ratio of the values of the optic variables [θ] and [V_θ] at any point in the sector of the optic flow field.*

Where:

θ : The angular separation of any two image points.

V_θ : The change in the angular separation by the time ($d\theta / dt$)

$$TTC(t) = \theta / (d\theta / dt) \quad \Rightarrow \quad TTC`(t) = (dTTC/dt) \qquad \ldots.7.7$$

Thus, the driver will start his closing process stage when the time to collision becomes equal to or less than a certain value [TTC_m]. After that, deceleration will be controlled by choosing the rate of [TTC`]. However, if the rate of TTC` is less than -0.5 the driver will be on a collision course and, consequently, he or she will try to keep [TTC`] greater than or equal -0.5. Figure 7.1 shows the relationship between the relative speed and the relative distance according to the change in the [TTC`].

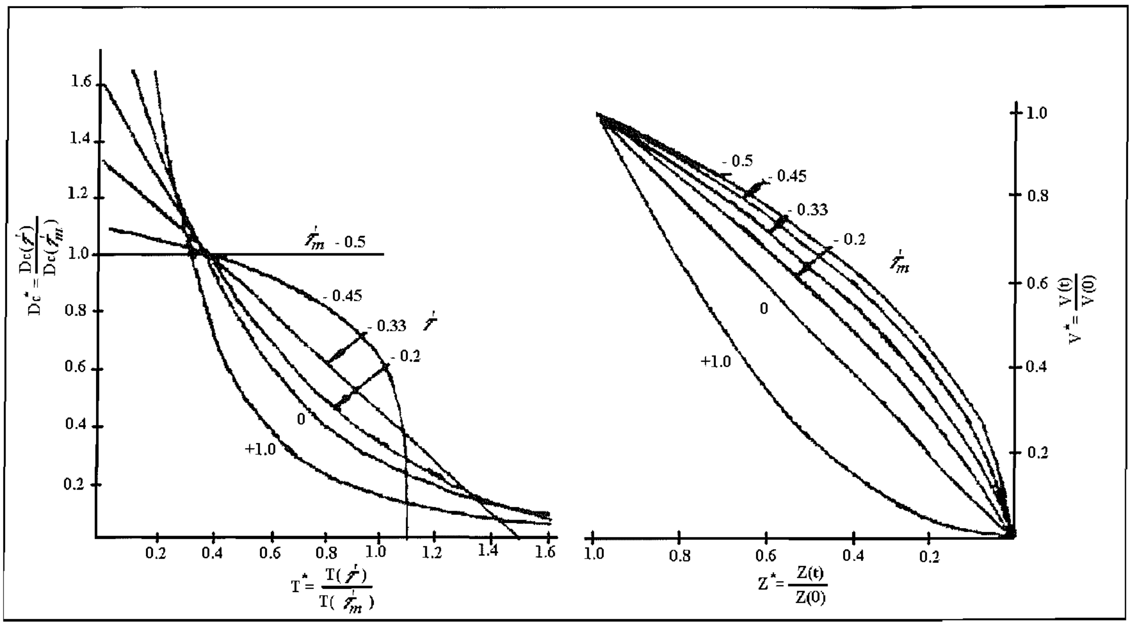


Figure 7.1: Normalised deceleration versus time and velocity versus separation curves for hypothetical driver who is controlling his braking, to stop at a stationary obstacle, by maintaining the visual variable $\dot{\tau}$ at different safe margin values $\dot{\tau}_m$. The visual variable $\dot{\tau}$ is the time derivative of the visual variable τ which is specifies the time to collision if the current closing velocity were kept constant. Dc(t), V(t) and Z(t) are respectively : the Closing deceleration, closing velocity and separation at time t. D_c ($\dot{\tau}_m$) is the constant deceleration which would be required. t_{bc} is the braking time under a constant deceleration. V(0) and Z(0) are initial closing velocity and separation. (LEE 1976)[54]

The theory of visual control of braking is an important step in understanding how the driver responds in the approach process. However, later research (will be presented later) revealed that drivers usually underestimate the real value of time to collision, thus the theory should be modified to incorporate this phenomenon. However, LEE assumed that the driver will always incorporate the deceleration of the lead vehicle, regardless its level and how distant the leader.

The Estimation Of Time to Collision:

The Time-To-Collision [TTC] is the time that would take a following vehicle to collide with a leading one, if the current relative speeds were maintained from the given headway. In principle, there are at least two distinct ways in which TTC information may be obtained by the driver:

1- Derived from low - order information (the cognitive method) :

TTC = Distance to object / velocity .

2- Directly from the changing optic array (the optic-flow method) :

$$TTC = \theta \, / \, (d\theta / dt) .$$

The first method gives an indication that there are two factors controlling the estimation of time to collision, while the second method indicates that changes in the visual angle by time ($d\theta / dt$) controls the estimation of time to collision.

Although there had been some work on the TTC before Lee's theory, the main effort came after Lee had proposed his model. The major research is:

- MCLEOD AND ROSS: examined the effects of viewing time on estimations of TTC. They found that the participants underestimated TTC as approximately 60% of the actual time to collision (MCLEOD &ROSS 1983)[55].

- CAVALLO AND LAURENT: examined driver experience levels, distance evaluations, and vehicle approach speeds on estimates TTC. The result of their study indicated that estimates of TTC were systematically underestimated (CAVALLO & LAURENT 1988)[56].

- GOEGER AND CAVALLO: examined the difference between the situation when the driver is not on collision course and when the driver is on collision course. They found that usually drivers estimate accurately when they are not on collision course (GOEGER & CAVALLO 1991)[57].

- HOFFMANN AND MORTIMER: examined the ability to estimate TTC and the influence of headway, approaching speed and viewing time on the estimation. They tried a different method which was more realistic and the result indicated a considerable speed influence on the estimation (HOFFMANN & MORTIMER 1994)[58].

- SIDAWAY AND FAIRWEATHER: examined the influence of viewing time and closing speed. The results also showed a significant influence to the closing speed on the estimation (SIDAWAY & FAIRWEATHER 1996)[59].

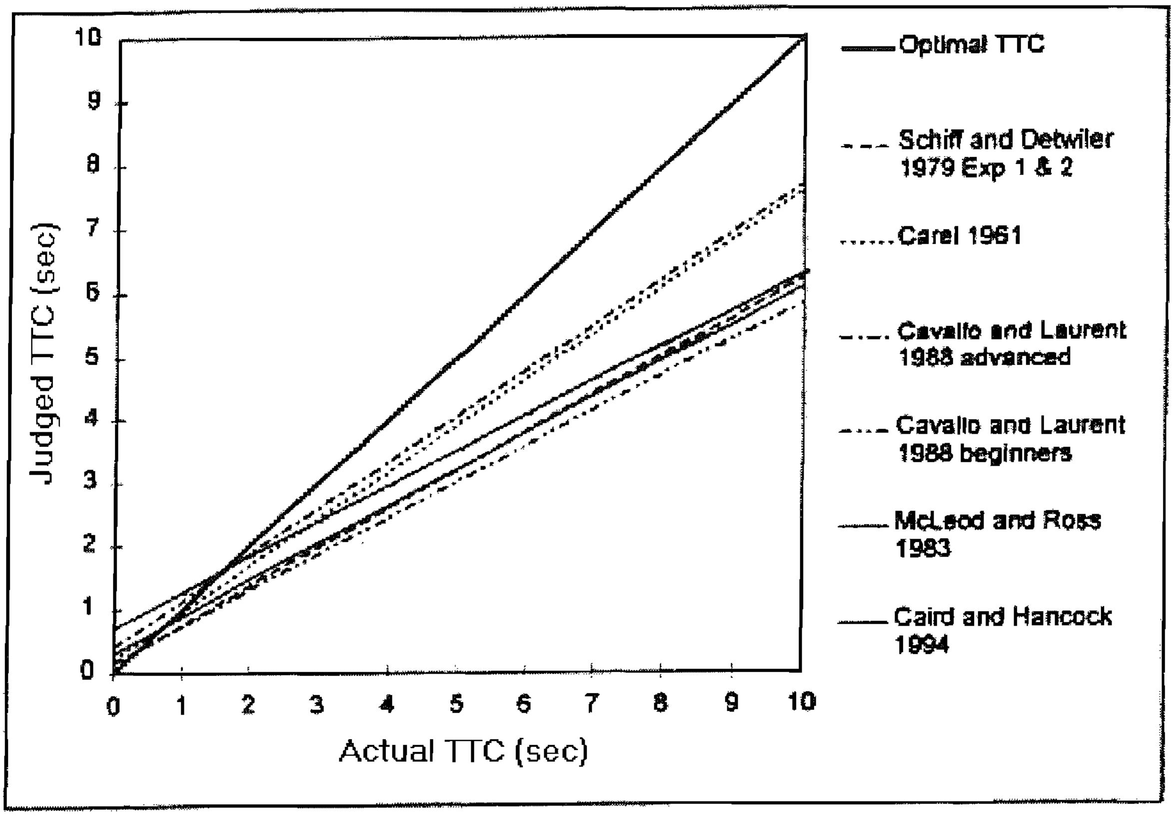


Figure 7.2: Results of previous time-to-collision research for the cited studies. The collective findings show that participants underestimate time-to-collision and that such underestimation grows with the absolute duration of actual time-to-collision.

All the above researches found the same result that drivers underestimate Time-To-Collision, and there is a relationship between the actual time to collision and the estimated one. Moreover, the accuracy of estimation increases with the experience of the driver and the optic flow rate. The results of research in estimating TTC are shown in Figure 7.2.

7.2.3 Other Studies:

SPURR examined the time series behaviour of fifteen drivers in approaching a stationary object on the road (SPURR, R.T. 1969)[60]. Although these individual deceleration traces seemed on inspection to be quite erratic, on reduction to a dimensionless co-ordinate system [(Actual speed/ Approach speed) against percent of brake application time]. He concluded that "*on average most drivers responses could be characterised by a sudden reaction to a maximum deceleration followed by a distinctive decay curve*".

7.2.4 Conclusion:

It is possible to conclude that although many indications exist regarding deceleration behaviour during the approach process, few dedicated studies have been undertaken. This leaves a substantial gap in knowledge, which this research will contribute to filling.

7.3 DATA COLLECTION:

In order to investigate drivers' behaviour on approaching slower vehicles, two series of experiments were performed. The objective of the first one was to determine driver perception threshold of relative speed. The second was to examine the factors and circumstances under which driver deceleration is started, and the process governing the control of the deceleration during the approach processes of different drivers.

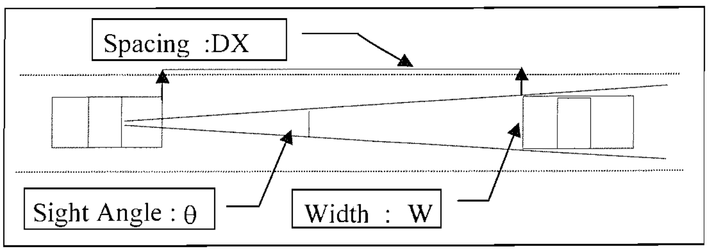
7.3.1 Determining The Perception Thresholds:

The TRG instrumented vehicle described earlier, provides an opportunity to measure the threshold in dynamic situations. In previous studies a simulator has been used generally, ie (TODOSIEV 1963)[13] and (HOFFMANN 1994)[58]. The threshold was defined in three ways:

1. By the Optic Flow : Which is the change in the angular separation by the time ($d\theta/dt$) :

$tg(\theta/2)=(W/2)/DX_0 \Rightarrow$

$(\theta/2)=arctg\,(W/(2*DX_0))$



$$\Rightarrow \quad \frac{d\theta}{dt}=\frac{-4*W*DV}{4*DX_0^2+W^2} \qquad \ldots 7.9$$

where :
W : The width of the lead vehicle [m]
DV : The relative speed [m/sec]
DX_0 : The net spacing between the two vehicles.

2. By the relative speed with the vehicle ahead of the driver DV.
3. By the reciprocal of the Time-To-Collision (1/TTC). Which essentially corresponds to DV scaled to remove any distance effects.

Four drivers (aged 23, 30, 33 and 37 years) were asked to drive the IV freely on the M3 motorway between Southampton and London. All the test subjects had at least three years driving experience and clean driving licences. The motorway traffic was moderate (about 3000 vph) and visibility was very good (mid-day and dry weather). In all nine experimental runs were performed, each of which consist of a 90 minute drive. Drivers were asked at a random intervals (the experimental supervisor giving an oral sign) about the relative situation with the vehicle ahead in the same lane (the spacing distance was less than 120m). The answer should be one of three:

1- Closing: If the driver thinks that he is closing to the vehicle a head of him.
2- Opening: If the driver thinks that he is slower than the vehicle ahead of him.
3- Constant: If the driver thinks that he is on the same speed as the vehicle ahead of him.

The previous answers are the simplest way to present a driver's perception. They offer a physical meaning to the perception in one word, which minimises the stress load over the participants in the experiment. All the subjects were asked to answer as quickly as possible (the delay must not be more than one second), otherwise the answer was rejected. The IV Audio-Video recording system was used to re-examine the subjects' answers before accepting corresponding data. The data [relative speed, relative distance, speed and acceleration] was recorded every time the driver answered.

In the situations where the test drivers are approaching a slower vehicle far ahead or the lead vehicle is pulling away, their perception might sometimes be judged according to the recent change in spacing distance. This is difficult to measure and detect compare to the relative speed or the optic flow. However, the data bias would occur in the closing and opening situations and where the spacing distance change rate is high enough to clear any confusion with the constant situations.

7.3.1.1 The Data Analyses And Discussion:

The analyses were undertaken by dividing the data initially into four groups according to the relative distance (DX). This will help to check the consistency in drivers' perception over different relative distance ranges.

The relative distance ranges are:

1- (DX < 20m) 2- [(DX< 40m) and (DX >= 20m)]

3- [(DX < 80m) and (DX >= 40m)] 4- (DX >= 80m)

The probability of perception was calculated over different ranges for each of the optic flow (dθ/dt), the relative speed (DV) and the ratio (DV/DX). Figures 7.3, 7.4 and 7.5 show how the probability of perception changes according to the value of (dθ/dt), (DV) and (DV/DX) respectively. Several conclusions can be drawn from these data:

1- The Optic Flow level is not steady over the change of the relative distance, and it increases with a decrease in relative distance. Also, it becomes unacceptable to use the optic flow to define the perception threshold when the relative distance is less than 20 meters. These results are different from those of TODOSIEV (TODOSIEV 1963)[13] in which he suggested that the Optic Flow could be used to replicate the perception threshold as an independent value. Perhaps the difference in the results is due to the differences in the method of measurement between these two experiments [dynamic situation in real traffic and simulator screen]. It can be argued that the dynamic situation and real life data is better than the artificial situation and simulator, because the first offers closer environment to reality and better viewing for the subject.

2- Relative speed seems to be more consistent (no fluctuation) than the optic flow regarding the relative distance is less than 20m. However, it is clear that the level of the relative speed increases with an increase in distance. The (DV/DX) percentage looks much more consistent over the distance ranges, although it suffers from slight shift in the opening situation. EVANS and ROTHERY used the ratio (DV/DX) to measure the perception threshold of human (EVANS and ROTHERY 1973 & 1977)[61,62]. They obtained their data by using a simulator.

3- There is a small but distinct possibility that a driver may make a wrong interpretation of events, for example he or she may believe that the inter-vehicle distance is increasing whereas, in reality, the opposite is happening. This phenomena has been well known for several decades and the result of those experiments is in agreement with the others (EVANS and ROTHERY 1977)[62]. It is very difficult to apply such phenomena in a deterministic model and its effect was neutralised by adding its probability to the constant situation probability in the model. This will be illustrated later.

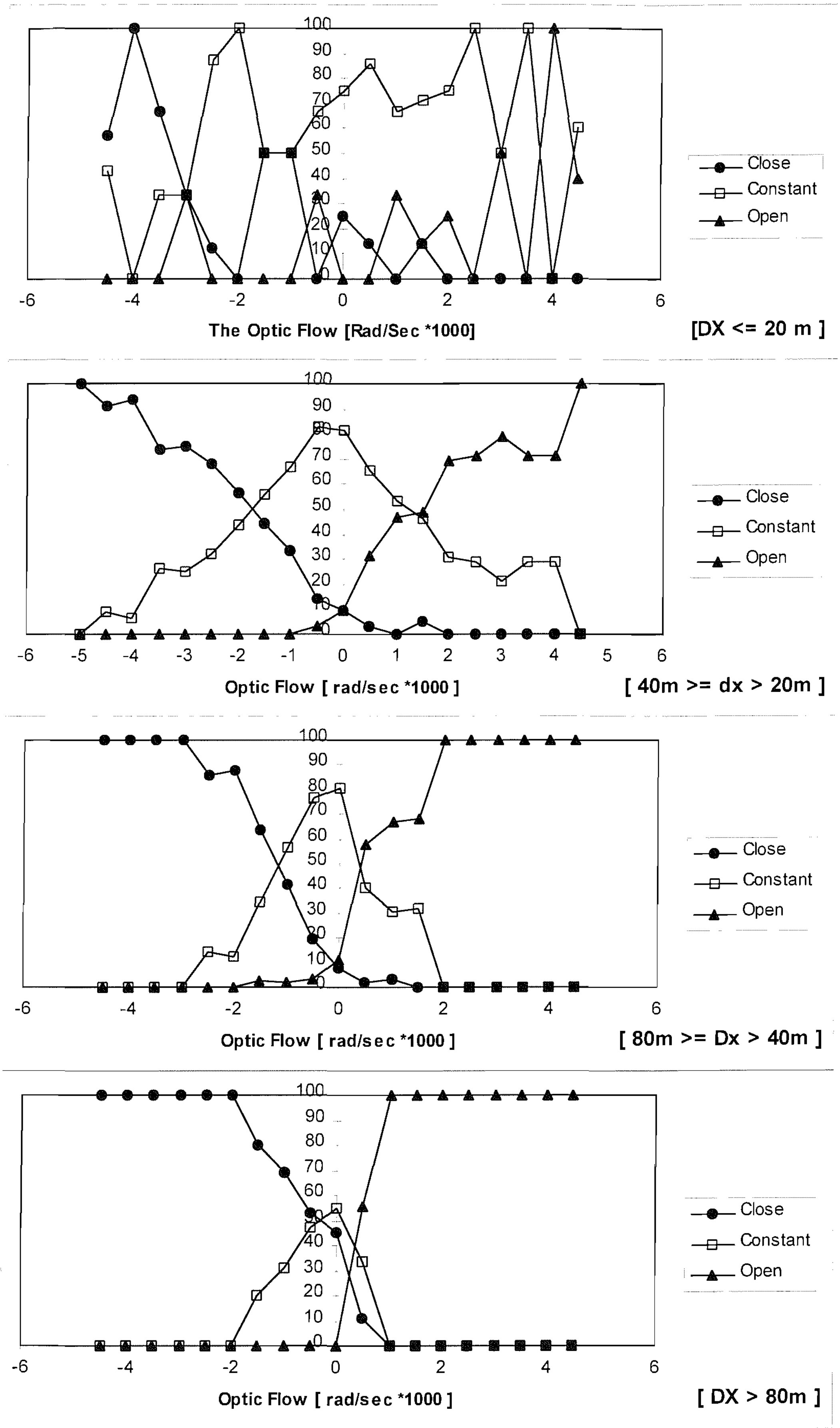


Figure 7.3: Probability of Perception according to the Optic Flow [DX: the spacing distance]

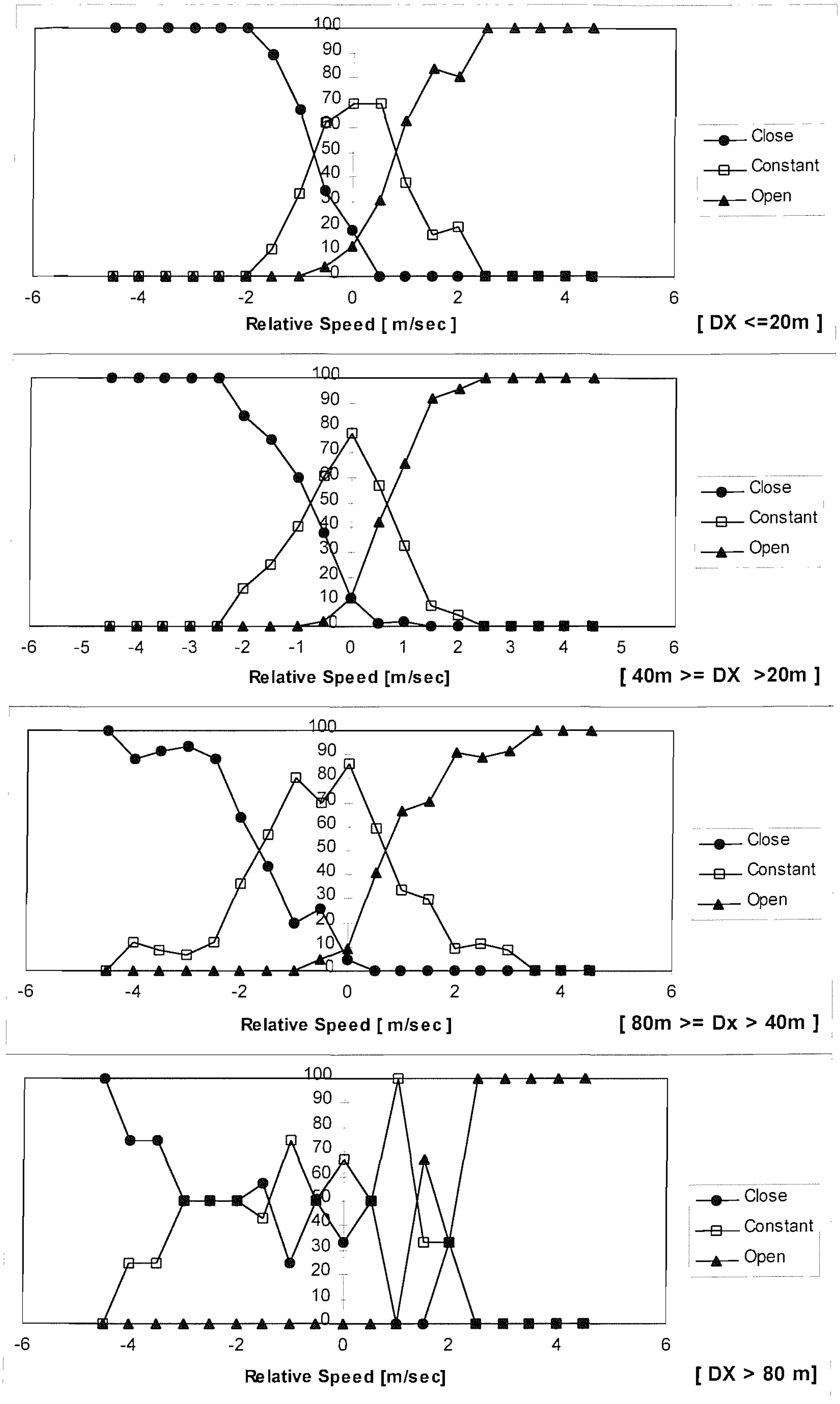


Figure 7.4: Probability of perception according to the relative speed [DX: the spacing distance]

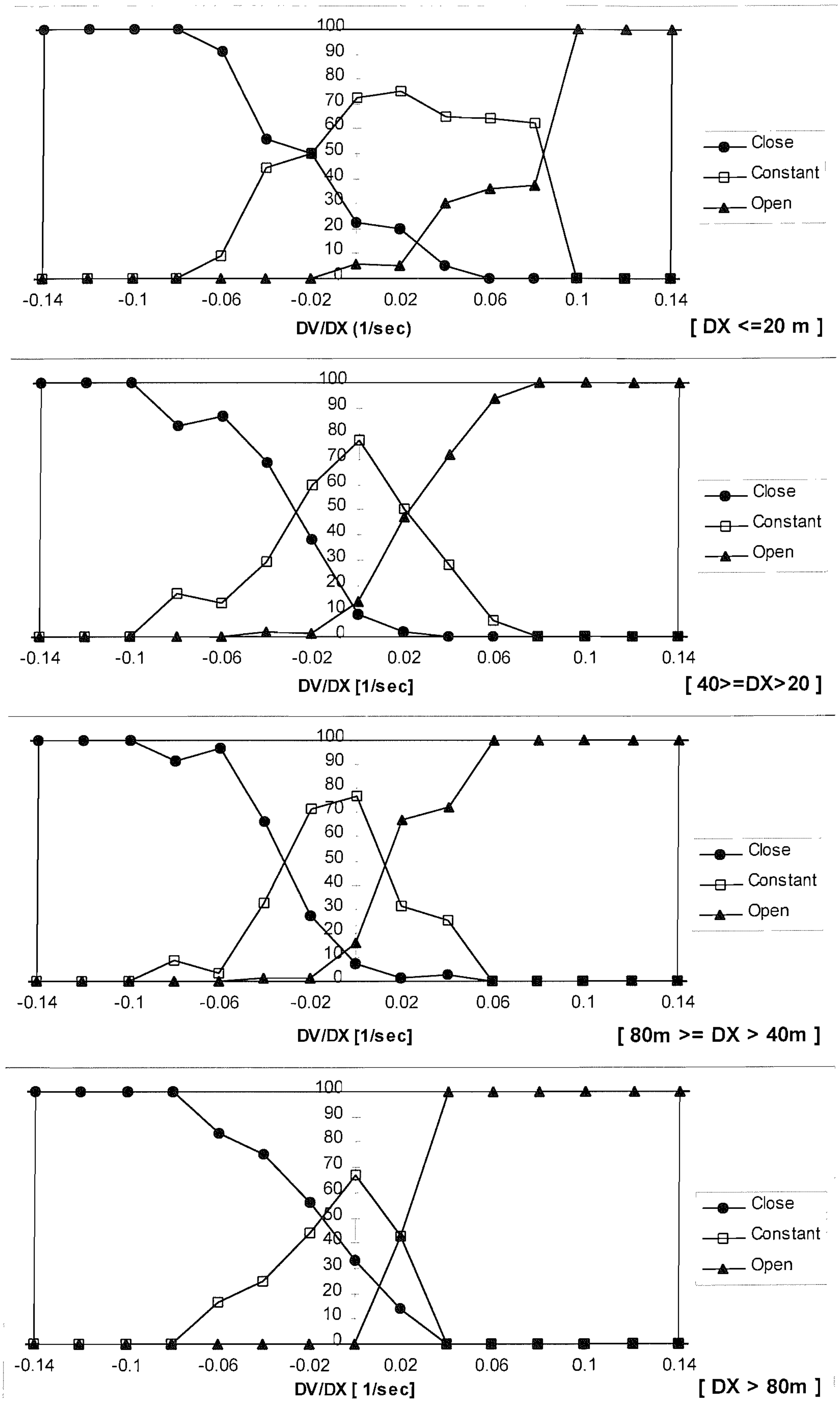


Figure 7.5: Probability of Perception according to the Ratio (DV/DX)

7.3.1.2 Modelling Perception Threshold:

The perception probability was calculated according to the percentage of (DV/DX) and regardless of the level of spacing distance. The result is shown in Figure 7.6. The phenomenon of the wrong decision was neutralised by transferring the percentage of the related probability percentage to the constant situation. A curve was fitted to each of the closing and opening probability sections. The probability for the constant decision can be calculated by cumulating the value from the curve lines to a hundred. The outcome is shown in Figure 7.7. The two equations for the curve lines are:

1- For Closing decision probability :

$$CL\% = -8136.9 * (DV/DX)^2 - 1721.7 * (DV/DX) + 7.9624 \quad .....7.10$$

While: $0.0 \geq (DV/DX) \geq -0.1$

2- For Opening decision probability :

$$OP\% = -8997.5 * (DV/DX)^2 + 1768.8 * (DV/DX) + 15.288 \quad .....7.11$$

While: $0.08 \geq (DV/DX) \geq 0.0$

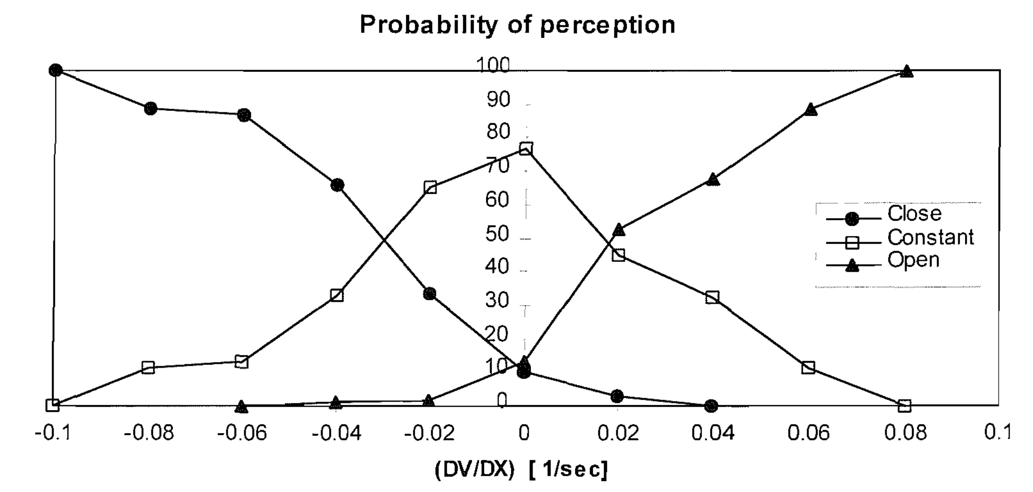


Figure 7.6: The Perception probability according to the ratio (DV/DX) regardless of spacing distance range.

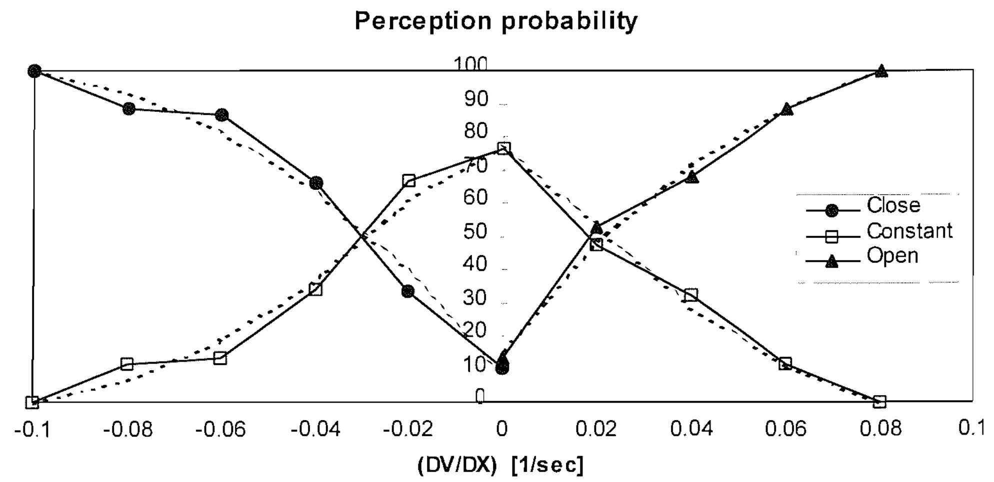


Figure 7.7: The Perception probability according to the ratio (DV/DX) after recovering incorrect and fitting a curve.

The following procedure is applied in the model to determine drivers' perception of threshold:

1- The value of (DV/DX) is calculated.

2- If (DV/DX) < -0.1 then the driver perception is : the relative distance is Closing.

3- If (DV/DX) > 0.08 then the driver perception is : the relative distance is Opening.

4- If (DV/DX) = 0.0 then the driver perception is : the relative distance is Constant.

5- If 0.08 ≥ (DV/DX) > 0.0 then the perception probability [OP%] will be calculated from the equation (7.10), then a random number [R] between [0 to 100] will be generated and compared with OP%: If R > OP% then the driver perception is : the relative distance is Constant, otherwise the relative distance is Opening.

6- If 0.0 > (DV/DX) ≥ -0.1 then the perception probability [CL%] will be calculated from the equation (7.11), then a random number [R] between [0 to 100] will be generated and compared with CL% : If R > CL% then the driver perception is : the relative distance is Constant, otherwise the relative distance is Closing.

7- If the driver decision was closing in the previous time step in the simulator he or she will not change there perception until the relative speed becomes positive or Zero. This is very important to avoid the model any arbitrary behaviour by the driver during the following process. The same is applied to the opening decision.

Figure 7.8: shows the flowchart for the driver decision to determine his perception.

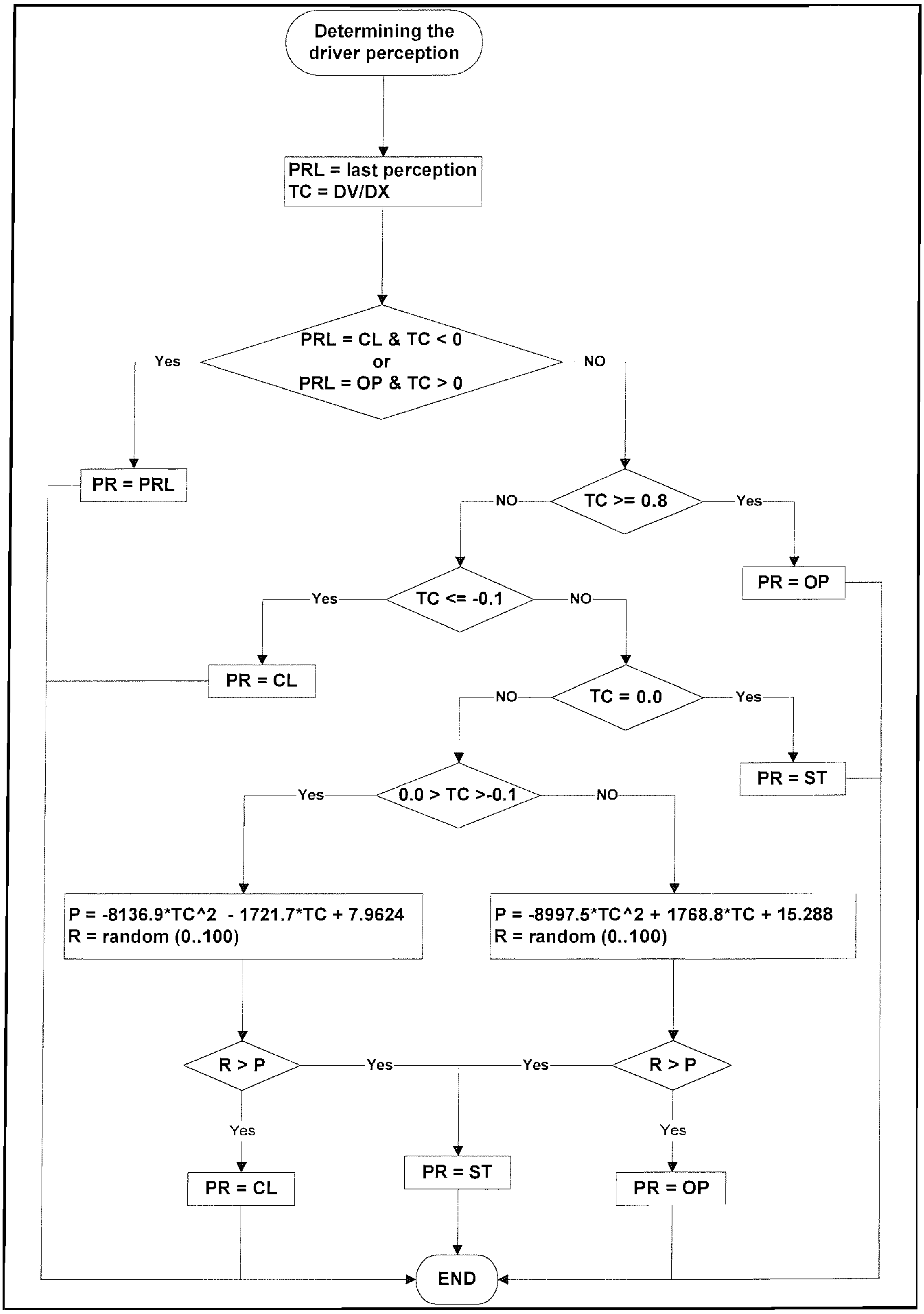


Figure 7.8: the flowchart of the perception decision of a driver in the model

7.3.2 Examination Of The Approach Process:

In this series of experiments the data was collected using three experienced drivers aged between 25 and 35, who drove the test vehicle on laps of a 21Km test course on the two lane dual carriageway A35, near Bournemouth in the U.K, for approximately 45 minutes. Each driver was instructed to drive in the near-side lane unless otherwise instructed, and that if their path were to become blocked by a slower vehicle, then they were to reduce speed and follow. The time of day chosen for the experiment (typically mid morning) was selected in order to minimise flow levels, and hence allow a clearer interaction between the vehicles (minimising the chance of a target vehicle changing lane, or other vehicles moving into the intervening gap during the approach process). (A familiarisation period of each subject with the vehicle of approximately 30 minutes was allowed, resulting from the 35km drive from experimental base to test site).

In total, 71 approach processes were observed and, in each case, a time series representing ground speed, relative distance DV and separation distance DX were isolated. Further examination of these traces was undertaken to ensure that the lead vehicle maintained an approximately constant speed during the approach process in order to ensure that the behaviour of the test driver was effected solely by changes in separation and relative speed caused by his actions alone. To this end, each trace was examined, and those subjectively judged as being 'unstable' (9 in total) were removed from the analysis. Typical cases from this examination are shown in Figures 7.9 and 7.10.

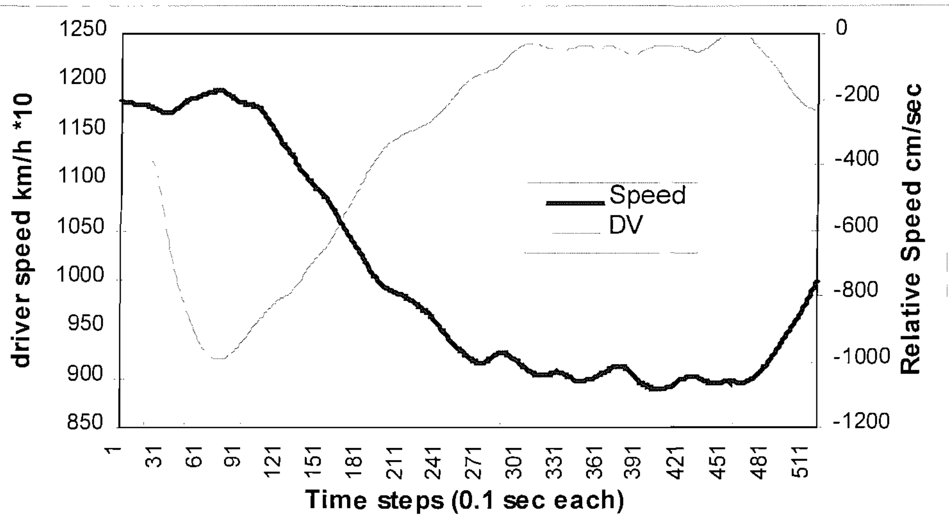


Figure 7.9: The stable condition where the lead vehicle has a constant speed

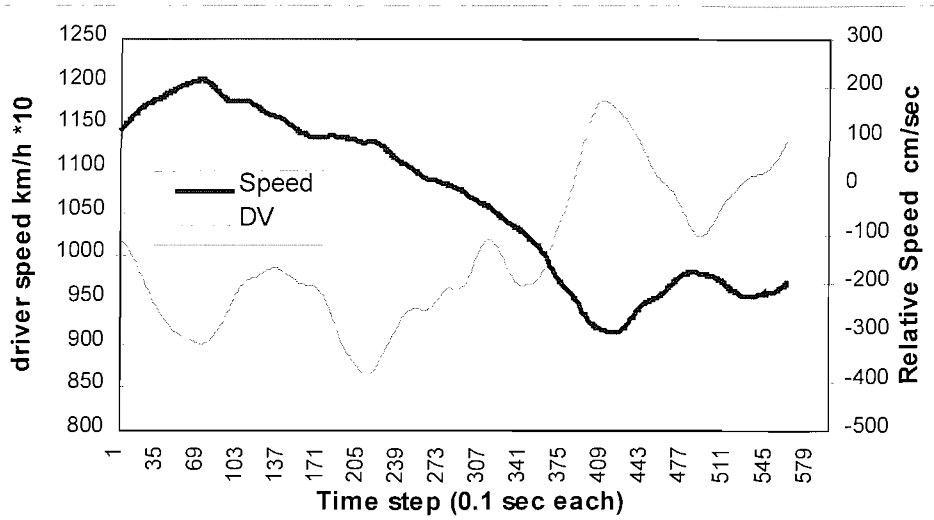


Figure 7.10: The unstable condition where the lead vehicle doesn't have a constant speed

7.3.2.1 When do drivers start their approach process.? :

From the time series, it is possible to isolate distinct 'Action Points' where the driver of the test vehicle starts to decelerate (identified as those points after which the response becomes continuously negative, hence eliminating 'throttle/brake switch over noise', (MONTROLL 1962)[63]) and these are presented in Figure 7.11. Several tests were undertaken to identify the proper model that may describe these action points, concerning the validity both overall, for all three subjects combined, and individually. They are:

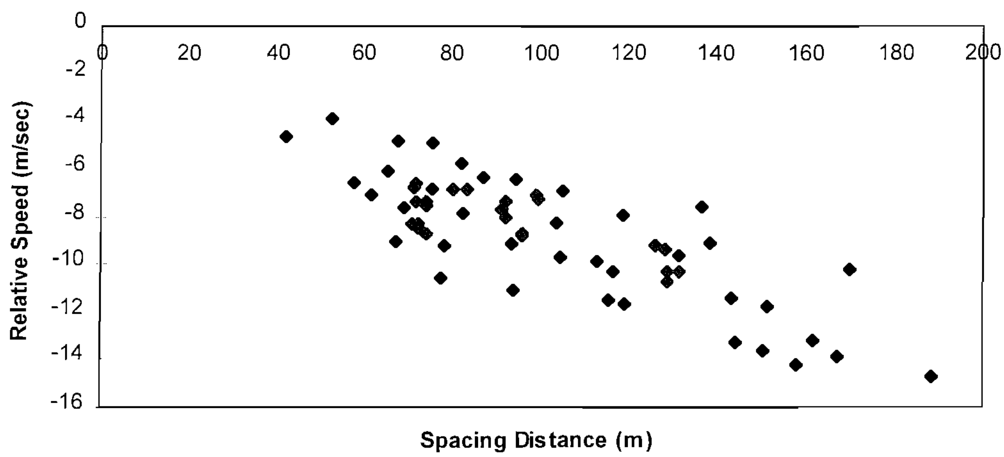


Figure 7.11: The relative distance vs. the relative speed for the starting points of the Approach Process [All subjects]

A- The relative speed DV and relative distance DX relationship: This was described by a linear relationship of the form: DV = a + b DX. The results show that subjects one and two had similar coefficients, whilst subject three had slightly different value. The value for a ,b and r^2 are shown in Table 7.1, and the fitted lines for DV vs DX are shown in Figure 7.12.

Subject	a	b	r^2
Subject 1	-3.19	-0.058	0.66
Subject 2	-3.15	-0.054	0.56
Subject 3	-1.79	-0.067	0.70
Overall	-2.71	-0.060	0.66

Table (7.1): Coefficient table for deceleration action points for DV vs DX

It is clear that there is a trend in the relationship between DV and DX, although the linear equation will not produce enough variation to model the action points.

B- The relative speed DV and. Headway DT relationship: This relationship shows the influence of speed on the driver's decision to start his approach process. The relationship is linear, although the coefficient values varied slightly between subjects. Table 7.2 present the coefficients of the equation DT = α + β DV , for every subject and overall of them.

Subject	α	β	r^2
Subject 1	0.27	-0.31	0.64
Subject 2	0.69	-0.30	0.47
Subject 3	0.87	-0.27	0.63
Overall	0.59	-0.29	0.58

Table 7.2: Coefficient table for deceleration action points for DT vs DX

As with the DV and DX relationship the variation between the action points is still too large to be modelled in linear equation (Figure 7.13).

From the above it is highly unlikely that an equation can be found that is able to simulate the action points where drivers start their approach process. The next two tests were undertaken to examine the optic flow ($d\theta/dt$) and the Time-To-Collision (TTC) to see if a better relationship can be found to predict the action points.

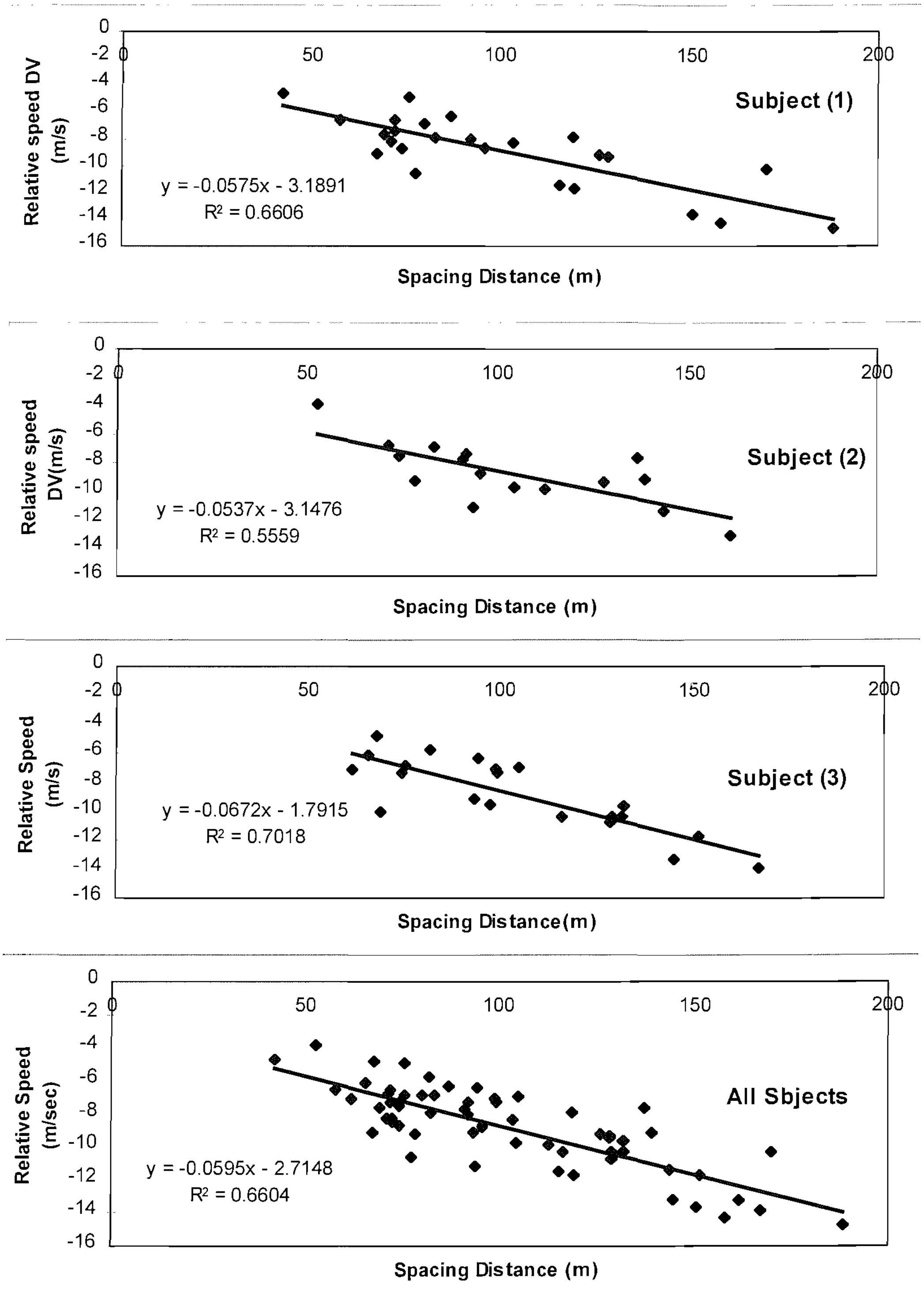


Figure7.12 : DV vs DX for the action points of the starting of the approach process, with linear regression

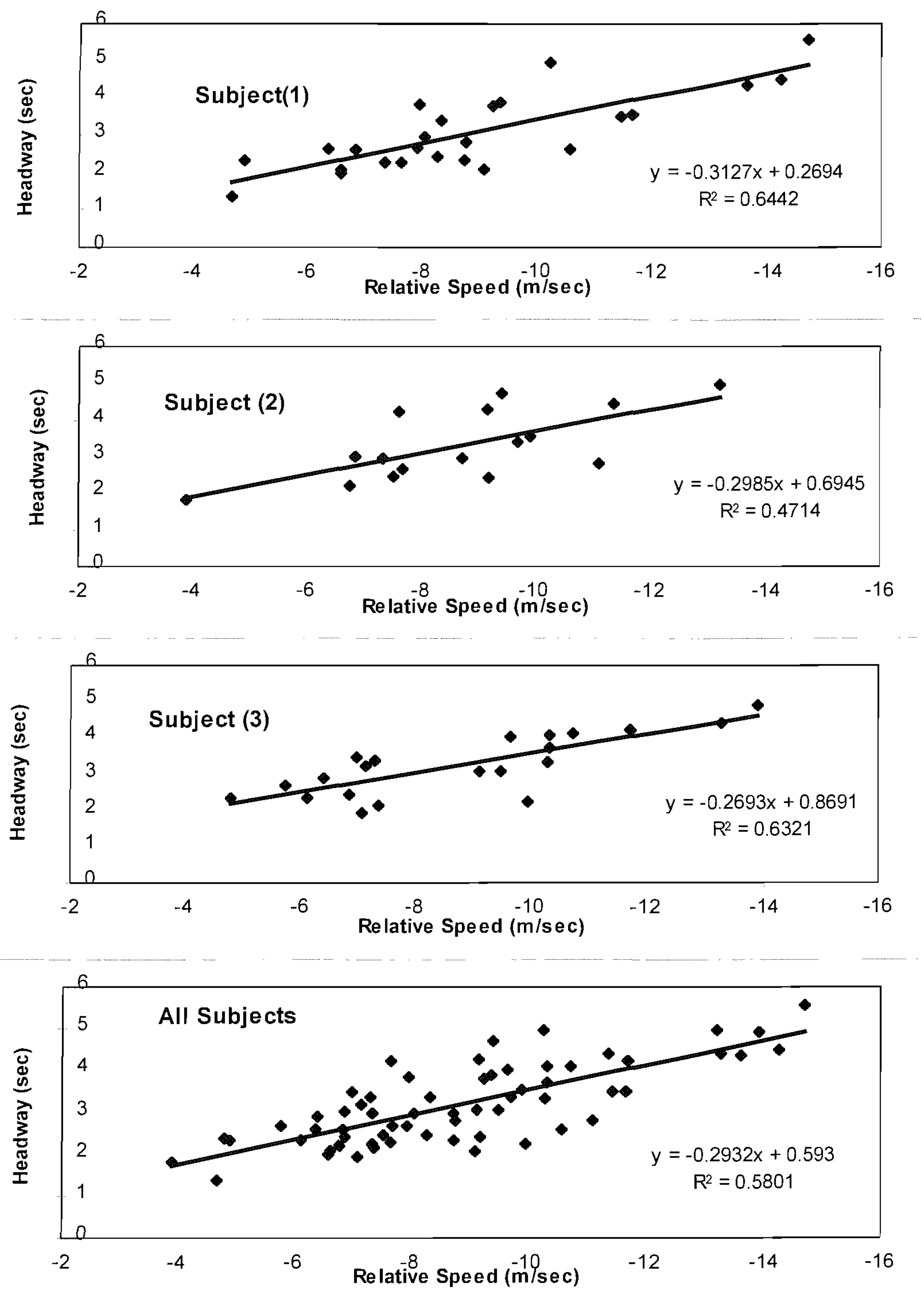


Figure 7.13: DV vs DT for the action points of the starting of the approach process, with linear regression

C- <u>Testing the Optic Flow (dθ/dt)</u>: The data shows a strong lognormal distribution with a group mean of 2.0 mr/s (KS=0.80, Table 7.3, Figure 7.14), closely matching values commonly suggested in the literature which are typically of the order of 2.4-2.7 mr/s (HOFFMAN and MORTIMER, 1994 and 1996)[58,64]. However it is possible that these results may be slightly affected by the methodology, ie. the subject may not be assessing the situation in terms of optic flow but may also be considering the overall dilation of the image over a period of time which is also known to be used as a perceptual evidence in the detection of relative motion. This is the so called 'Weber ratio' equal to ($dS/S = 0.12$) where S is spacing distance (HOFFMANN 1966)[20]. Due to the nature of the experiment, ie real road trial, the elimination of this factor through occlusion is clearly impossible.

Subject	mean (mr/s)	STD.(mr/s)	KS test
Overall	2.00	0.95	0.80
Subject 1	2.22	1.16	0.96
Subject 2	1.84	0.72	0.91
Subject 3	1.86	0.82	0.45

Table 7.3: Statistics for the lognormal distribution of (dθ/dt)

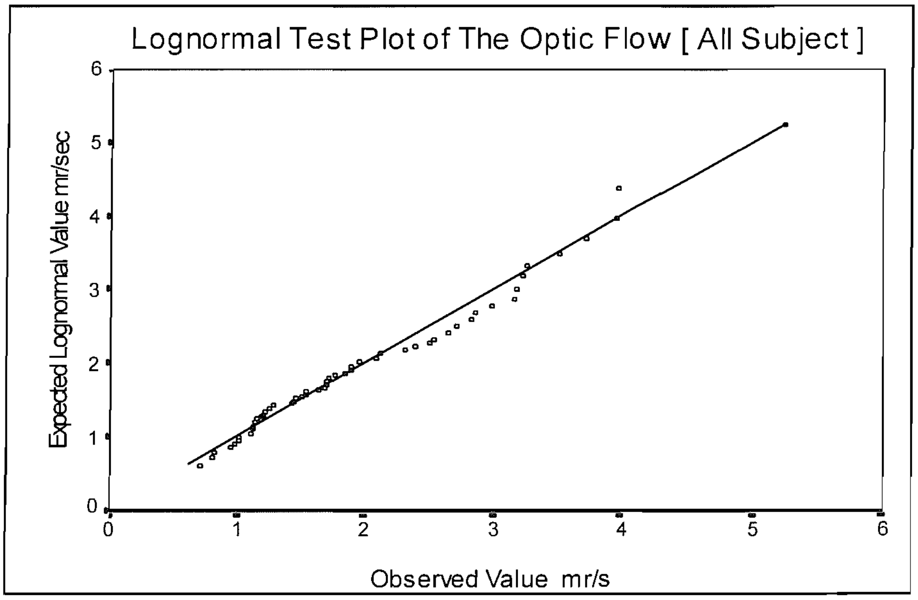


Figure 7.14: the lognormal distribution fitting for the Optic Flow

It is interesting to compare these values with those predicted by LEUTZBACH and WEIDEMANN (LEUTZBACH & WEIDEMANN 1986)[21]. They implemented a normal distribution factor in the MISSION simulation model to produce just such a spread between a minimum and maximum threshold corresponding to 0.3 and 3.2 mr/s with an average 1.4 mr/s. This is illustrated in Figure 7.15, along with a new minimum threshold later suggested by REITER of 0.8 mr/s (REITER 1994)[65].

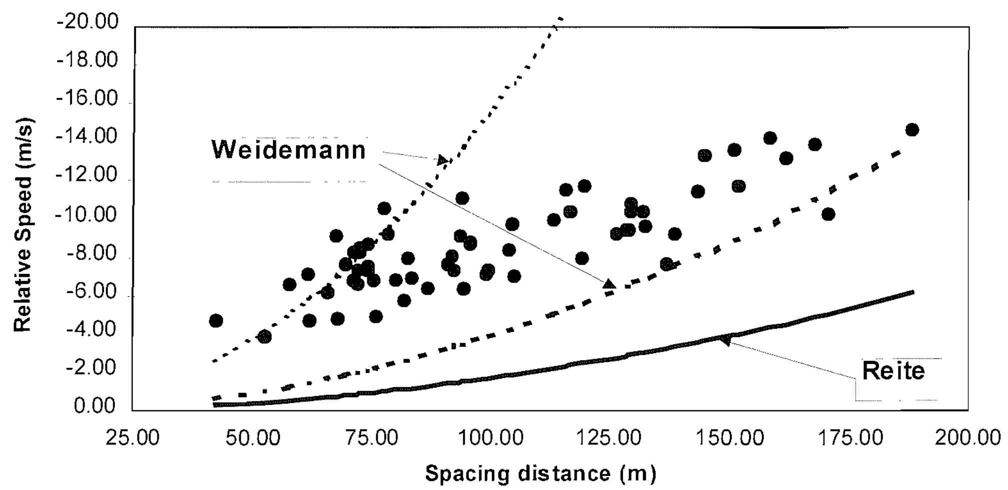


Figure 7.15: The approaching process action points and the thresholds suggested for the MISSION model

An examination of the dependence of the optic flow on DX reveals that there is a strong reciprocal relationship between the threshold used and the inter-vehicle spacing. This indicates that a more appropriate threshold may be one related to TTC. The relationship between (dθ/dt) and DX is shown in Figure 7.16.

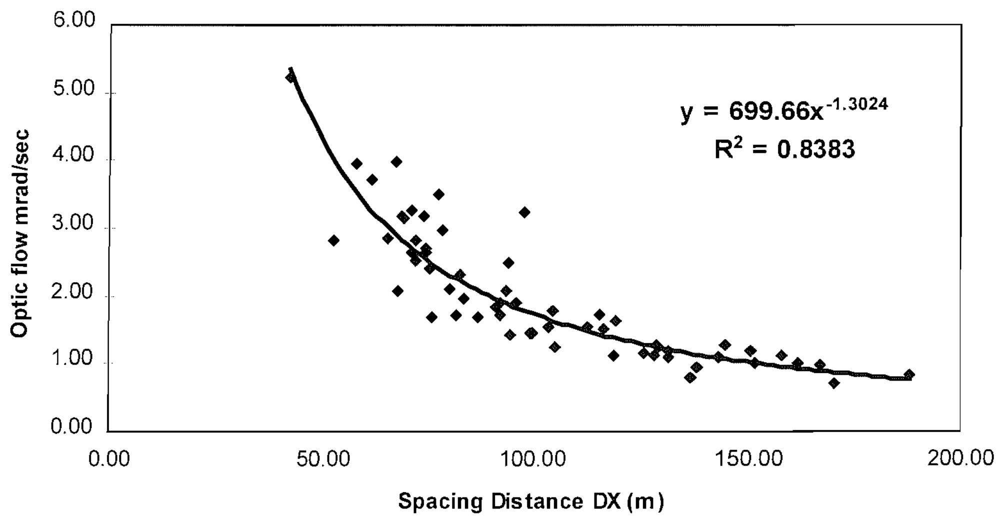


Figure 7.16: The Optic Flow vs The Spacing Distance for the starting approach process action points

D- Testing The Time To Collision TTC_{start} : From the results in test A (a linear relationship between DX and DV, see Figure 7.12), it is tempting to hypothesise that the TTC would give the best results in modelling these action points [TTC = - DX/DV]. The analysis of TTC_{start} values reveals that they are normally distributed around a group mean of 11.70 sec. (standard deviation 2.29 sec.) to a degree of significance (through a Kolmogorov-Smirnov test [KS]) of 0.93 (Table 7.4, Figure 7.17).

Subject	mean (sec.)	StDev. (sec.	KS
Overall	11.70	2.29	0.93
Subject 1	11.20	2.48	0.88
Subject 2	12.02	2.38	0.84
Subject 3	12.00	2.03	0.88

Table 7.4: Statistics for the normal distribution of TTC_{start}

An examination of the dependence on DX for the TTC_{start} shows that the data is more scattered than that found for the DX vs (dθ/dt) relationship (Figure 7.18). This finding gives more evidence that TTC_{start} distribution is better than the optic flow in modelling the starting points of drivers approach process.

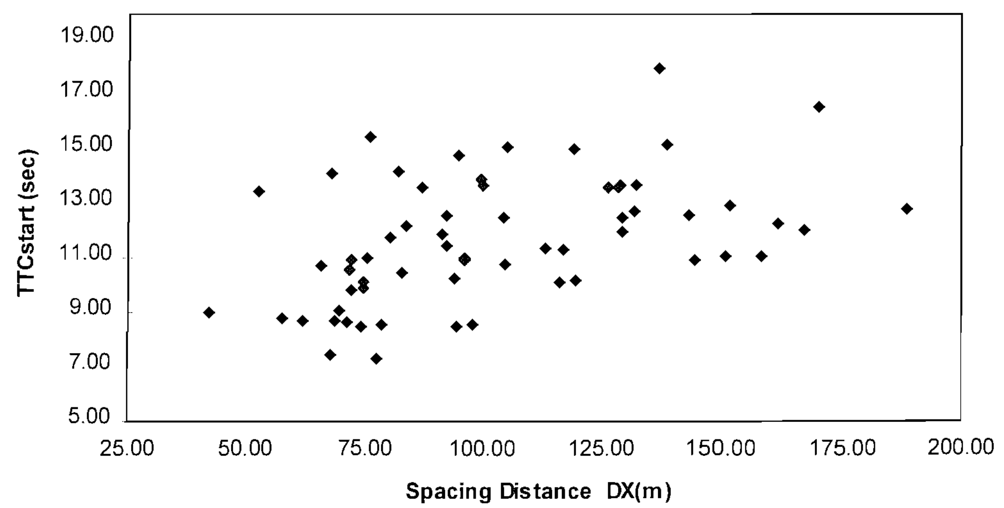


Figure 7.18: The TTC_{start} vs spacing distance for the starting approach process action points

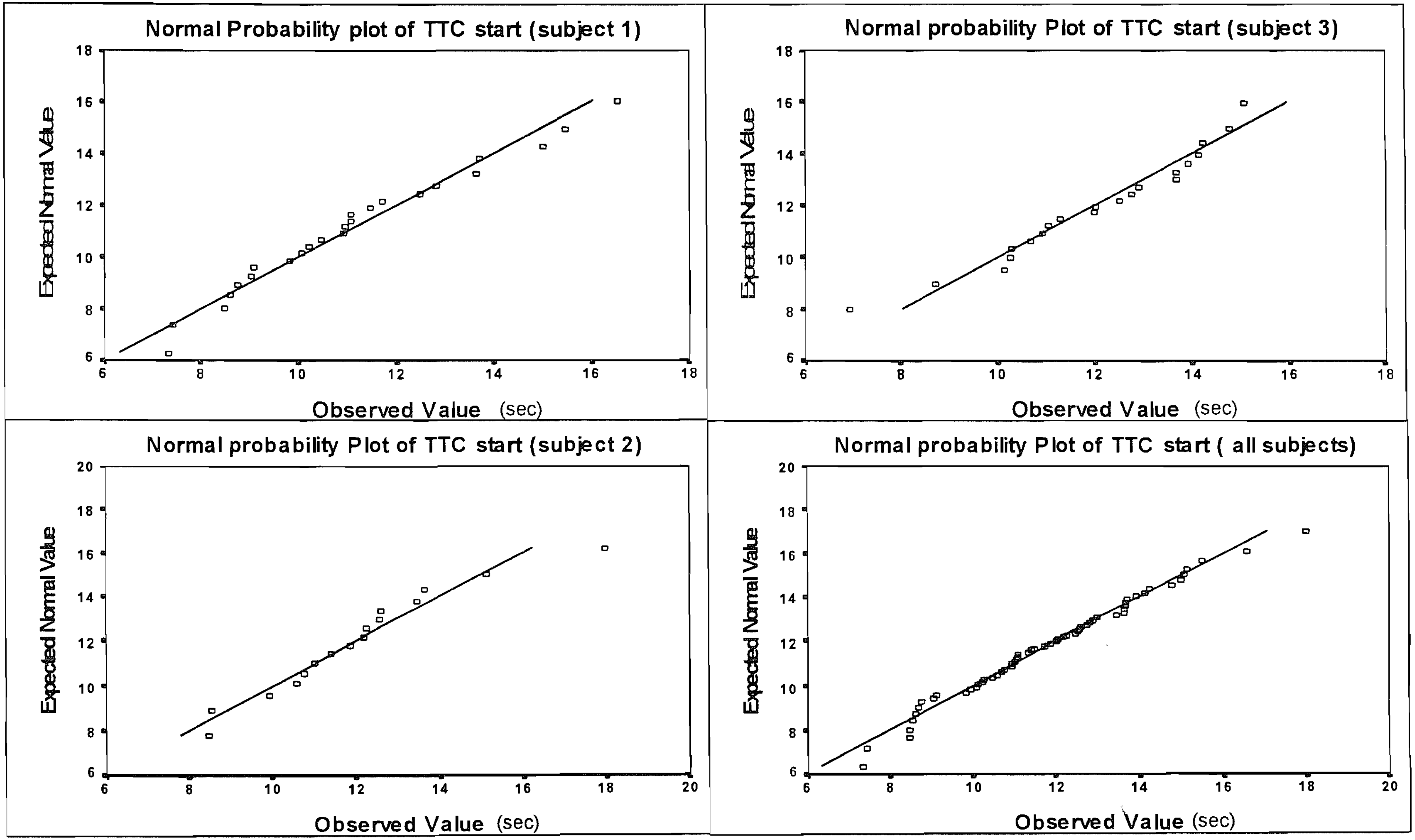


Figure 7.17 : The Normal Probability plots for TTCstart

7.3.2.2.How do drivers control there approach.?

The second issue to be addressed is once the approach process has begun, how do drivers affect their decelerations in order to approach smoothly lead vehicles. The initial step taken to examine this process was to plot the deceleration time profiles in dimensionless co-ordinates following (SPURR 1969)[60]. In contrast to SPURR conclusions, a great deal of variation in the observed profiles was found and, in some cases, the first peak of the profile is not the maximum value, with that occurring at a second or sometimes a third peak. The degree of variation within the profiles observed is high and at times may change from deceleration to acceleration (Figures 7.19 and 7.20).

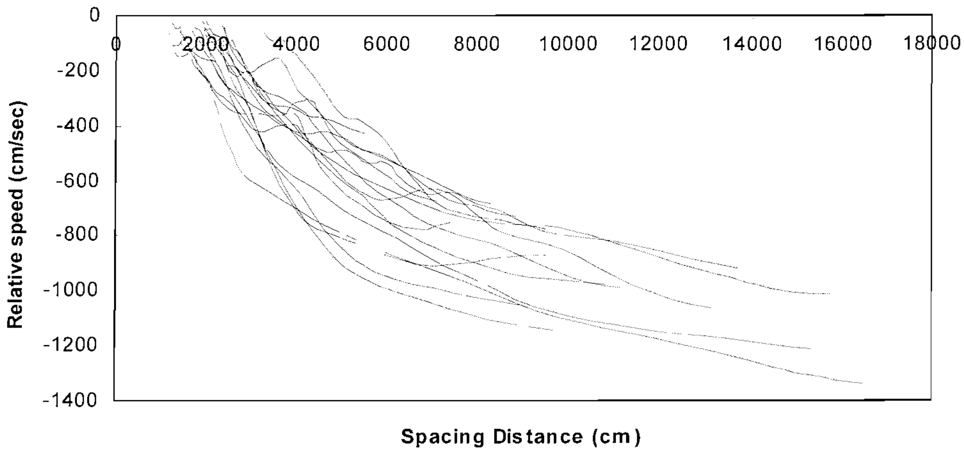


Figure 7.19: The Approaches trails for subject (2)

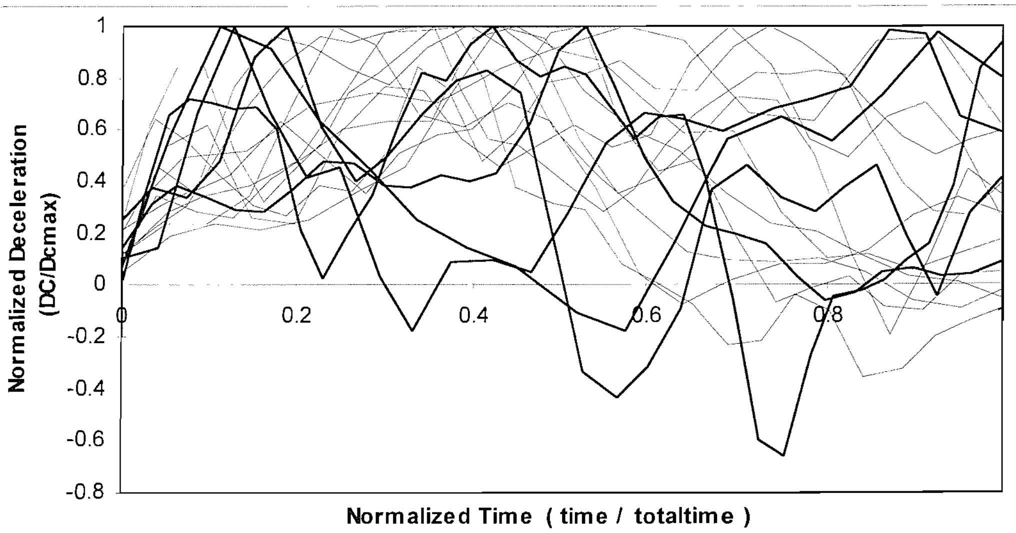


Figure 7.20: The normalised Deceleration with relation to the time for all the approaches trails (Subject 2)

On examining the values of TTC_{min}, which is the minimum measured value for TTC during the approach process, a distinct lognormal grouping was found with an overall mean of 8.41 sec. (KS = 0.63). (Table 7.5 and Figure 7.21).

Subject	mean	STD.	KS
Overall	8.41	2.01	0.63
Subject 1	7.72	1.57	0.84
Subject 2	7.82	1.89	0.95
Subject 3	9.40	2.55	0.95

Table 7.5: Statistics for the distribution of TTC_{min}.

Also, the value taken by TTC_{min} was seen to be related to the maximal deceleration (DC_{max}) observed during this time, where a lognormal distribution is again observed with an overall mean of -0.87 m/s^2 (KS test = 0.64). (Table 7.6 and Figure 7.22).

Subject	mean	STD.	KS
Overall	-0.87	0.34	0.64
Subject 1	-0.89	0.32	0.55
Subject 2	-0.82	0.34	0.71
Subject 3	-0.88	0.37	0.99

Table 7.6: Statistics for the distribution of DC_{max}.

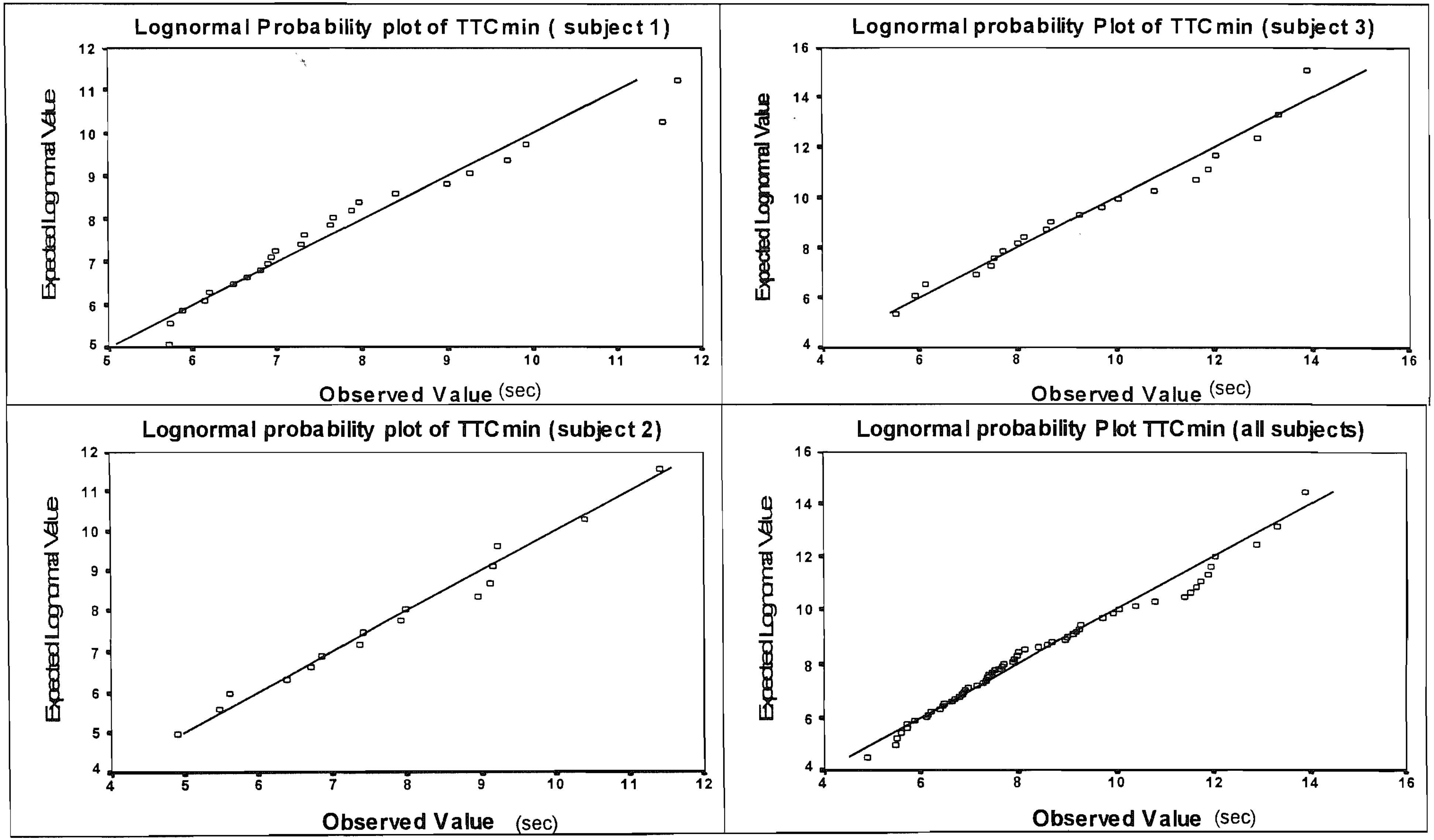


Figure 7.21 : The LogNormal Probability plots for TTCmin

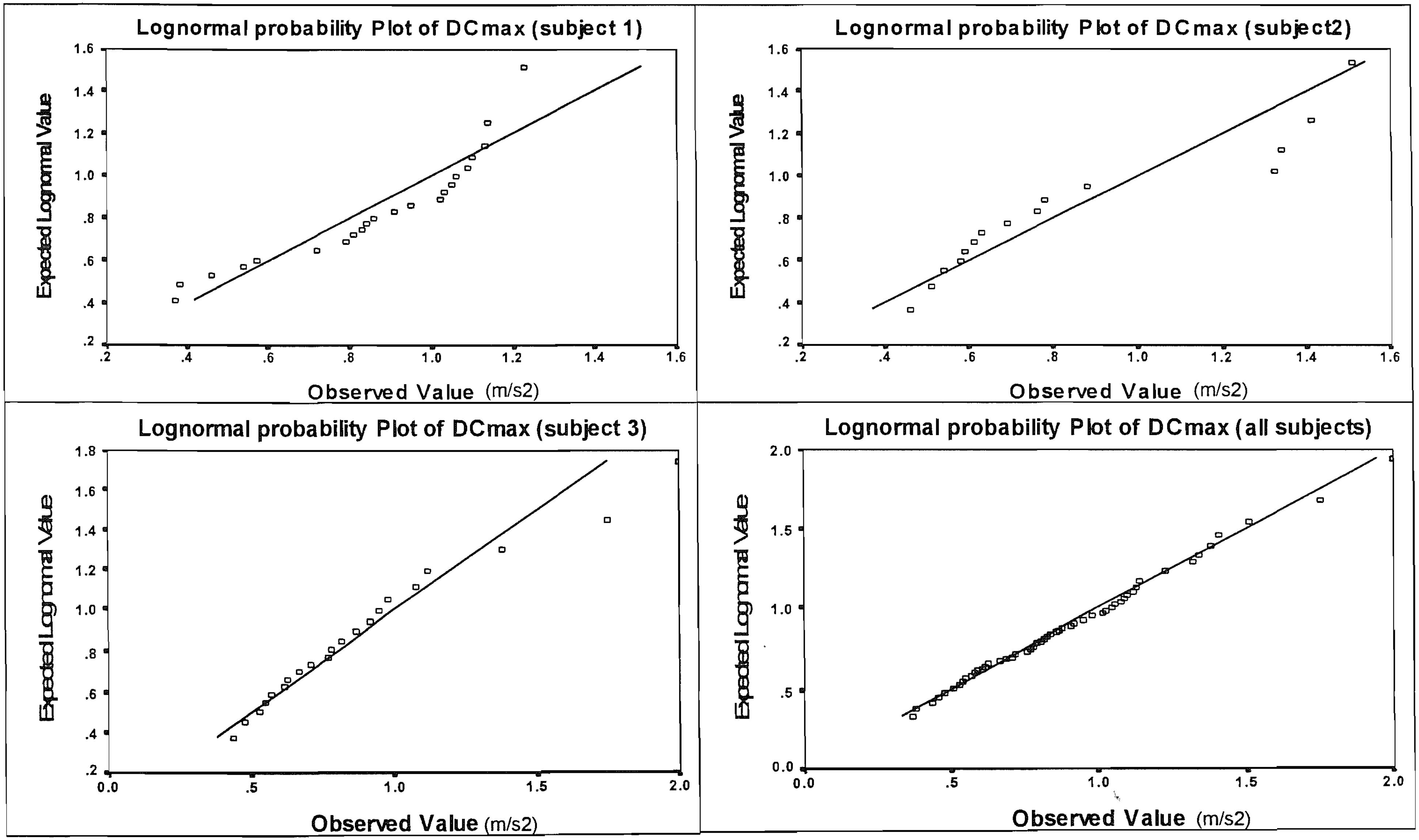


Figure 7.22 : The LogNormal Probability plots for DCmax

7.3.2.3. Discussion:

1- Clearly, the experiment itself is not without weaknesses, over and above the issue of sample size. Also, it is possible to question the derivation of the 'start points' as these have been subjectively extracted from the time series traces. (A more exacting method could involve direct monitoring of brake/accelerator displacement and pressure). However, the experiment has significantly increased our understanding of the approach process.

2- From the above analysis it is possible to come to some clear conclusions regarding this phase of driver behaviour and show, as suggested in previous studies, that it is possible to formulate a good model of the approach process characterised by:

   a- A set TTC_{start}, which varies by event and may be drawn from a normal distribution of mean 11.7 sec.

   b- A particular TTC_{min}, again varying by event and drawn from a lognormal distribution with a mean of 8.4 sec.

3- From the literature it was found that drivers tend to underestimate TTC. The model should be able to simulate this shortcoming which influences the deceleration level. It was decided that the HOFFMANN & MORTIMER equation (HOFFMANN & MORTIMER 1994)[58] will be used. The reason for this is that they collected their data in a dynamic way, which most closely matches the real situation:

$$EST = -0.18 + 0.767 * TTC \qquad r^2 = 0.94 \qquad \text{....7.12}$$

Where: EST represents the mean estimate of TTC.

Also, in all the experiments that has been undertaken in this field, TTC was more than one second, and the relative distance was greater than 33 meters. Therefore, to simplify the calculation and to avoid any contradiction in the results. The model assumes: [The EST will equal TTC if TTC is less than 1 sec.]

7.4. MODELING THE APPROACH PROCESS:

Before a procedure is suggested for the approach process model, other factors need to be clarified especially when the lead vehicle does not have a constant speed. These factors can be summarised as : the perceived deceleration of the lead vehicle, the maximum acceptable deceleration and the deceleration level to sustain a constant TTC.

7.4.1 The Perceived Deceleration or Acceleration of the lead vehicle :

The driver will perceive the deceleration / acceleration of the lead vehicle, by using the estimated time to collision at the moment T , and the change in the estimated time to collision during the epoch t. There are two ways to calculate the Perceive Deceleration/Acceleration:

<u>The First Method:</u>

Supposing that:

Dx : Is the relative distance at moment T + t .
Dx_0 : Is the relative distance at moment T .
Dv : Is the relative Speed at moment T + t .
Dv_0 : Is the relative Speed at moment T .
Dc : Is the relative deceleration at moment T ($Dc = Dc_L - Dc_F$).
Tc : Is the estimated time to collision at the moment T + t (Tc = -Dx / Dv).
Tc_0 : Is the estimated time to collision at the moment T ($Tc = -Dx_0 / Dv_0$).

And if we assumed that Dc is constant during the epoch (t) then :

$$Dx = Dx_0 + Dv_0 * t + 0.5 * (Dv - Dv_0) * t \quad \Rightarrow$$

$$Dx = Dx_0 + 0.5 * (Dv + Dv_0) * t \quad \Rightarrow$$

$$\frac{Dx}{Dv} = \frac{Dx_0}{Dv} + 0.5*(1+\frac{Dv_0}{Dv})*t \quad \Rightarrow$$

$$-Tc = -\frac{Dv_0}{Dv}*Tc_0 + 0.5*(1+\frac{Dv_0}{Dv})*t \quad \Rightarrow$$

$$\boxed{Tc = (Tc_0 - 0.5*t)*\frac{Dv_0}{Dv} - 0.5*t} \qquad ......7.13$$

$$\frac{Dv}{Dv_0} = \frac{Tc_0 - 0.5*t}{Tc + 0.5*t} \quad \Rightarrow \quad \frac{Dv_0 + Dc*t}{Dv_0} = \frac{Tc_0 - 0.5*t}{Tc + 0.5*t} \quad \Rightarrow \quad \frac{Dc*t}{Dv_0} = \frac{Tc_0 - Tc - t}{Tc + 0.5*t}$$

$$\Delta Tc = Tc - Tc_0 \quad \Rightarrow \quad Dc = \frac{-\Delta Tc - t}{Tc_0 + \Delta Tc + 0.5*t} * \frac{Dv_0}{t} \quad \Rightarrow$$

$$\boxed{Dc_L = \frac{-(\Delta Tc + t)}{Tc_0 + \Delta Tc + 0.5*t} * \frac{Dv_0}{t} + Dc_F} \qquad ....7.14$$

Equation 7.14 gives the perceived deceleration or acceleration of the lead vehicle. Obviously as the Time-To-Collision is estimated by the driver, he or she is estimating the deceleration of the vehicle ahead.

The Second Method :

The time to collision at any moment can be calculated by :

$$Tc = -\frac{DX}{DV} \qquad \text{......7.15}$$

While : $DX = X_L - X_F - Ax$ & $DV = V_L - V_F$ & $DV < 0$.

By differentiating equation (7.15) ⇒

$$Tc' = \frac{-DV^2 + DX * DC}{DV^2} \qquad \text{......7.16}$$

While :

Dc : Is the relative deceleration at moment ($Dc = Dc_L - Dc_F$).

Tc' is the change rate in the amount of the Tc in the unit of time.

From equation (7.16) :

$$Tc' = -1 - \frac{DC}{DV} * Tc \quad \Rightarrow \quad DC = -(Tc' + 1) * \frac{DV}{Tc} \quad \Rightarrow$$

$$\boxed{DC_L = DC_F - \frac{DV}{Tc} * (Tc' + 1)} \qquad \text{......7.17}$$

Equation 7.17 gives the perceived deceleration or acceleration of the lead vehicle.

Discussion:

1- Equations 7.14 or 7.17 will give the same result.

2- Because drivers underestimate the time to collision, it can be argued that they overestimate the relative speed. The changing in the values of Dv_0 , Tc_0 and ΔTc due to the under-estimation of time to collision, makes the equations 7.14 and 7.17 give an underestimates of the deceleration of the lead vehicle.

3- One can argue that drivers will not respond to every deceleration or acceleration of the lead vehicle. Thus, there is a need to put a limit on the perception of deceleration or acceleration. However, although it is very difficult to measure this limit, common sense shows us that the greater the relative distance the greater the limit of perception and vice versa. The model assumes that the limit is $-0.5 m/sec^2$ for deceleration and $+0.5\ m/sec^2$ for acceleration. These were adopted after examining the deceleration and acceleration variation in calm close following, which will be discussed later in the next Chapter of this thesis.

7.4.2 The Maximum Acceptable Deceleration:

When the lead vehicle decelerates the following driver will have a safe margin of time ΔT:

$$\Delta T = (DX + DV*(\text{reaction time}) + 0.5*DC_L *(\text{reaction time})^2 - AX) / V_F$$

where :

DX : is the relative distance m.
DV : is the relative speed m/sec.
AX : is the safe margin when the driver comes to stop.
V_F : The speed of the following distance.
DC_L : The deceleration of the lead vehicle.

This time would give the driver an indication about how much time he or she has to decelerate, ie. the suitable deceleration Dc_F to stop behind a lead vehicle stopping at Dc_L :

$$-\frac{V_F^2}{2 * Dc_F} = V_F * \Delta T - \frac{V_L^2}{2 * Dc_L} \qquad \Rightarrow$$

$$\boxed{Dc_{max} = \frac{Dc_L * V_F^2}{V_L^2 - 2 * Dc_L * V_F * \Delta T}} \qquad ......7.18$$

The model will use a reaction time of 0.5 second.

7.4.3 The Deceleration Level That Sustains a Constant TTC:

As the experiment's results showed that drivers do usually sustain the TTC level after reaching its minimum level, the need arises to calculate the deceleration value that keeps a constant level of TTC over an epoch time (T). It is already known that $TTC_0 = TTC_T$, therefore:

$$TTC_0 = TTC_T \Rightarrow \frac{DX_0}{DV_0} = \frac{DX_T}{DV_T} \Rightarrow DX_T = \frac{DV_T}{DV_0} * DX_0 \qquad .....7.19$$

Where:

DX_T & DV_T : The spacing distance and relative speed after epoch (T).

DX_0 & DV_0: The starting spacing distance and relative speed.

From the motion equations:

$$DV_T = DV_0 + DC_0 * T \qquad .....7.20$$

$$DC_0 = \frac{DV_T^2 - DV_0^2}{2*(DX_T - DX_0)} \qquad .....7.21$$

Where: DC_0 : The staring relative deceleration.

Equation 7.19 can be modified to:

$$DX_T = \left(1 + \frac{DC_0 * T}{DV_0}\right) * DX_0 \qquad .....7.22$$

By replacing the value of DV_T (equation 7.20) and DX_T (equation 7.22) in the equation 7.21 gives the equation:

$$DC_0 = \frac{[DV_0 + DC_0 * T]^2 - DV_0^2}{2*\left(DC_0 * T * \frac{DX_0}{DV_0}\right)} \qquad .....7.23$$

Which can be modified to

$$DC_0 = \frac{-2*DV_0}{2*TTC_0 + T} \qquad .....7.24$$

Finally:

$$\boxed{DC_F = DC_L + \frac{2*DV_0}{2*TTC_0 + T}} \qquad .....7.25$$

Equation 7.25 calculates the deceleration level that is needed to sustain a constant TTC.

7.4.4 How Does The Model Work? :

- After a driver cross the perception threshold, two random numbers will be generated. The first has a normal distribution [mean =11.7026 sec and StdDev = 2.292 sec] which defines the TTC_{start} at which the approach process starts. The second has a lognormal Distribution [mean = 8.41sec and StdDev=2.01 sec] which determines the TTC_{min} accepted to the driver. However, TTC_{min} should be less than or equal to (TTC_{start} -2 sec), and this will help to replicate the graduate increase in the driver deceleration. Before, the driver crosses the TTC_{start} , the acceleration will not be more than NULL [the minimum controllable acceleration . The process is out lines in Figure 7.25.

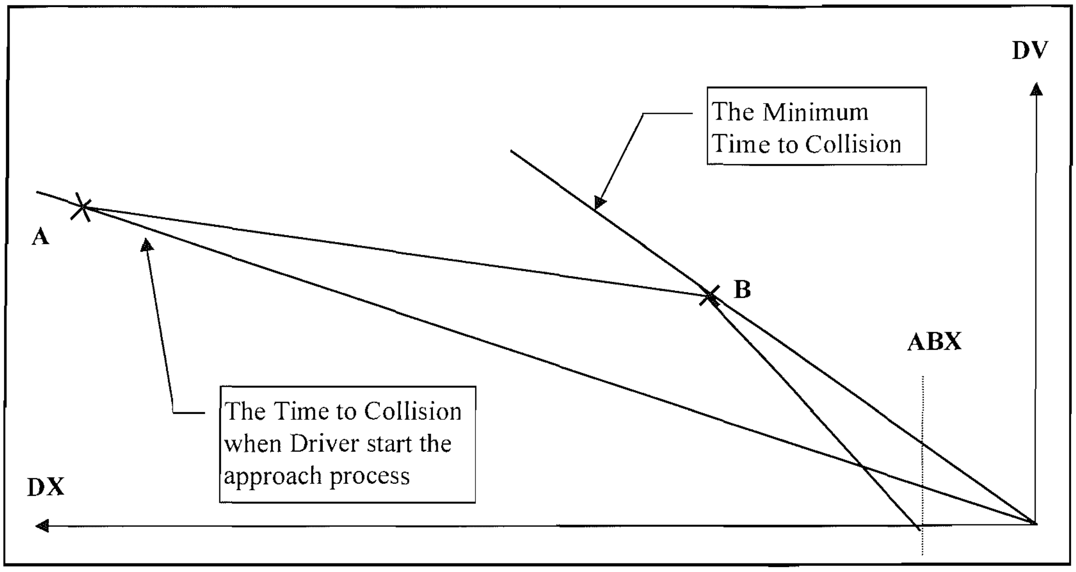


Figure 7.25: a simple graph for the idea of the Model

- As soon as the driver crosses the value of TTC_{start} , three steps will follow:

  (i)- TTC will be estimated according to the equation

  $$EST = -0.18 + 0.767 * TTC$$

  (ii)- Then, the estimated relative speed from the equation: $EDV = DX / EST$

  (iii)- The deceleration would be calculated according to the equation :

  $$DC = F1 * EDV / (EST - RT) + DC_L$$

  $$F1 = F2*EXP(((TTC - TTC_{start}) / (TTC_{min} - TTC_{start})) - 1)$$

  Where :

  RT : The reaction time in perceiving DV and DX [0.5 sec]

  DC_L : The perceived Deceleration from the lead vehicle.

  F2 : Is a calibrated constant. F2 was found to have a value of (0.45).

  F1 : a factor which has an exponential relationship with the value of TTC, TTC_{start} and TTC_{min}

- Once the TTC becomes less or equal to TTC_{min} the model will calculate the deceleration from equation 7.25. However, in order to let the model achieves the ABX following distance the value of DC_F will be compared with the value from equation 7.26 and adopt the harder deceleration.

$$DC = DC_L - \frac{EDV_0^2}{2*(DX_0 - ABX)}$$ .....(7.26)

- Always, the model checks the value of DC with the value of Dc_{max} (equation 7.18) when the lead vehicle is decelerating.

7.4.5 Testing The Model:

A test program was developed in Turbo Pascal to simulate the approach process between two vehicles. The leader always has a constant speed, while the follower has a higher starting relative speed, which was chosen randomly between DV equals 2 to 14 m/sec. The starting relative distance was assumed to be (DX: 200 m). The flowchart for the approach process model is shown in Figure 7.26. The program is listed in Appendix (2).

The test program was run 100 times, and the values of TTC_{start} , TTC_{min} and DC were then compared with the real data. Table 7.7 shows the mean and Std for TTC_{start} , TTC_{min} and DC from the Model and real data. The Kolmogorov-Smironov test shows that the model is not contradicting the real data in Distribution terms.

	Real DATA		Model DATA		K S Test
	Mean	Std	Mean	Std	
TTC_{start}	11.702 sec	2.29	11.75 sec	1.88 sec	0.704
TTC_{min}	8.41 sec	2.1	8.73 sec	1.54 sec	0.419
DC	-0.86 m/sec^2	0.34	-0.89 m/sec^2	0.32 m/sec^2	0.892

Table 7.7: Comparing between the real data and the output from the model.

Outputs from the model for each TTC_{start} , TTC_{min} and DC_{max} are shown in Figures 7.27, 7.28 and 7.29. The DV-DX relationship for each of the model and the real data is shown in Figure 7.30.

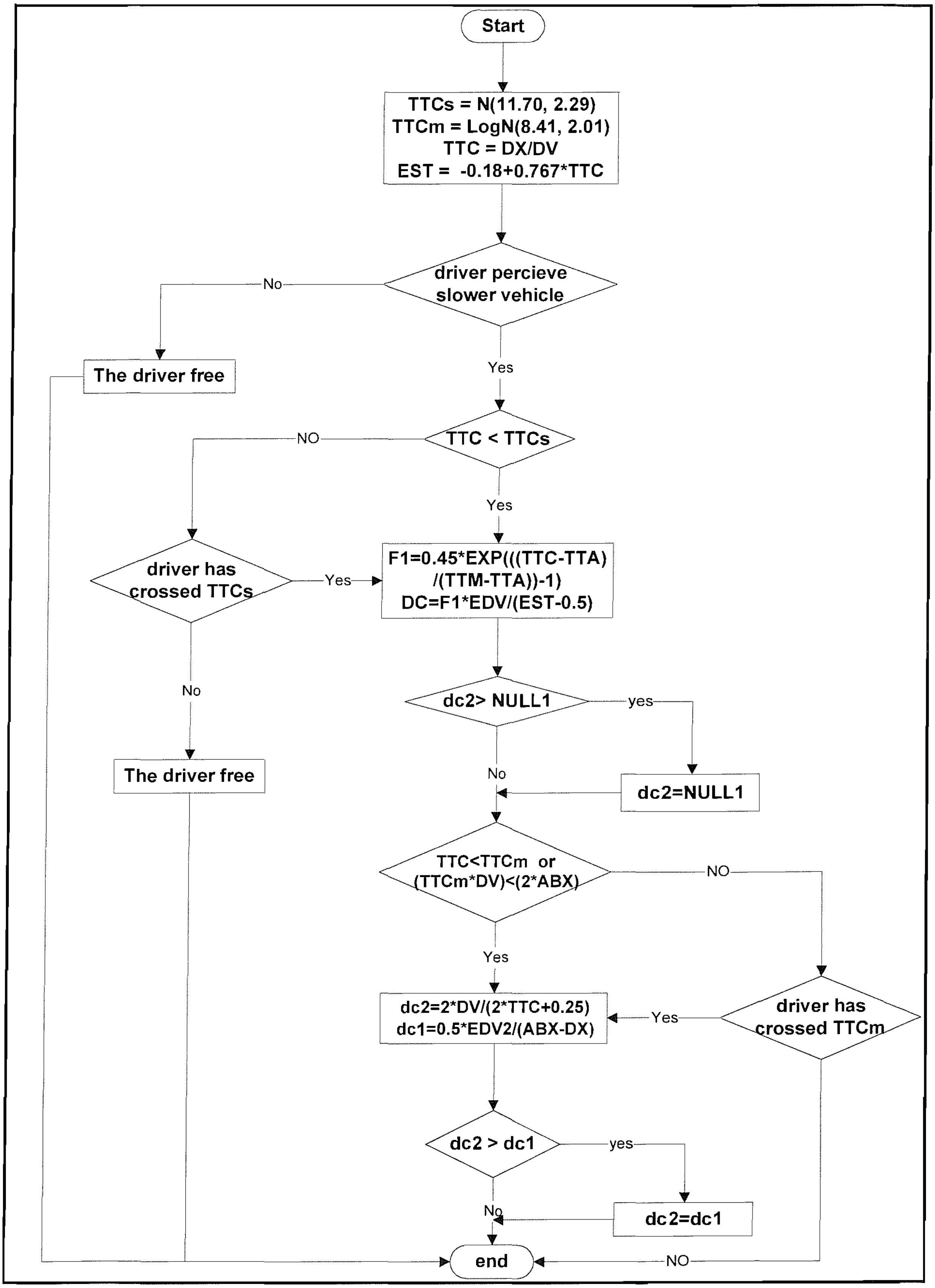


Figure 7.26: The Flowchart for the approach process model

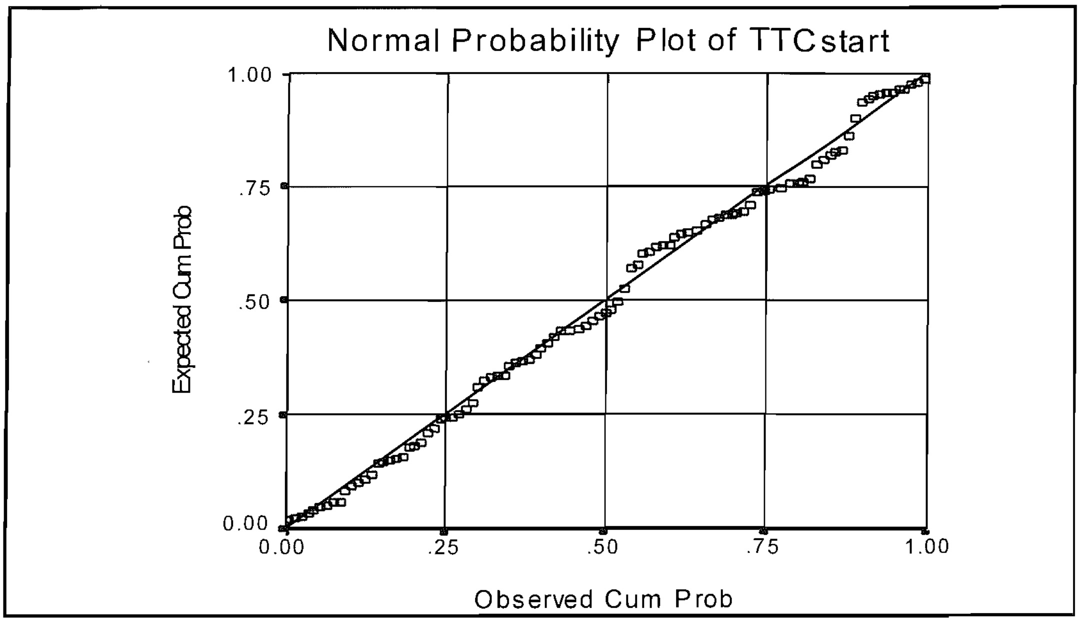


Figure 7.27: The distribution of TTC_{start} as an output from the model.

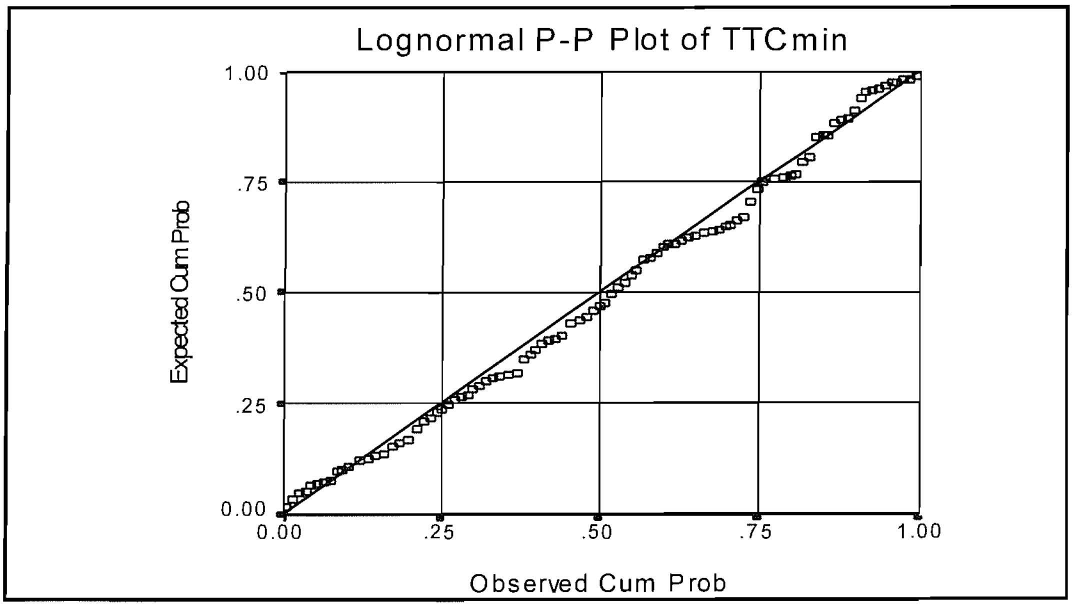


Figure 7.28: The distribution of TTC_{min} as an output from the model.

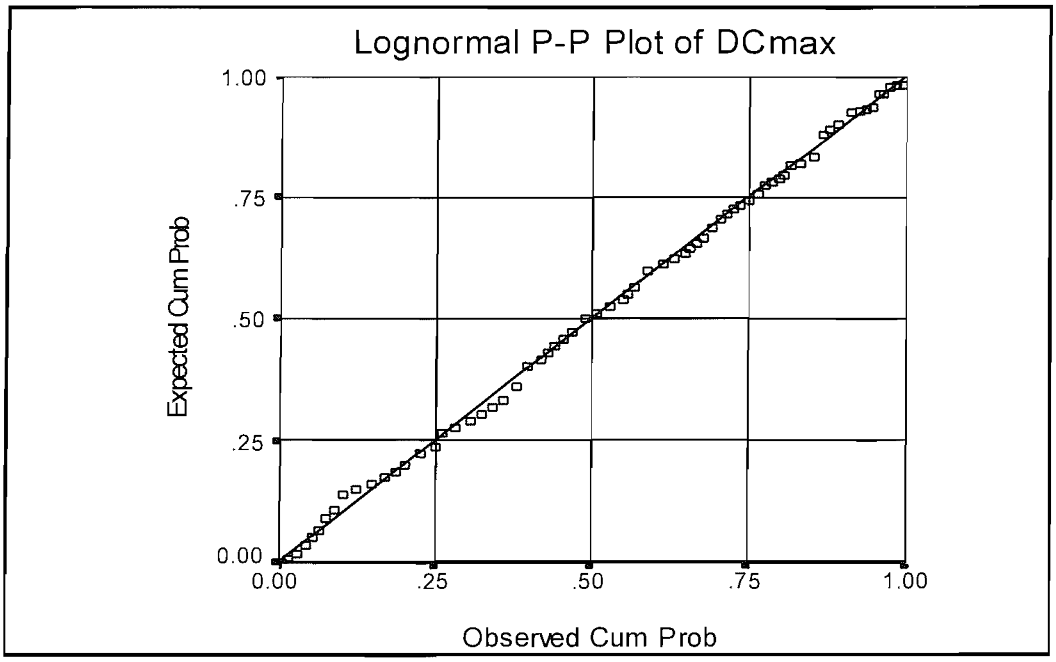


Figure 7.29: The distribution of DC_{max} as an output from the model.

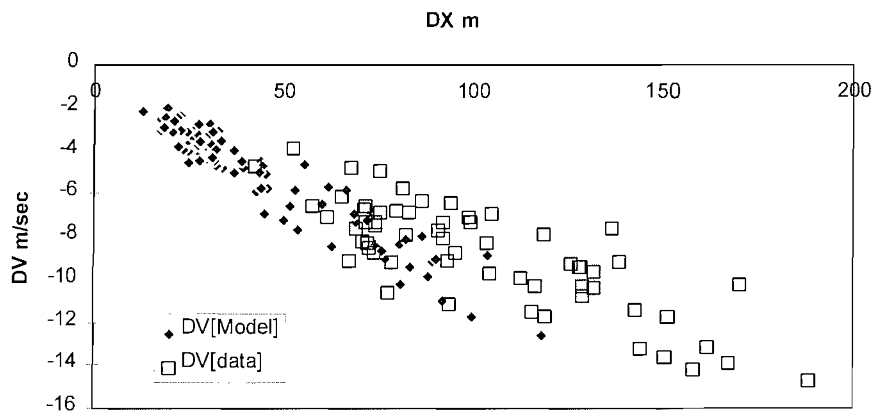


Figure 7.30: the DV-DX relationship for each of the model and the real data.

CHAPTER 8

MODELLING THE CLOSE FOLLOWING PROCESS

8.1 INTODUCTION:

An understanding, and the availability, of a reliable driver behavioural model in the close following situation is the cornerstone of developing a reliable simulation model for traffic on motorway. Although a great deal of exemplary work has been performed in this field, the databases used were collected using either static laboratory simulators, for example (WINSUM and HEINO 1996)[66], or on test tracks, for example (CHANDLER 1958)[14].

The close following process describes process of drivers' behaviour controlling their acceleration or deceleration to maintain a certain range of headway [minimum and maximum] behind a lead vehicle. The importance of this process is that the behaviour of a platoon of vehicles will depend on it, and hence the simulation of traffic shock-wave phenomena. Perhaps the most justifiable formulation for modelling the close following process is that of so called 'Action Point Model', see the literature review [Chapter 2]. Different versions of this model have been independently derived by a range of authors since the 1960s {(MICHAELS 1963)[18], (TODOSIEV 1963)[19] and later (LEUTZBACH W. & WIEDEMANN R. 1986)[21]}.

In this chapter, the 'Action Point Model' used in the MISSION model (WIEDEMANN 1986)[21] is reviewed, and it modification based on the TRG instrumented vehicle data described.

8.2 MISSION MODEL REVIEW:

This model assumes that there are four thresholds which control driver decision in close following process. On crossing a threshold, a driver may perceive that an unacceptable change in either the relative distance or the relative speed has occurred and will execute a minute change in the sign of his or her acceleration or deceleration, typically of the order of

[DC_{NULL} = +/-0.2 m/s^2], (LEUTZBACH & WIEDEMANN 1986)[21] . The four thresholds are described below:

ABX: The desired minimum following distance, It was explained in Chapter 7:

$$ABX = AX + BX \quad ......8.1$$

$$BX = (BX_{add} + BX_{mult} * RND1(i)) * \sqrt{V} \quad ......8.2$$

$$AX = L + AX_{add} + RND1(i) * AX_{mult} \quad ......8.3$$

SDX: This threshold occurs when the driver consciously recognises that he or she is leaving the following process and falling behind the vehicle in front. This threshold varies for each driver from time to time SDX:

$$SDX = AX + EX * BX \quad ......8.4$$

Where:

$$EX = EX_{add} + EX_{mult} * (NRAND - RAND2(i)) \quad ......8.5$$

NRND : A random number has a normal distribution N(0.5,.15) for the same driver.

EX_{add} , EX_{mult} : Calibration parameters, these will produce an increase of SDX over ABX of an additional 0.5 to 1.5 times.

CLDV: The threshold for recognising small speed difference at short decreasing distances. CLDV has similar nature to the SDV (starting approach process Chapter 7) but with larger range of variation. TODOSIVE found CLDV to be about 3 times SDV (TODOSIVE 1963)[19]:

$$CLDV = SDV * EX^2 \quad ......8.6$$

OPDV: The threshold for recognising small speed difference at short increasing distances. TODOSIVE found OPDV to be between one to three times CLDV (TODOSIVE 1963)[19]:

$$OPDV = CLDV * (- OPDV_{add} - OPDV_{mult} * NRND) \quad ......8.7$$

REITER (REITER 1994)[65] found that CLDV and OPDV had a different shape to that suggested by TODOSIVE:

$$CLDV = - 0.5 * RND1 * (0.3 - 0.017 * DX) \quad .....8.8$$

$$OPDV = 0.5 * RND1 * (0.1 + 0.0083 * DX) \quad .....8.9$$

Finally, it was suggested in the model that the driver will add the acceleration or the deceleration of the lead vehicle to his or her deceleration or acceleration in order to avoid any potential collision. It is clear that oscillation between these thresholds will produce the characteristic 'spiral plots' remarked on by many authors, for example (GORDON 1971)[67] . Figure 8.1 shows the thresholds and the spiral plots of the action point model.

Discussion:

1- Although the basis and the structure of this model is now well known, little has been attempted regarding its further calibration and expansion. In this research the model will be re-examined and, where necessary, amended to describe the close following behaviour of U.K drivers.

2- The assumption of adding the deceleration or acceleration of the lead vehicle does not work for a platoon of vehicles, and modification is needed.

3- The influence of the vehicle ahead of the lead vehicle may be important in helping to simulate the close following behaviour in a platoons.

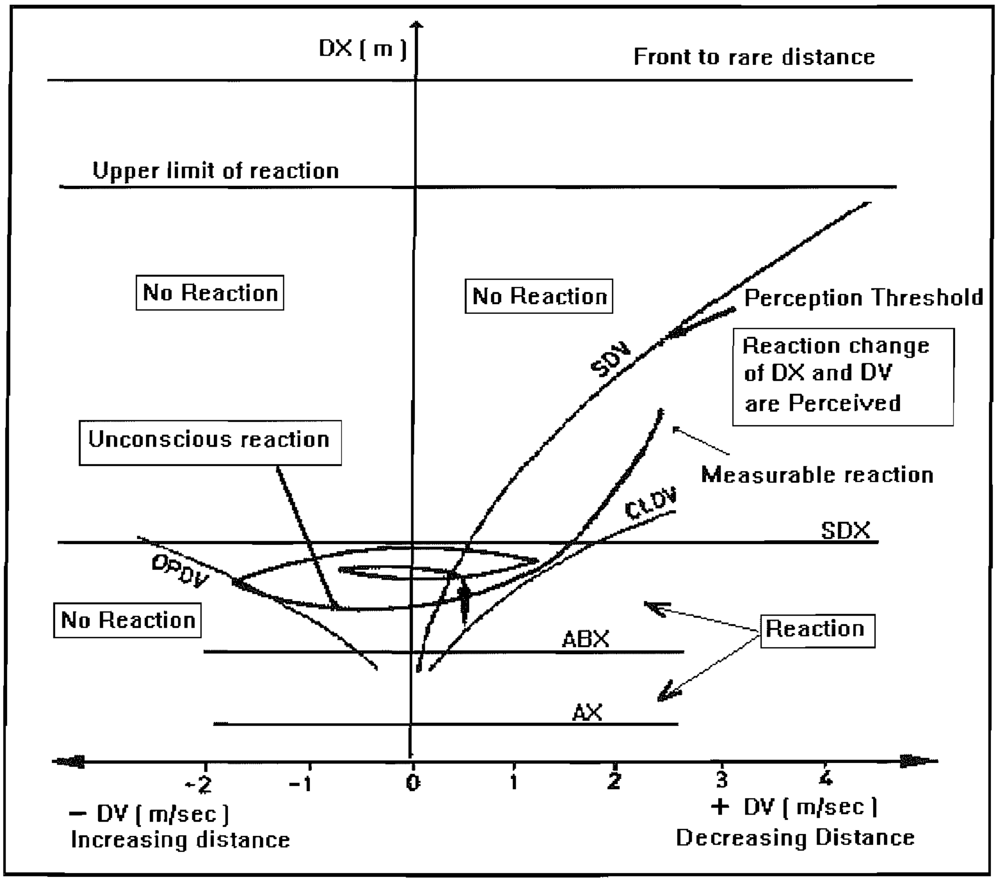


Figure 8.1: the thresholds and the whole concept of the action point model (LEUTZBACH & WIEDEMANN, 1986)[21]

8.3 The Data Collection For Drivers' Thresholds:

The TRG instrumented vehicle (IV) was used to collect the data for the analysis of driver thresholds in the following process. The database was collected in 1997 using the passive mode of collection where the radar in the IV was fitted facing rearward and observations made of adjacent following drivers. This approach allows the collection of data on a wide distribution of driver behaviour in the close following process. The data was collected in two major phases:

- Firstly, experiments were performed during April and May 1997 on the M27 three-lane motorway between junctions 3 and 8 (a total of 13.5 Km), chosen due to the relatively high flow levels found during the morning peak between 7 and 8:30AM. Using this approach 'high speed' traffic (60mph and over) could be monitored in the direction heading away from the city of Southampton, and peak hour traffic, exhibiting congestion and flow breakdown, monitored in the other direction providing data on following at lower speeds. In total, seventy six observed following vehicle sequences were recorded, of approximately two minutes average.
- Experiments were subsequently performed during October 1997 on the M3 three-lane motorway between junctions 2 and 4a (a total of 22.2 Km), during the morning peak between 7:30 and 8:30AM. Thirty-three observed following vehicle sequences were recorded, of little under four minutes average.

Every following sequence consisted of the time, the IV speed, acceleration/deceleration, the relative speed with the follower [Positive ⇔ Opening & Negative ⇔ Closing] and the separation distance. Example of the collected data series are illustrated in Figures 8.2, 8.3, 8.4 and 8.5.

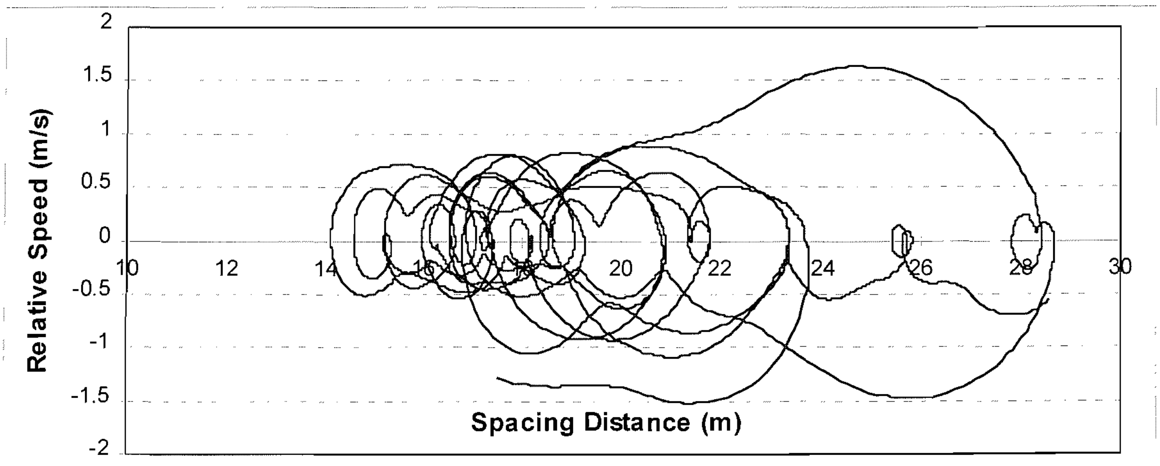


Figure 8.2: The relative speed and spacing distance relationship, (a data sample Long series).

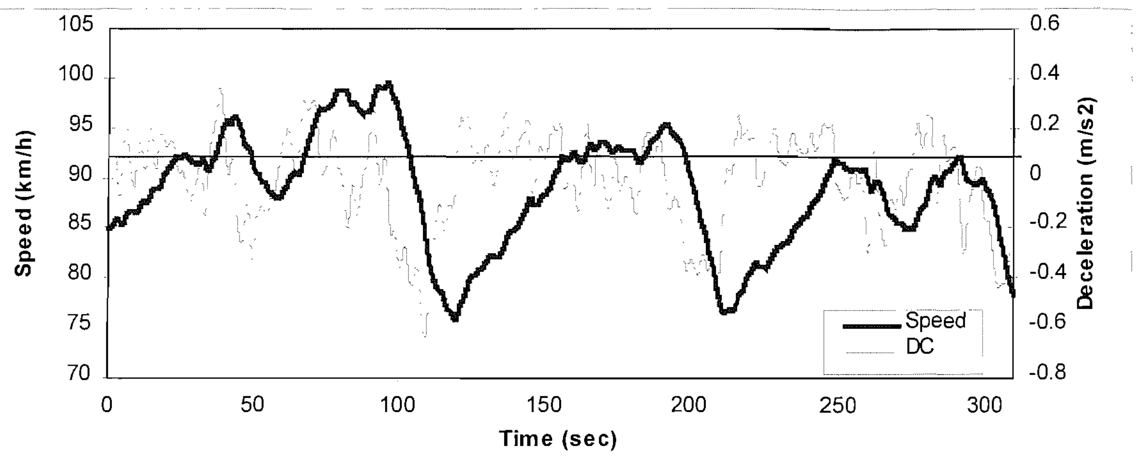


Figure 8.3: The speed and acceleration for the same following trace in Figure 8.2.

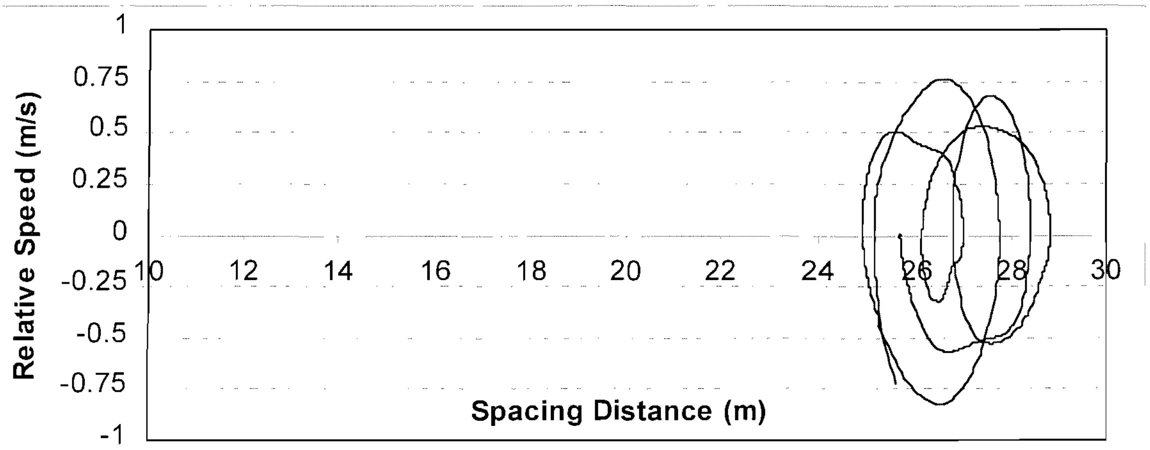


Figure 8.4: The relative speed and spacing distance relationship, a data sample [Short series].

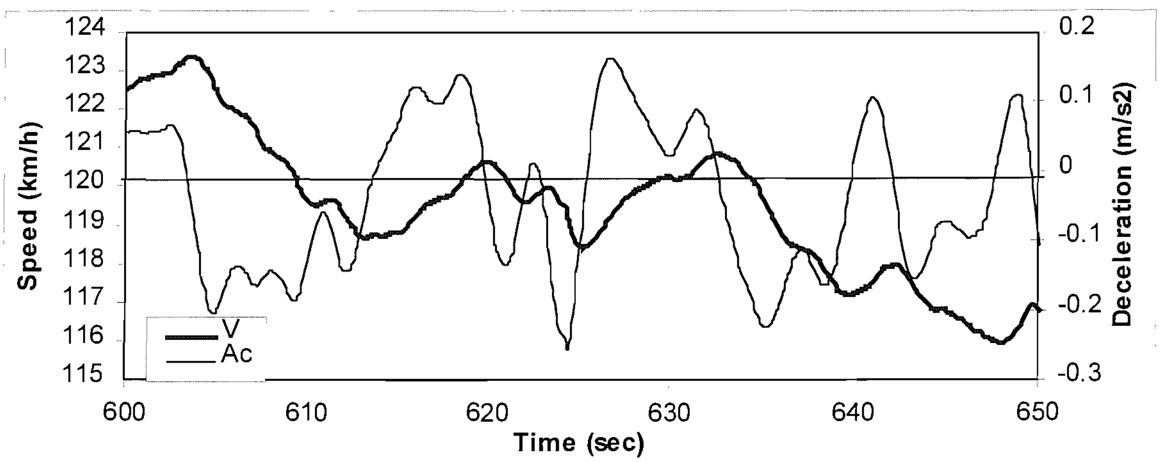


Figure 8.5: The speed and acceleration for the same following trace in Figure 8.4

8.4 The Headway's Thresholds (ABX & SDX):

The first step in calibrating an action point model is to examine the distance keeping behaviour of each driver and find parameters for the two thresholds ABX and SDX. In so doing it is first necessary to identify the specific action or turning points which characterise these thresholds:

- **ABX:** where a trace with decreasing DX and negative DV changes to one with increasing DX and positive DV.
- **SDX:** where a trace with DX increasing and positive DV changes to one with DX decreasing and negative DV.

Although identification of these points is straightforward, care must be taken in their analysis and all traces associated with each observed driver were grouped together and subject to the same five-stage analysis processes described below:

i. Firstly for each observed driver, the traces were divided into 'semi-spirals' or half-cycles, that link each of the ABX and SDX points and the transition time, as well as the ground speed for each of the points noted. Figure 8.6 shows the semi-spirals and the definition of ABX and SDX.

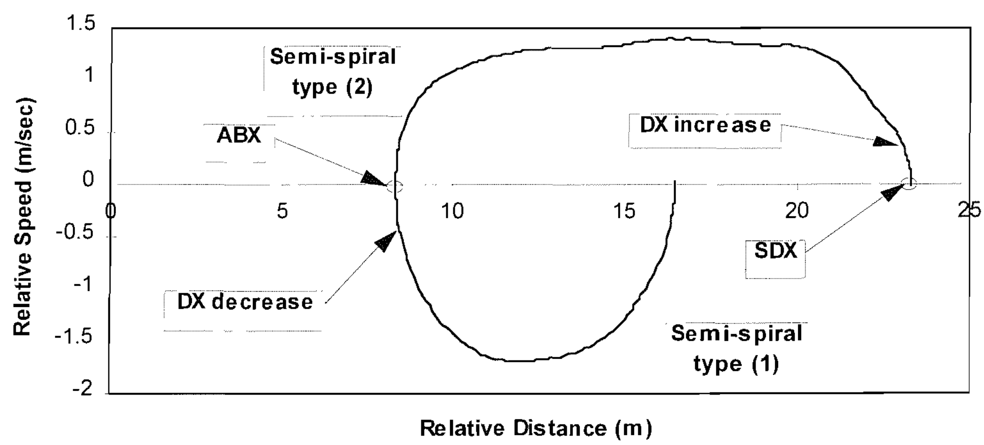


Figure 8.6: The semi-spirals and the definition of ABX and SDX

ii. Next, and in order to ensure only comparatively stable following sequences contributed to the analysis, a cut off was imposed such that any semi-spiral time series was eliminated if the lead vehicle (deceleration / acceleration) was larger than ± 0.6 m/s^2. These limits correspond to the (deceleration / acceleration) variation in calm close following which will be discussed later. Performing steps (i) and (ii) on a every data series is time consuming and a program was developed using Visual Basic code in Excel spreadsheet macro facilities. Figure 8.7 shows the flowchart of the program's stages to isolate the action points ABX and SDX.

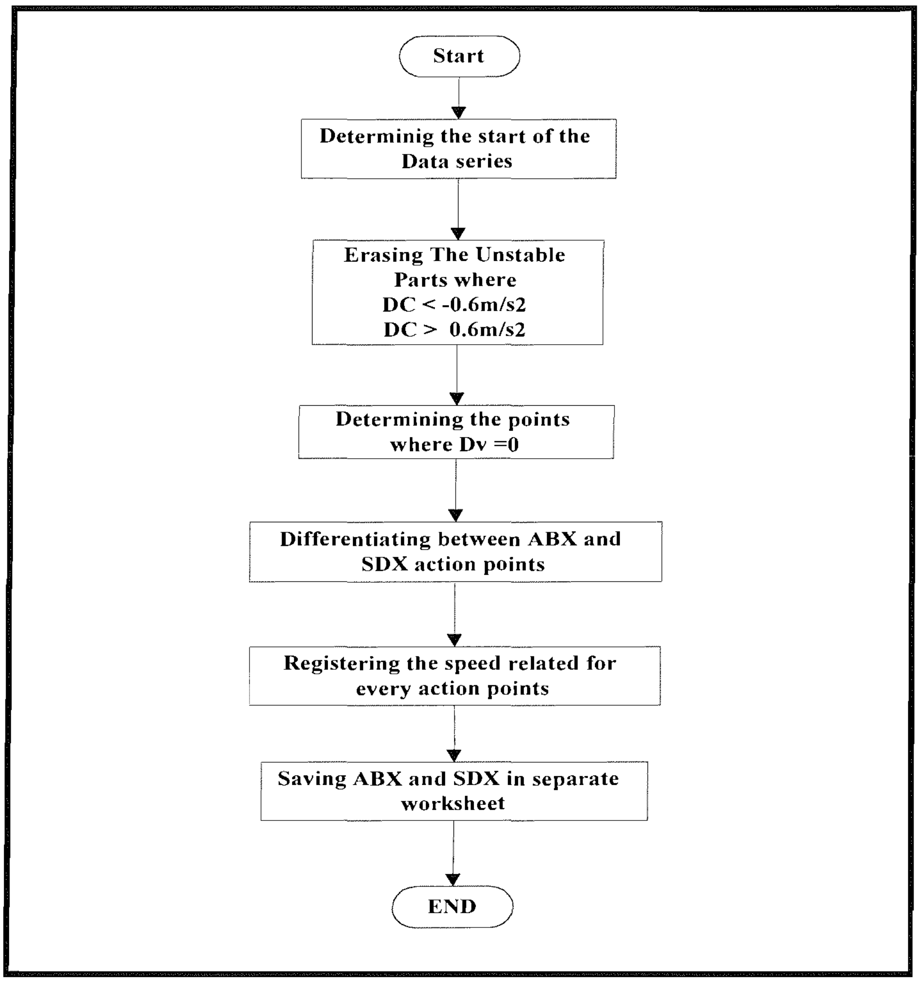


Figure 8.7: The program's stages to isolate the action points ABX and SDX.

iii. Thirdly, any semi-spiral with a proportion change in the relative distance DX of less than (0.06) was discarded. This is because the reported minimum measured value for the just noticeable distance (JND) is [JND $\geq$ 0.06] (EVANS L. & ROTHERY R. 1977)[62]. The reason behind considering such a restriction is that not all of the identified action points may necessarily be associated with a change in the driver's status. For example, there are several places where a trace will perform a 'mini spiral' over a spatial distance of maybe less than one meter. Such spirals are a straightforward result of the natural fluctuations present in traffic. Although we are examining the adjustment of a driver to the behaviour of the vehicle in front, it must be born in mind that the lead driver is varying his or her speed as part of their own distance keeping process. Several researchers, have suggested that JND might be in the limits of (0.12) (BRAUNSTEIN M. L. & LAUGHERTY K. R. 1964)[68]. Therefore, another criteria was considered by discarding any semi-spiral with [(SDX / ABX) < 1.12]. This issue is discussed in Appendix (3) by presenting and comparing the results from each assumption.

iv. Each set was divided according to the speed interval within which each 'semi-spirals' took place, in 10 km/h intervals, with a minimum of 10 seconds worth of time series data being required from any one driver for inclusion.

v. The minimum observed value of DX in each interval (or in the case of the analysis of SDX, the maximum) for each driver was identified, along with its associated ground speed, and defined as that particular drivers value for ABX (SDX) for that interval.

The last part of this analysis was particularly important, as it is tempting for statistical reasons to include all points for each driver for each interval and not just the maximum and minimum. The reason behind this however, is that drivers also try to adjust their speed to match the leaders speed which generates internal spiral during the following process. (It is to be noted that these internal spirals are an essential part of the microscopic process and indeed their amplification and propagation may play an important part in the start of flow breakdown (LOW & ADDISON 1995)[69]). Therefore, the production of one point for each speed interval will eliminate all spurious points related to the speed control, with the remaining point being likely to be caused by the driver reaction to his or her minimum and maximum acceptable distance ABX and SDX. To a certain extent, this treatment also minimises the effect that any one driver or time series may have on the distribution, with at most one point being produced for each speed range, and hence producing effectively a distribution equally weighted over the observed population. (In practice, the maximum number of speed ranges contributed to by any one driver was 8, with on average 2 ranges being contributed to by most drivers, and 4 only being exceeded in 6 of the 109 cases). A plot of points produced at this stage is presented in Figure 8.8.

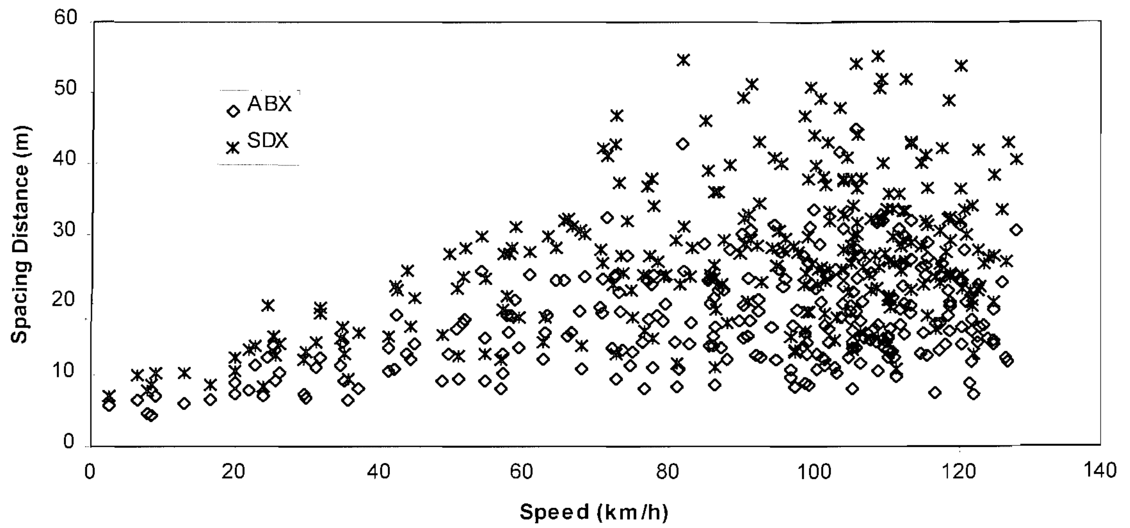


Figure 8.8: The ABX and SDX with respect to the speed.

8.4.1 ABX and SDX Limits:

The output of the preceding analysis was a distribution of ABX and SDX points across the observed driving population. The absolute maximum and minimum for each ABX and SDX value in every speed interval was identified in order to draw the limits for these two thresholds. Different curves were fitted to model the ABX and SDX limits. The best fit was found to be a power relationship with the speed level and these are presented in equations 8.10, 8.11, 8.12 and 8.13 and Figure 8.9.

$$ABX_{min} = 3.2485 * V^{0.2118} \quad .....8.10$$

$$ABX_{max} = 1.6184 * V^{0.6655} \quad .....8.11$$

$$SDX_{min} = 3.4033 * V^{0.2976} \quad .....8.12$$

$$SDX_{max} = 2.0687 * V^{0.6944} \quad .....8.13$$

The data shows that there is a clear decline in all ABX and SDX limits with lower speed. Nevertheless, the ABX_{max} and SDX_{max} have a noticeable drop in the value over the highest two speed levels [(110-120 Km/h) and (120-130 Km/h)]. A possible explanation for this is that drivers take more risk at high speed by following with shorter headways.

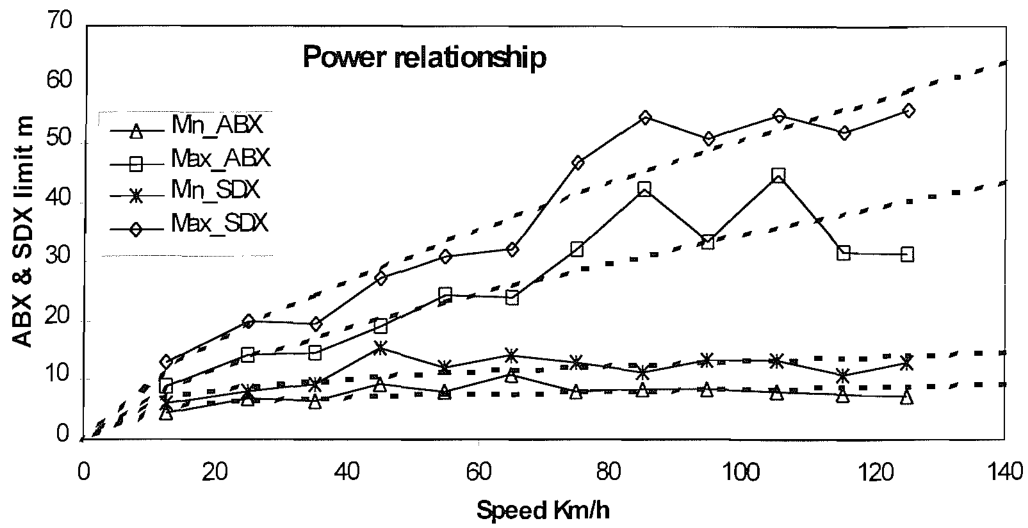


Figure 8.9: The ABX and SDX limits and the fitted curves with respect to the speed.

Also, the level of flow associated with this speed ranges is less than other ranges [Speed < 110 Km/h], so the congestion is less and drivers to be more risks. Although, the power relationship between ABX and SDX limits and the speed does not provide this type of decrease in the ABX_{max} and SDX_{max} at the high speed levels, it provides a better model for the lower speed levels. This is acceptable for two main reasons:

1- The model objective is to simulate conditions of high congestion on motorways, and in this situation it is very rare to achieve high speeds level. At lower flow levels, this minor difference between the model and the data will not affect the macroscopic measurement that is taken from the model.

2- The data did not provide any information on speeds higher than 130 km/h. Thus, it is better not to show a decline in the model of ABX_{max} and SDX_{max} limit with speed, as this would affect the continuity of the model and might lead to unacceptable values at very high speed. (ie excess of the speed limit).

8.4.2 The ABX And SDX Variation:

Action points has been presented as a function of a probability distribution based on ground speed rather than as a functional description of the ABX and SDX points based on an average and standard deviation for each speed interval. (ie. as previous research by LEUTZBACH and WIEDEMANN 1986)[21]. Figure 8.10 presents the cumulative probability of ABX for every speed interval.

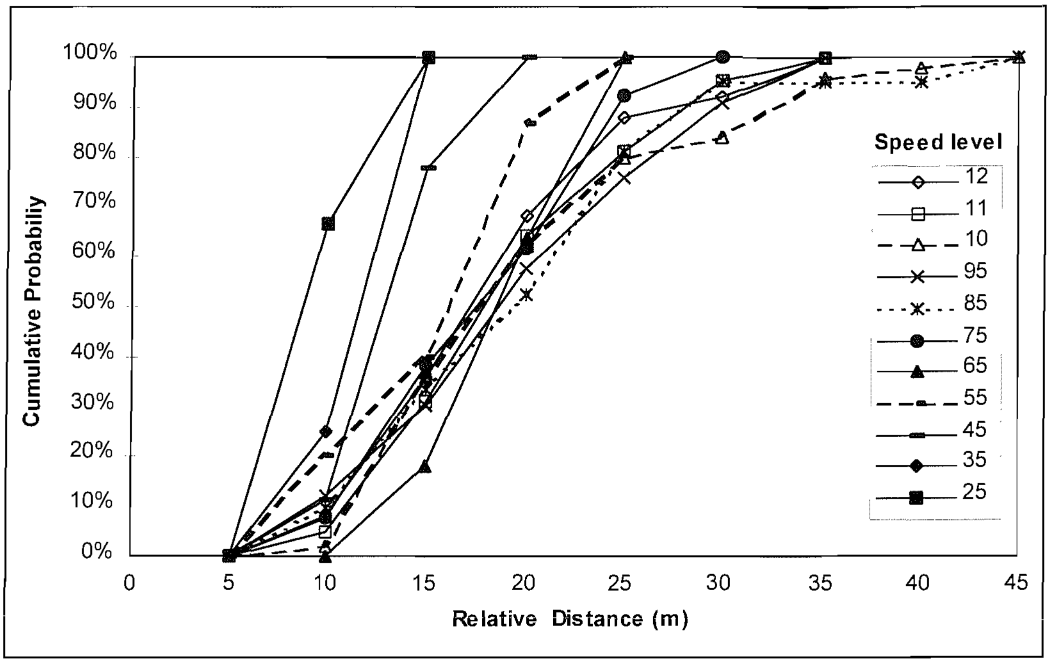


Figure 8.10: the cumulative probability of ABX for every speed interval.

It is difficult to interpret and draw conclusions from this graph. Therefore, the cumulative probability was reconfigured between following distance levels (Figure 8.11). Later, curves were fitted for every line and the result presented in Figure 8.12.

The equations for the fitted curves is listed below:

$$\begin{aligned}
&\text{Percentage (30m)} = +14.521\text{E-}03\ V^2 - 3.07080\ V + 250.92 \\
&\text{Percentage (25m)} = +20.822\text{E-}03\ V^2 - 4.21380\ V + 290.08 \\
&\text{Percentage (20m)} = -368.84\text{E-}06\ V^3 + 0.116460\ V^2 - 11.7954\ V + 442.64 \\
&\text{Percentage (15m)} = -555.36\text{E-}06\ V^3 + 0.157100\ V^2 - 14.3114\ V + 450.14 \\
&\text{Percentage (10m)} = +7.035\text{E-}06\ V^4 - 2.3742\text{E-}3\ V^3 + 0.28992\ V^2 - 15.198V + 297.746
\end{aligned} \qquad ....8.14$$

Figure 8.12, demonstrates that as speed increases, the probability of a driver's action point being at a short distance decreases, with an increase of 'action point density' at higher distances. This process continues up to a speed of about 70kph where the outward progression of the density profile slows, and around 95kph stops entirely with subsequent increases in speed actually producing a decrease in action point density at higher values of ABX. In essence, a distance keeping behaviour with two phases wo0uld seem to be in evidence.

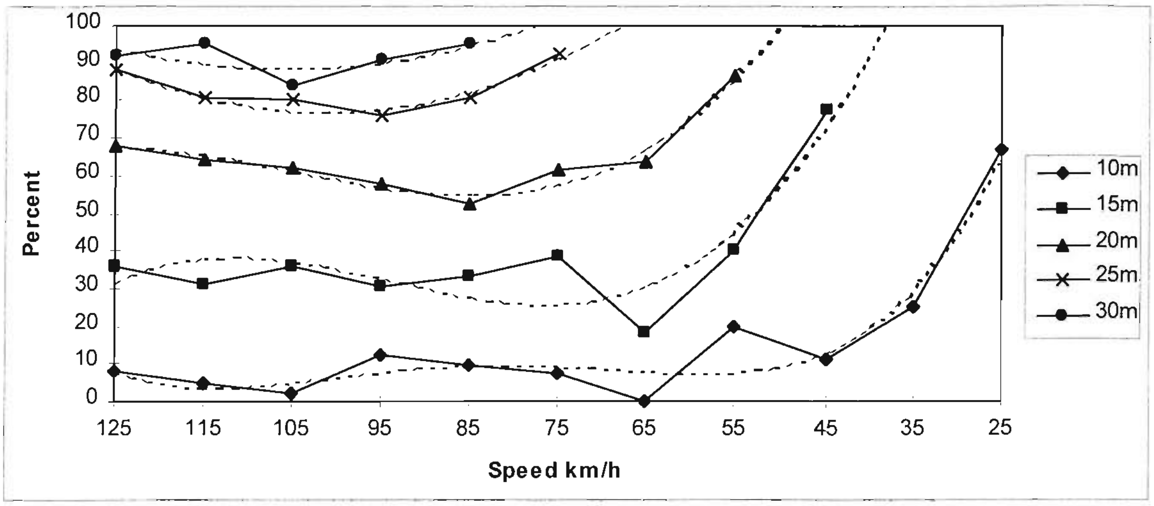


Figure 8.11: The Probability of ABX with respect to speed level (fitted curve).

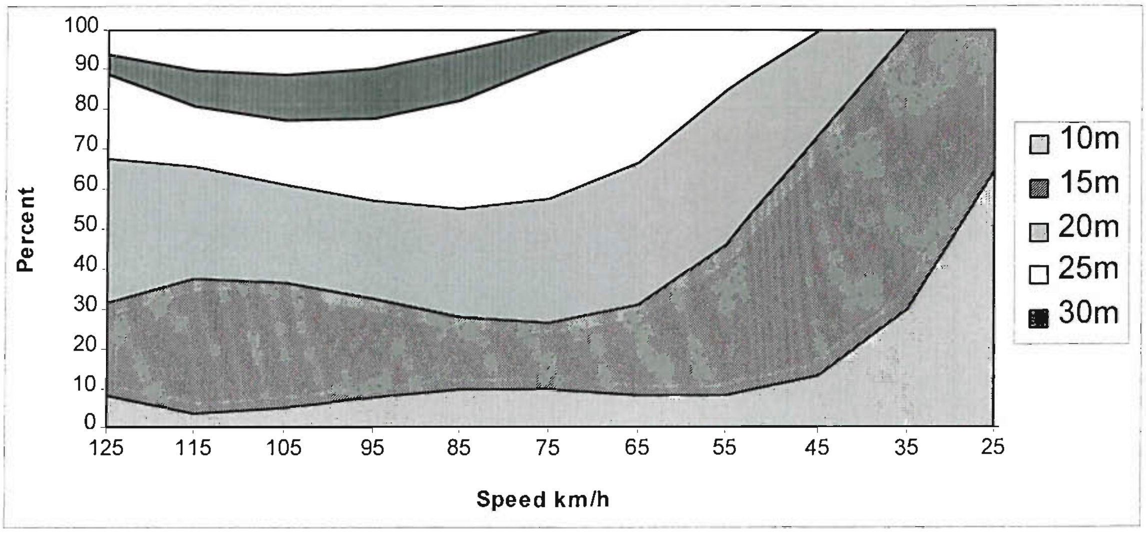


Figure 8.12: The Probability of ABX with respect to speed level (smoothed version).

- Firstly, there is an increasing following distance with speed, giving way to a constant value of around 65 to 75 kph. Such a trend is perhaps unsurprising, as it would seem intuitively obvious that the faster a vehicle travels, the more space a driver is going to allow to account for stopping distances.

- The second transition, however, is more surprising with the speed invariant headway reducing above 105 kph, perhaps reflecting the onset of a more aggressive type of behaviour. (This has been discussed previously).

SDX has a similar probability distribution. However, as the definition of this threshold is the maximum acceptable following distance for a driver for a specific speed level and, in order to avoid any conflict between ABX and SDX, the ratio (SDX/ABX) has been examined.

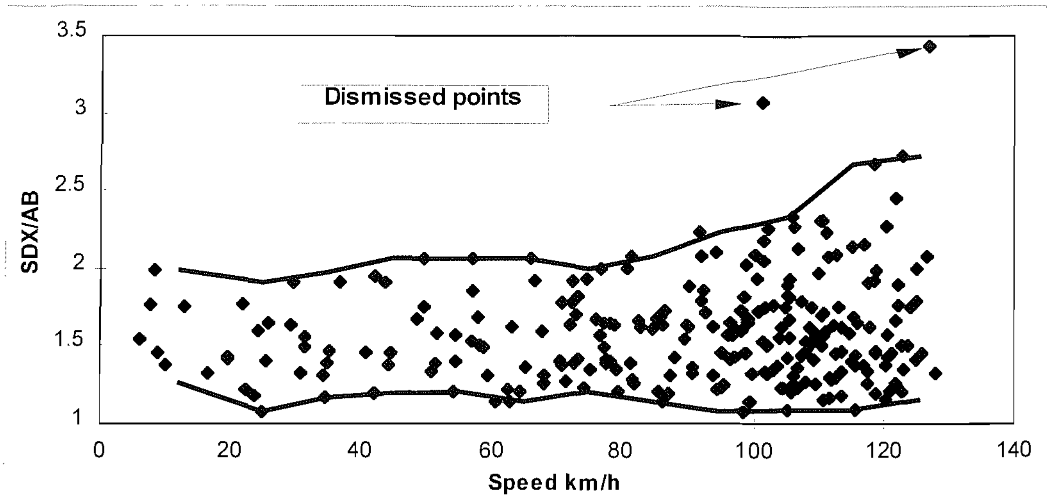


Figure 8.13: The (SDX/ABX) ratio for all drivers and its bands.

The SDX/ABX ratio for all drivers is shown in Figure 8.13. General boundaries have been drawn on the Figure. Two points were ignored as out lies, and may be the result of the driver intending to leave the following process. (Other interpretations are possible). The curve fitted for the maximum value of the ratio (SDX/ABX) against speed was found to be:

$$(SDX/ABX)_{max} = 104.69E\text{-}6*V^2 - 8.092E\text{-}3*V + 2.0991 \qquad ....8.15$$

For the lower boundaries, the values were fairly independent of speed, with a mean of: $(SDX/ABX)_{min} = 1.15$ (Figure 8.14).

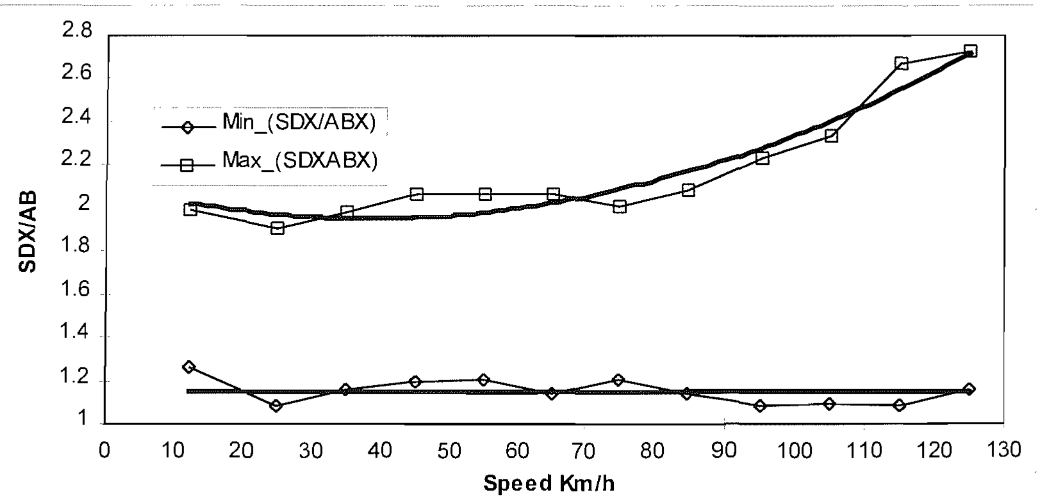


Figure 8.14: The (SDX/ABX) boundaries and fitted curves.

The same approach was followed when studying the variation of the (SDX/ABX) ratio. The cumulative probability percentage was plotted against speed, and a linear regression was determined to model this probability. Figures 8.15 and 8.16 show the probability of the SDX/ABX ratio with speed. The equations are listed below :

Per[(SDX/ABX)=1.25] = -0.06 V + 19.68	R^2=0.05
Per[(SDX/ABX)=1.50] = -0.17 V + 63.2	R^2=0.33
Per[(SDX/ABX)=1.75] = -0.11 V + 84.55	R^2=0.21
Per[(SDX/ABX)=2.00] = -0.16 V + 102.62	R^2=0.60
Per[(SDX/ABX)=2.25] = -0.37 V + 133.4	R^2=0.84
Per[(SDX/ABX)=2.50] = -0.21 V + 121.78	R^2=0.99

.....8.16

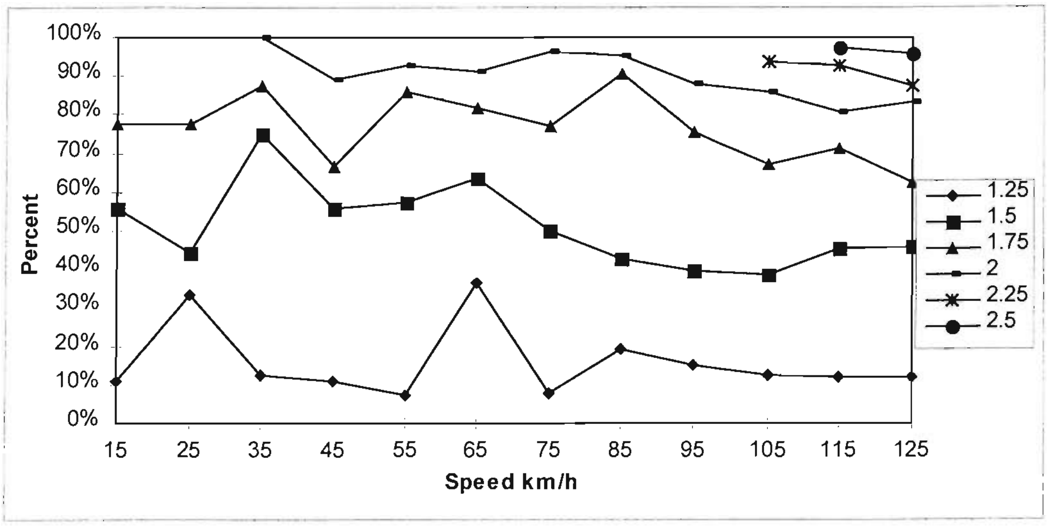


Figure 8.15: The probability of (SDX/ABX) with respect to the speed level.

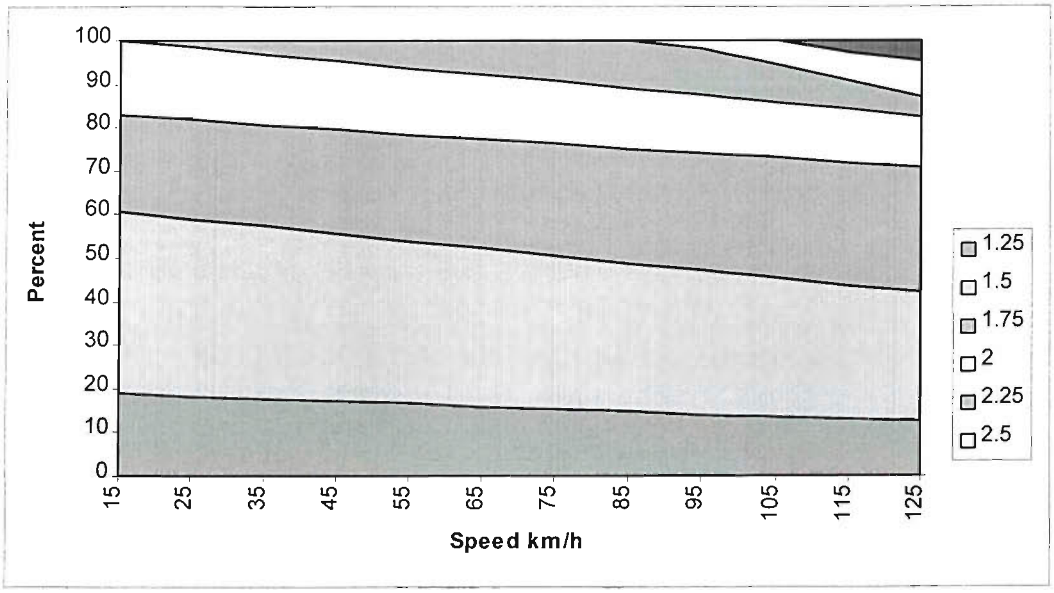


Figure 8.16: The probability of (SDX/ABX) with respect to the speed level smoothed.

It can be seen that a linearly increasing trend is present, with around 80% of the points lying between 1.5 and 2.00 and lower speeds, and 1.5 and 2.5 at higher values. However, this result is not different to that was used in the MISSION model (value of (SDX/ABX) around 1.5 to 2.5) (LEUTZBACH W. & WIEDEMANN R. 1986)[21].

Discussion :

In order to compare the headway results with those from elsewhere, two differences should be pointed out between the data from this study and that of others need to be noted.

- Firstly, all static surveys of headway consist of spot measurements in space and time across the driving population. The TRG surveys contains far more variation, and is not equally weighted across all the observed drivers. (This point has been discussed earlier).
- Secondly, when examining spot measurements there is no way of telling at exactly which point on a spiral plot of following behaviour that the observation has been made. The observation could be a point close to SDX or ABX. The best can be hoped for is that over the course of the study an average between the two can be reached.

In order to retain comparability, the average following distance under consideration here has been taken as an average of the observed ABX and SDX points in our data. An illustration of how closely the data collected in our experiments agrees with other sources, including the US 'California Code' (1958)[14], Highway Code HMSO (1993)[70], XING (1995)[71], HOGEMA (1996)[72], HUDDART AND LAFONT (1990)[73], and PARKER (1996)[74] is shown in Figure 8.17.

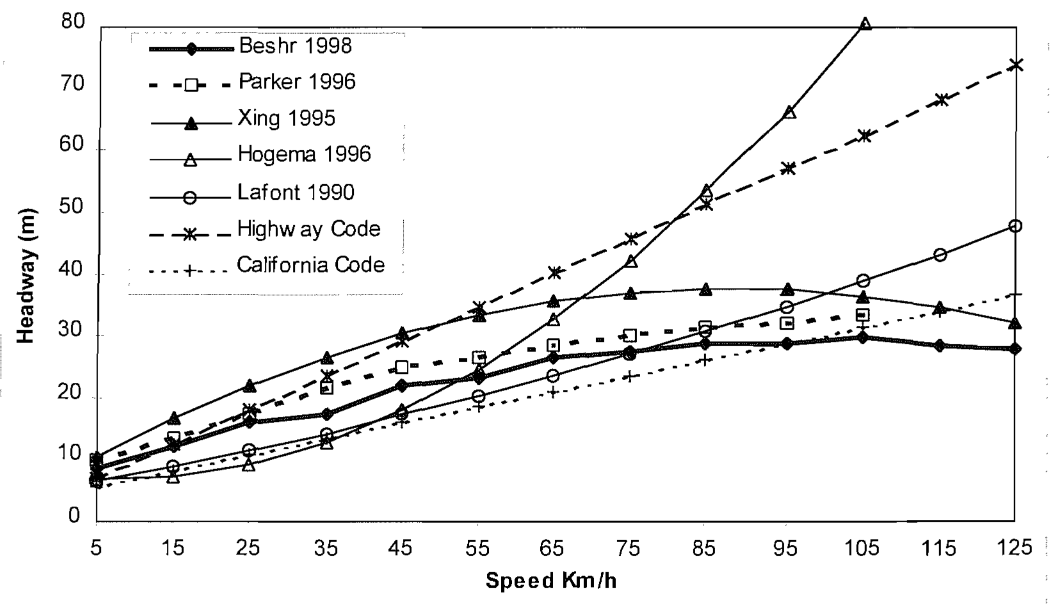


Figure 8.17: Headway comparisons.

Some studies produced results which broadly agreed with our results (PARKER and XING). COLBOURN in investigating driver judgement of safe distances using an instrumented vehicle on a test track, reported an average following time headway of around 2 seconds at 80kph (only reducing to 1.7 sec. at 48kph) (COLBOURN 1978)[75],. Additionally, HOGEMA has reported a mean following headway of 2.4 sec at 100kph in simulator experiments (HOGEMA 1996)[72]. There are two possible explanations for these discrepancies. Firstly, the methodology of COLBOURN and HOGEMA used test vehicles with test drivers on a test track. (ie low risk circumstances). Secondly, it may be possible that the absence of traffic in neighbouring lanes influenced driver behaviour. (In real life and high flow conditions, it is highly likely that a neighbouring vehicle may move into it so large a gap, potentially delaying the driver further). These, and potentially other factors, highlight to the need to collect normative driver behaviour in realistic situations, from within the traffic stream.

8.4.3 Modelling ABX and SDX Thresholds:

As the probability of ABX and (SDX/ABX) changes with speed, a probability model will work better than deterministic equations. In order to mimic the variation between drivers in these thresholds, every driver will have two random numbers [Rand1 and Rand2] with a values between [0 to 1]. The first, [Rand1], is to generate the variation in ABX, and the second, [Rand2], is to generate the variation in (SDX/ABX) ratio. However, Rand1 and Rand2 will have different values whenever the driver follows another vehicle or becomes free following(DX > SDX and DV >OPDV).

The model determines the ABX threshold by performing the following steps:

1- Calculating the ABX minimum and maximum limits from equations 8.10 and 8.11.

2- Determining the percentage values for the following distance boundaries [10, 15, 20, 25 and 30m] from equation 8.14.

3- Finding which boundaries accommodate the Rand1 number, and assessing the ABX value related to the Rand1.

4- Making sure that the ABX value is inside the limits ABX_{min} and ABX_{max} .

Figures 8.18 and 8.19 show the flowcharts for the above subroutine.

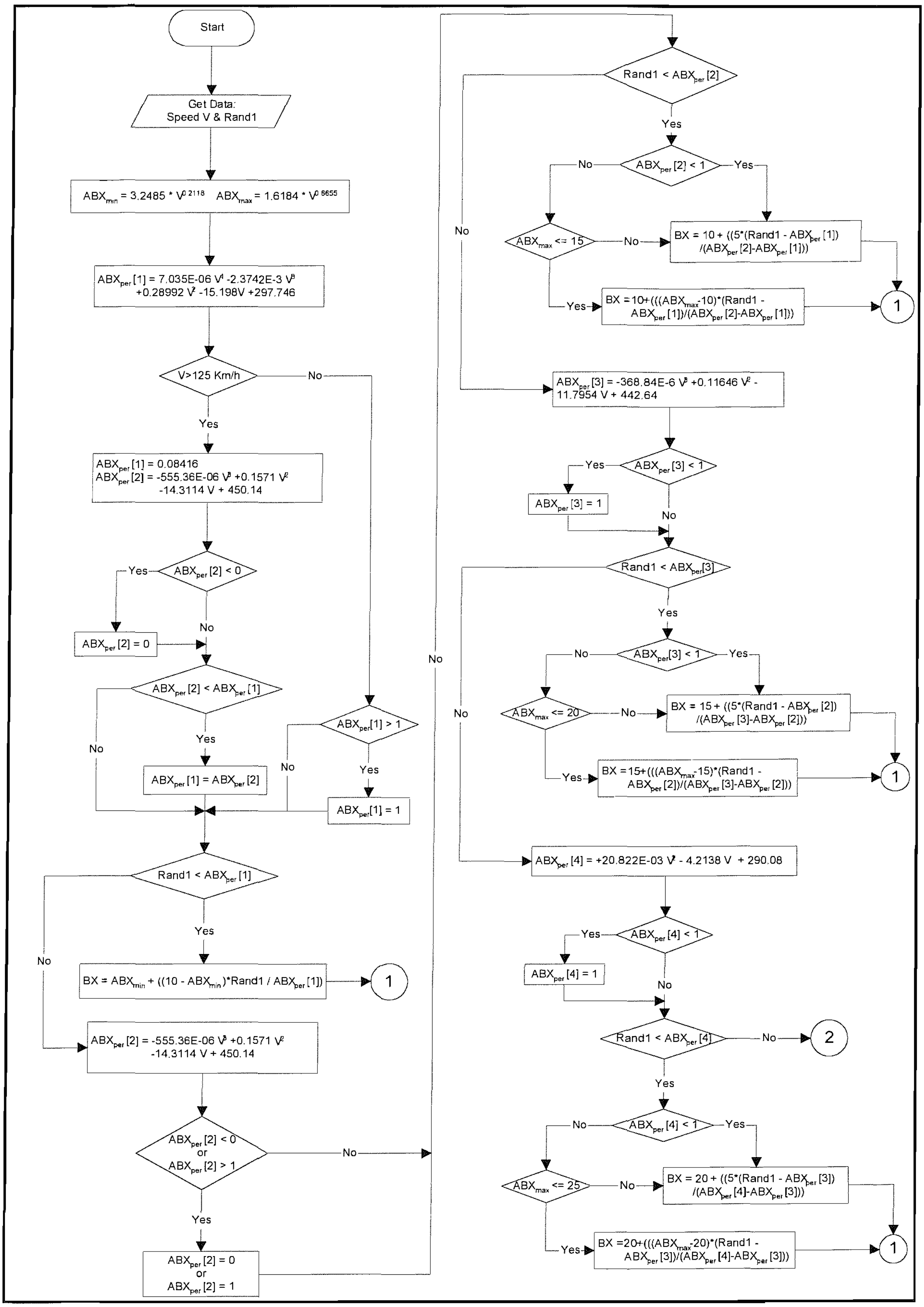


Figure 8.18: The ABX threshold subroutine (1)

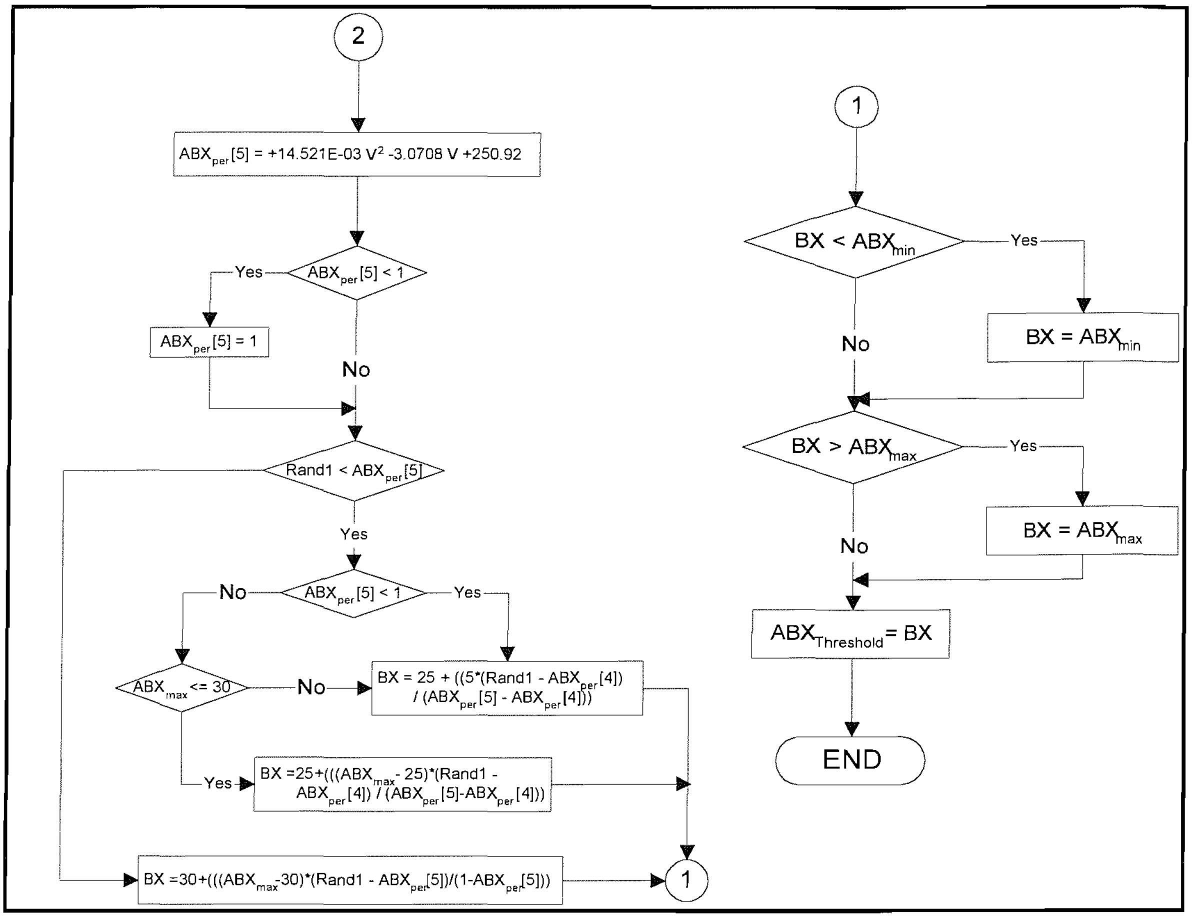


Figure 8.19: The ABX threshold subroutine (2).

After determining the ABX threshold, the model will determine SDX in a similar way as for the ABX subroutine:

1- Calculating the SDX minimum and maximum limits from equations 8.12 and 8.13.

2- Calculating the (SDX/ABX) maximum limits from equation 8.15, the minimum is 1.15.

3- Determining the percentage values for the (SDX/ABX) boundaries [1.25, 1.5, 1.75, 2.0, 2.25 and 2.5] from equations 8.16.

4- Finding which boundaries accommodate the Rand2 number, and assessing the (SDX/ABX) value related to the Rand2.

5- Calculating SDX from (SDX/ABX) and ABX , and making sure that the SDX value is inside the limits SDX_{min} and SDX_{max} .

Figure 8.20 shows the flowcharts for the above subroutine.

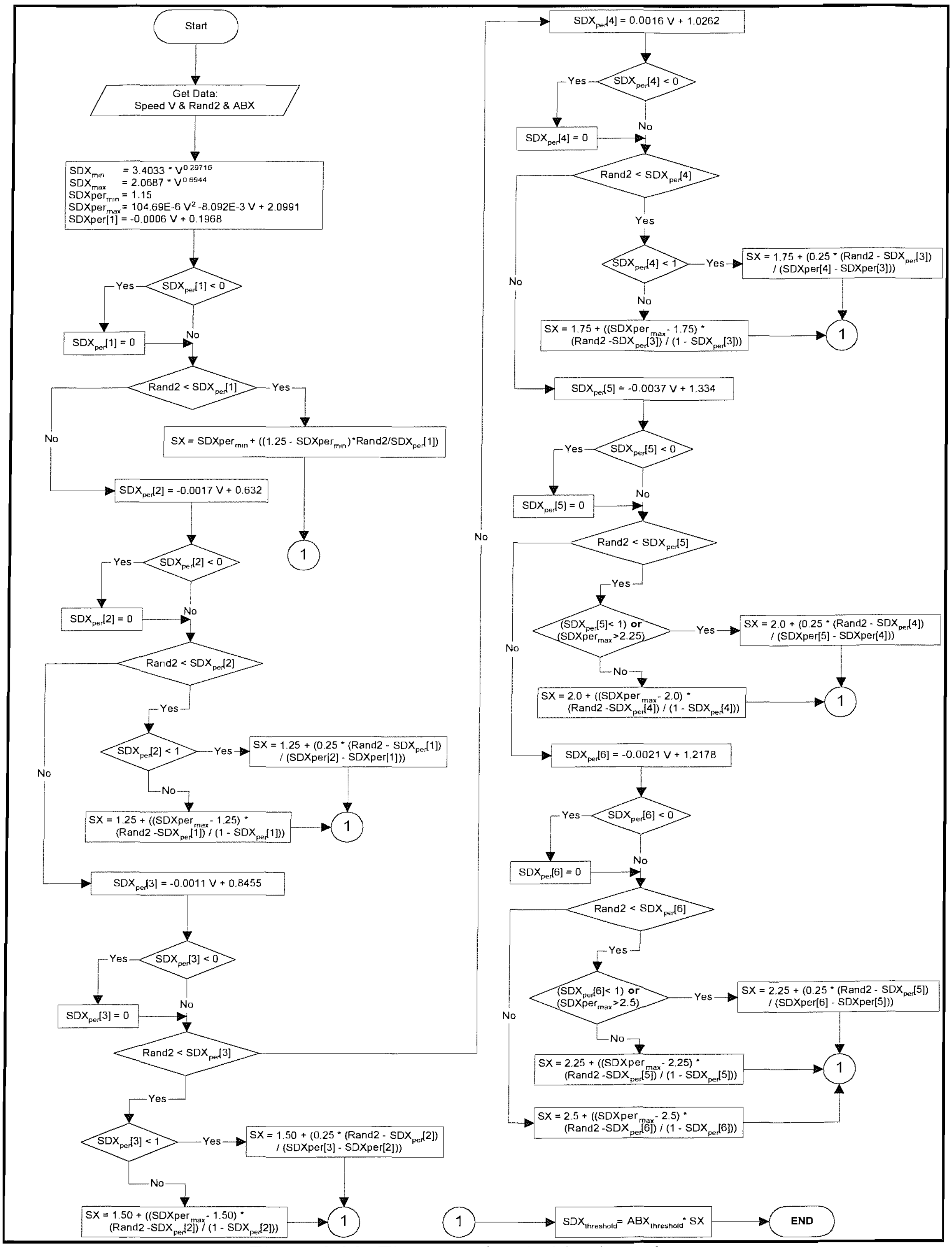


Figure 8.20: The SDX threshold subroutine

8.5 The Relative Speed Thresholds [CLDV & OPDV]:

The same basic process for the selection of action points was implemented, as in the previous section, to analysis of CLDV and OPDV. However, two changes were made to the process:

- Firstly, CLDV was defined as the minimum value of the relative speed in the semi-spirals where DV is negative, while the OPDV was defined as the maximum value of the relative speed in the semi-spirals where DV is positive. Therefore, at stage [v)] (Section 8.4), the action point derived is no longer the minimal/maximal value of ABX etc., but is now taken to be the point with the maximum value of DV for OPDV and minimum value for CLDV.

- Secondly, introducing a new stage between [iv)] and [v)] (Section 8.4), where a cut-off threshold for the selection of action points of [+0.01 > (DV/DX) > -0.01 m/s] was imposed. High frequency transients were removed, the principal reason behind being to exclude any points where an action was being taken that was not controlled by the driver's perception of the relative speed. The drivers' perception threshold was discussed and amplified in the previous Chapter [Section 7.3.1].

A program was developed to automate the process by using *Excel Spread-Sheet Macro*, and the output was saved in separate file. Figures 8.21 and 8.22 show the CLDV and OPDV action points according to the relative distance.

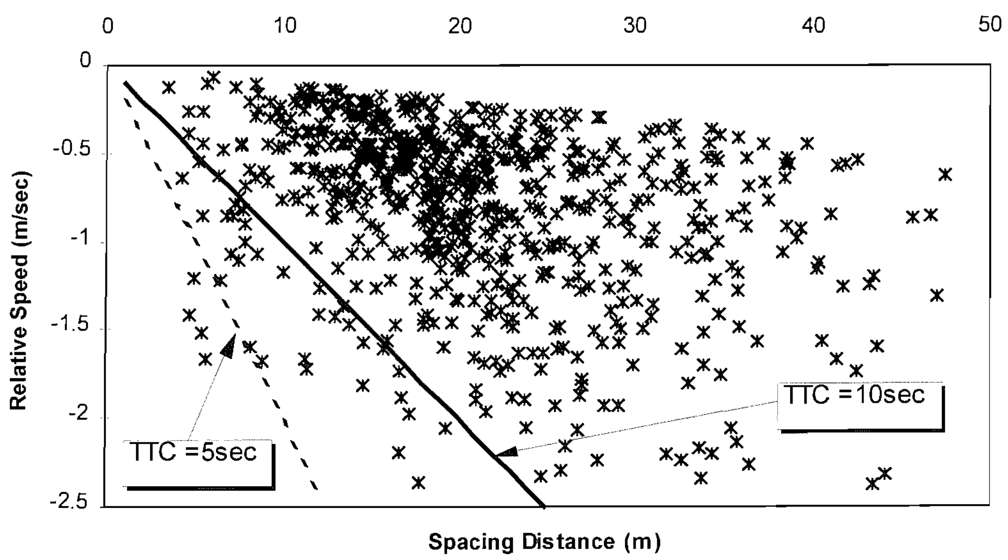


Figure 8.21: The CLDV action points with respect to the Spacing distance.

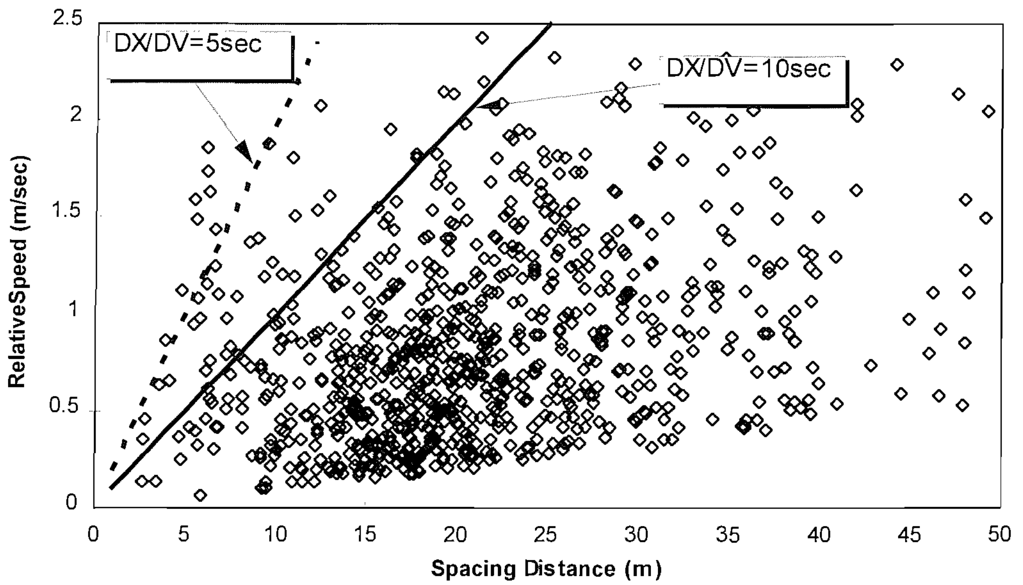


Figure 8.22: The OPDV action points with respect to the spacing distance.

It is noticeable from the graphs that neither CLDV or OPDV has the characteristic curves, (ie quadratic in DX), that would be indicative of the use of a threshold, or even a process related to the optic flow angle (dθ/dt). The most recent work to date on OPDV and CLDV thresholds in this context confirms this result (REITER 1994)[65], with the thresholds being approximately linear in DX with only a slight dependence. However, the way in which the data was collected has not been explained.

An alternative interpretation, suggested by LEE (LEE D. 1976)[54] and measured by WINSUM and HEINO (WINSUM & HEINO 1996)[66] was that decision points are in the closing process are in some way related to the perception of the time to collision (TTC=DX/DV). In their experiments, drivers in a simulated vehicle were exposed to car following and their reaction to a deceleration of the lead vehicle measured. Here it was noted that drivers started to decelerate themselves when TTC dropped under around 8 to 15 seconds (TTC_0), and increased their level of braking in order to prevent TTC from falling under a subsequent minimum (TTC_{min}) for the process of 2.5 to 5 sec. Although these experiments do not really replicate the conditions which we have observed (all excessive decelerations were removed from our analysis), it should be possible to transfer TTC_0 to CLDV. The examination of our data in the 50-60km/h range used in the van WINSUM study reveals that the vast majority of the action points has a TTC value more than 20 sec. Figure 8.23 presents the CLDV points over the speed range [50-60 km/h] and the TTC bands of 10, 15, 20 and 25 sec. Therefore, a general consideration of the data would seem to suggest that using TTC as a characteristic would appear to be less valid that the DV approach that we have adopted.

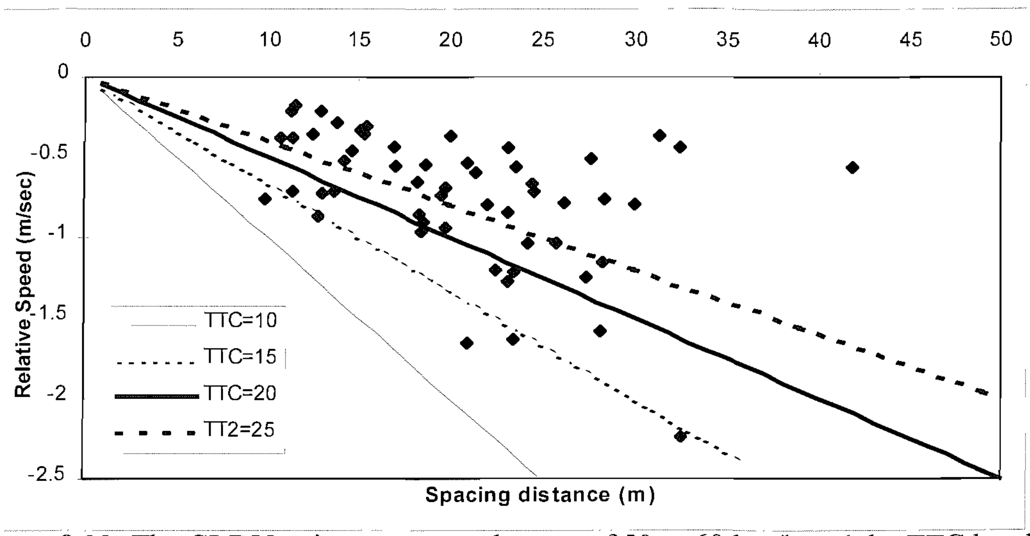


Figure 8.23: The CLDV points over speed range of 50 to 60 km/h and the TTC bands.

The next step was to examine whether or not the CLDV and OPDV action points were affected by speed. The results were obvious from Figures 8.24 and 8.25 where the two thresholds (CLDV and OPDV) were shown to be almost totally speed invariant, with around two thirds of the (almost lognormally distributed) points lying under a value of 1 m/s.

The previous results direct us to the following conclusions:

1- The probability approach seems to be the best way of modelling the CLDV and OPDV action points.

2- The data should be analysed regardless of the speed level.

3- Figures 8.21 and 8.22 show that for a relative distance less than approximately 20 meters the number of action points which are more than (1.5m/sec), in absolute value, becomes less and less. Therefore it was suggested to divide the data into two sets according to the relative distance DX and they are: [DX ≤ 20m] & [DX > 20m].

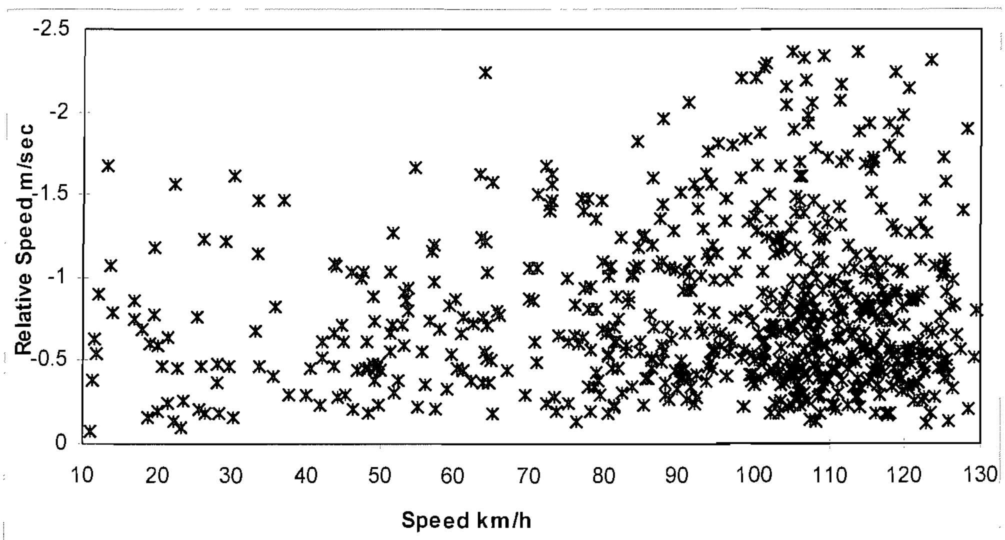


Figure 8.24: The CLDV action points against speed.

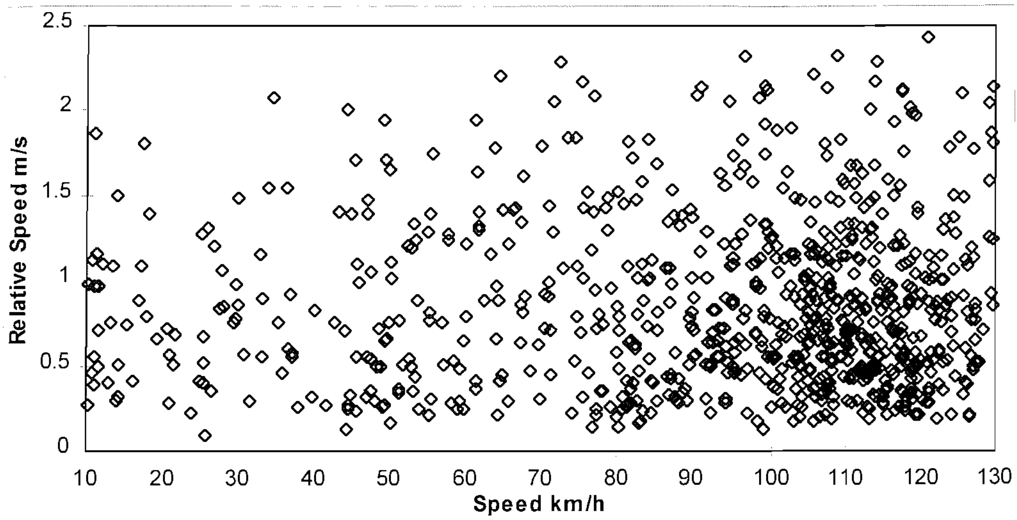


Figure 8.25: The OPDV action points against speed.

8.5.1 Modelling CLDV and OPDV Thresholds:

The first step was to find a distribution to fits the data. There are two possible ways to do this task ***either*** by the cumulative percentage and curve fitting ***or*** by distribution fitting and K-S test.

- ***The Cumulative Percentage And Curve Fitting Method*** **:**

  This method is practical and suitable for any type of distribution, the only draw back being that it adopts the data characteristics precisely, and any defect or shortcoming in the data will appear in the distribution. In this case, the cumulative percentage was calculated for every data set [CLDV (DX≤20m), CPDV(DX>20m), OPDV(DX≤20m), and OPDV(DX>20m)]. Then curves were fitted to the data in order to have a continuity in cumulative percentage over the field of the relative Speed. The equations were found to be :

$$\begin{aligned} P[CLDV(DX\leq 20m)] &= -16.100*DV^3 -96.010*DV^2 -191.20*DV -28.733 \\ P[CLDV(DX>20m)] &= -6.6715*DV^3 -54.110*DV^2 -151.73*DV -45.325 \\ P[OLDV(DX\leq 20m)] &= 10.863*DV^3 -73.993*DV^2 +167.79*DV -27.066 \\ P[OLDV(DX>20m)] &= 3.7440*DV^3 -39.950*DV^2 +131.83*DV -38.607 \end{aligned} \quad \text{....8.17}$$

Figure 8.26 presents the cumulative percentages for the CLDV and OPDV action points data with the fitted curves. It is clear that the curves' equations provide a good fit, and the error margin is very small, (The approximate r^2 is 0.946%).

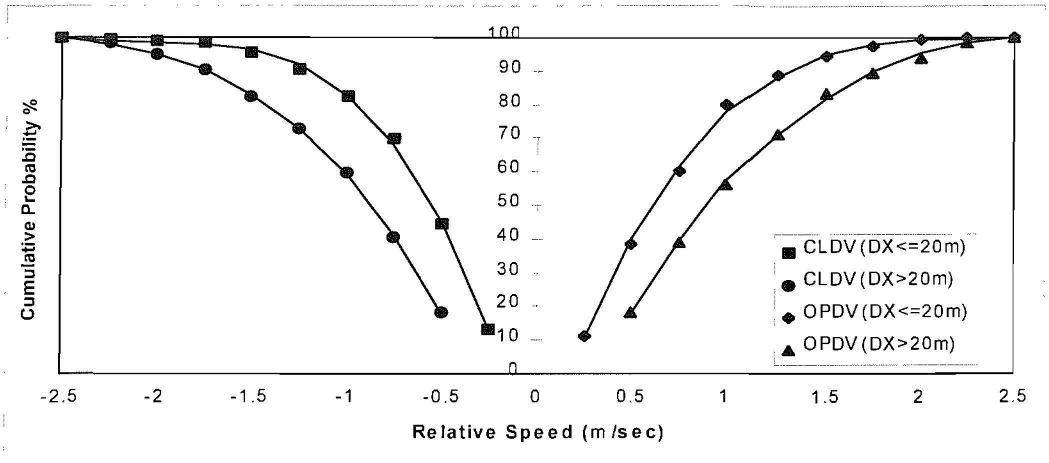


Figure 8.26: The Cumulative percentage for the CLDV and OPDV with the fitted curves.

- ***The Distribution Fitting and K-S test :***

The K-S [Kolmogorov - Smirnov] test the goodness of fit of a proposed distribution to the data. It was found that the Lognormal Distribution gave the best fitted distribution for the data. Table 8.1 present the significant level of K-S test for the data which has to be more than 0.05 to accept the distribution.

	CLDV		OPDV	
	(DX≤20m)	(DX>20m)	(DX≤20m)	(DX>20m)
Mean	0.656	0.981	0.716	1.008
Std	0.465	0.576	0.509	0.605
K-S test	0.297	0.44	0.091	0.164

Table 8.1: The K-S test of the Lognormal distribution for the Data

The lognormal distribution is acceptable one according to the K-S test. However, by plotting the quantity of the variable's distribution against the quantity of the lognormal distributions (see Figure 8.27) it is become clear the data has a significant deviation at the high end values:

(CLDV(DX≤20m) < -1.5 m/sec) (CLDV(DX>20m) < -2 m/sec)

(OPDV(DX≤20m) > 1.5 m/sec) (OPDV(DX>20m) > 2 m/sec)

In conclusion, the model will use the cumulative percentage method to generate the values of CLDV and OPDV. Thus, every driver in the model will have another two random numbers [Rand3 and Rand4] with a value between [0 to 1]. The first, [Rand3], is to generate the variation in CLDV, and the second, [Rand4], is to generate the variation in OPDV. However, Rand3 [Rand4] will have different values whenever the driver crossed the threshold OPDV [CLDV]. In this way the distribution of CLDV [OPDV] will be generated over the time of simulation. Figures 8.28 and 8.29 show the flowcharts of the subroutines that generate CLDV and OPDV respectively.

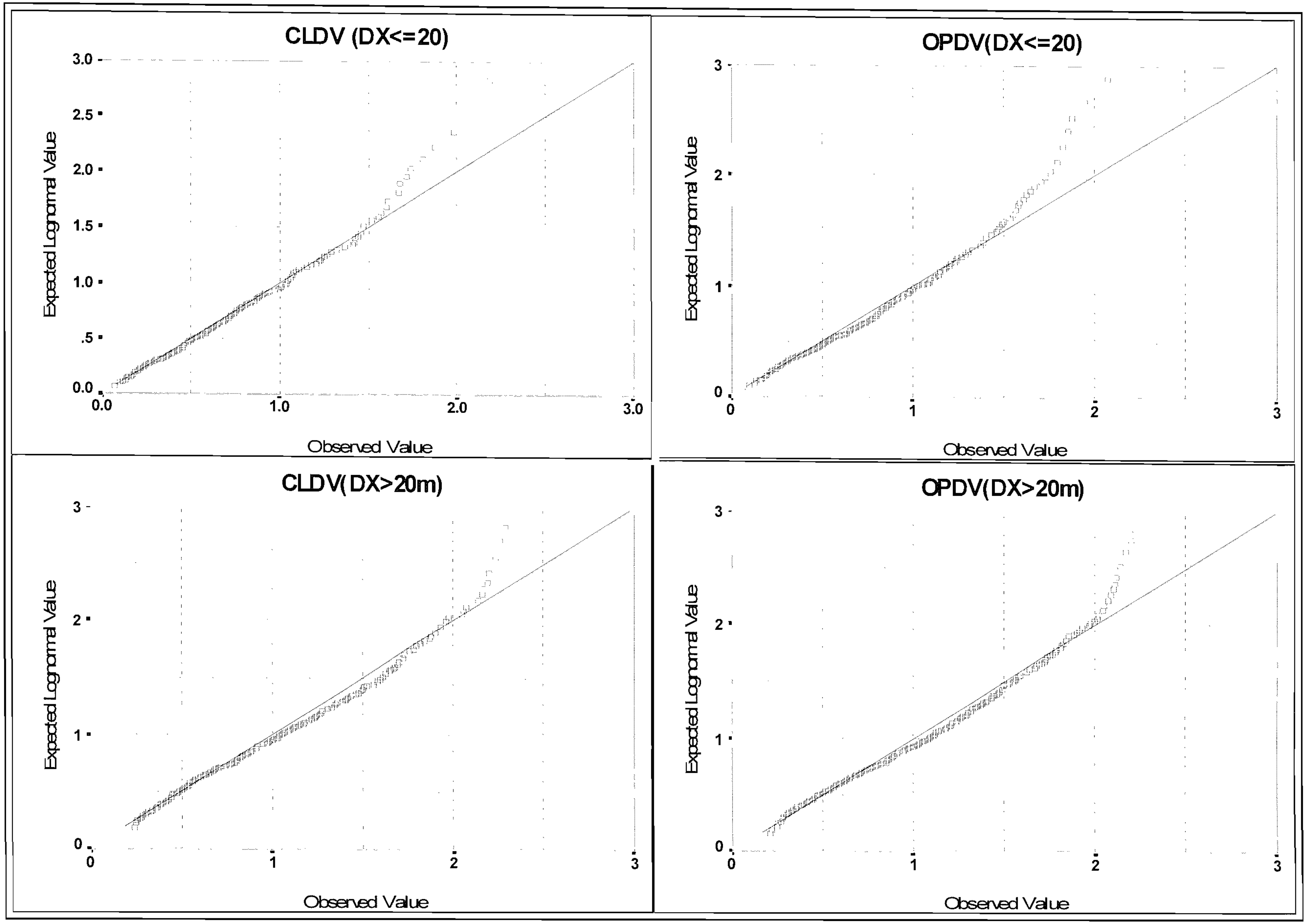


Figure (8.27): The lognormal probabilit fit for CLDV and OPDV.

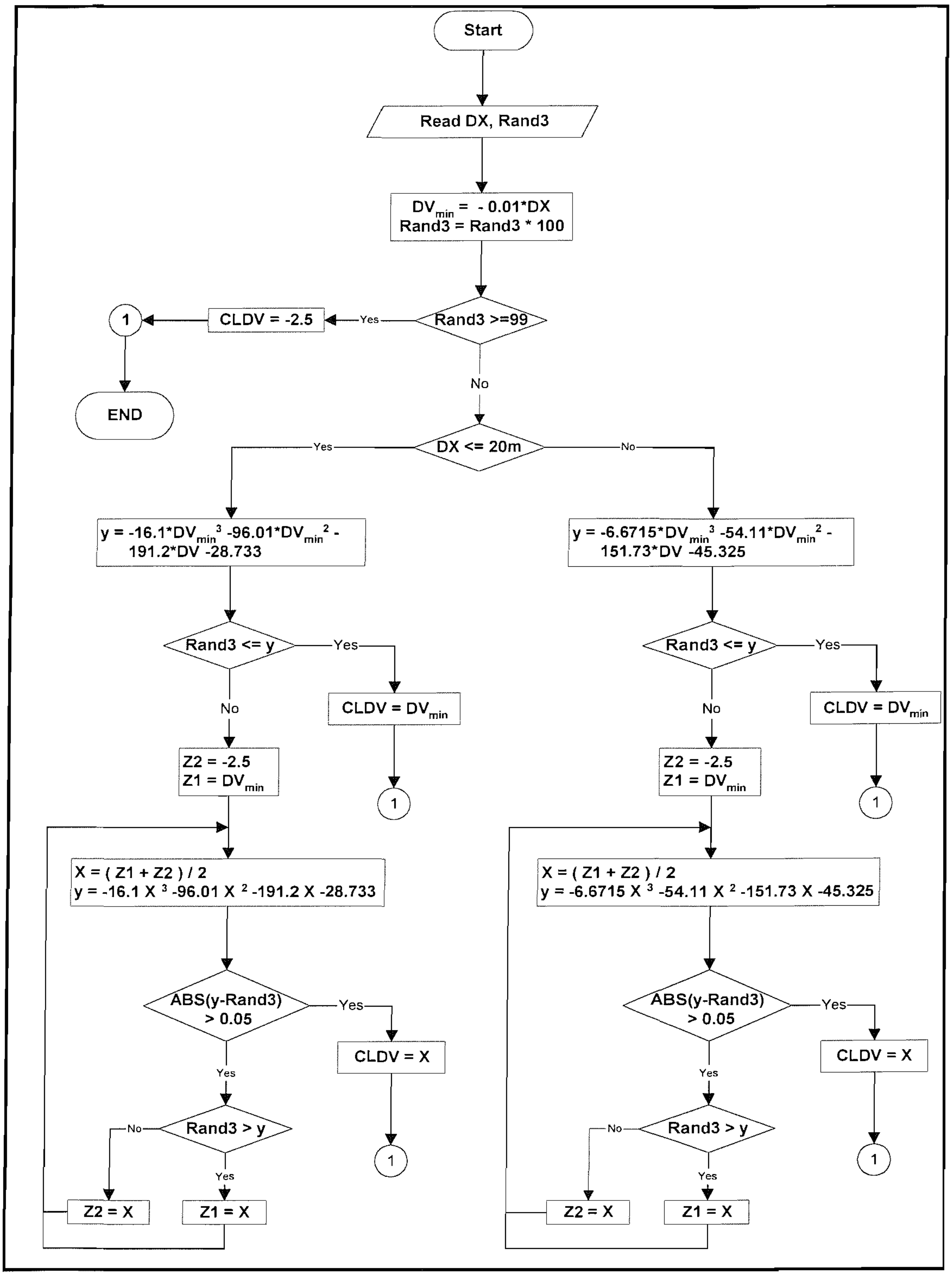


Figure 8.28: The CLDV threshold subroutine.

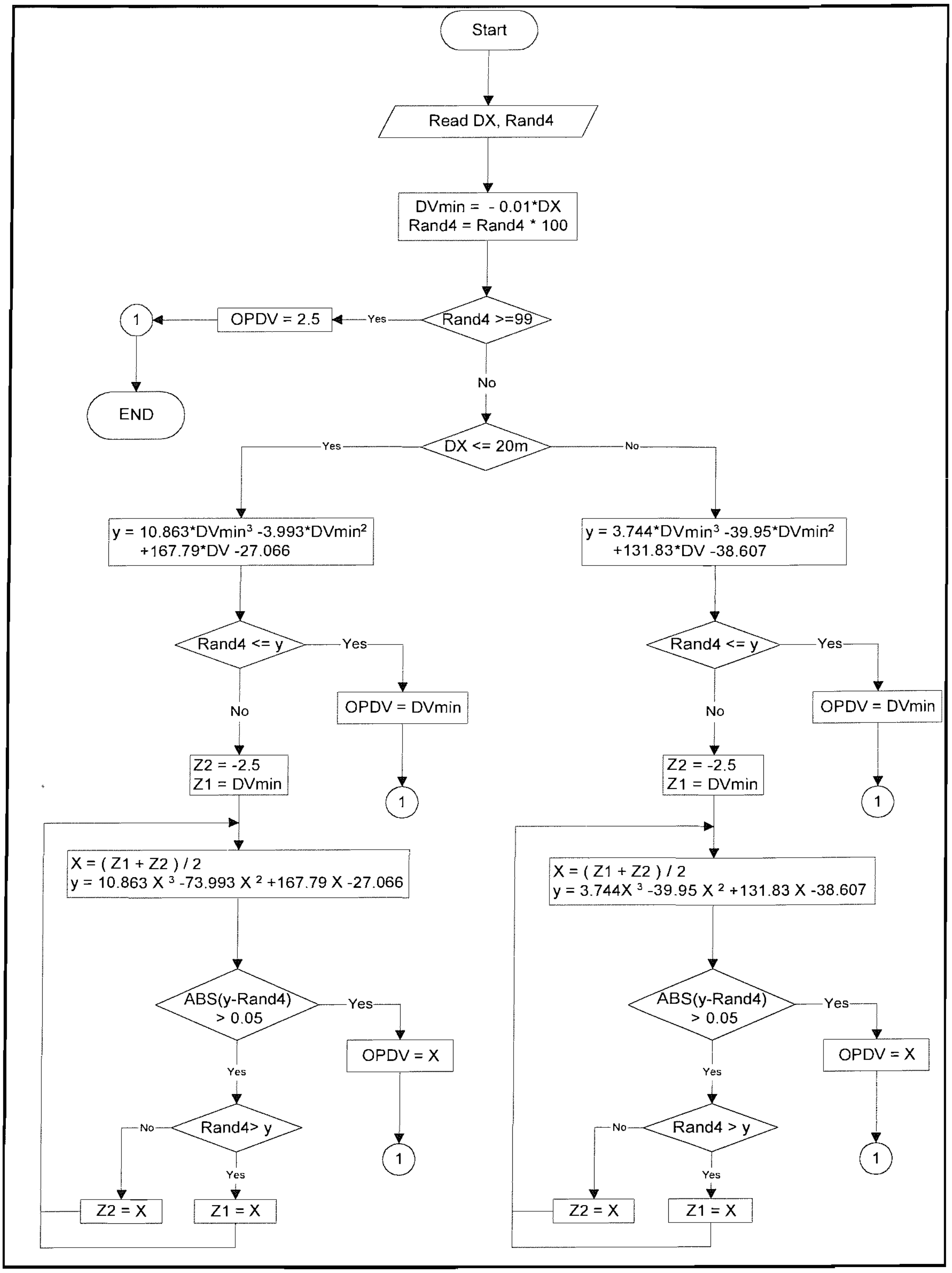


Figure 8.29: The OPDV threshold Subroutine.

8.6 Drivers Deceleration / Acceleration Noise In Constant Speed And Calm Close Following :

Previous pioneer research has indicate that a driver cannot maintain a fixed speed (MONTROLL 1958)[76] and the acceleration noise was found to be Gaussian distributed with dispersions of [0.1 ± 0.02 m/s^2]. The data at that time was collected using mechanical instrumented vehicle in which the accuracy and reliability of the data was uncertain. Later, WIEDEMANN and REITER revealed that the action points' deceleration (DC_{NULL}) / acceleration (AC_{NULL}) level in calm close following situations were found to be normally distributed [m: 0.2 - σ: 0.1 m/s^2] (WIEDEMANN, R. & REITER, U. 1992)[28]. However, their results were not supported by any explanation of the approach methodology and the tools they used in measuring their data. The TRG IV has helped to resolve the questions about drivers acceleration noise in constant speed situations and the values of the deceleration (DC_{NULL}) / acceleration (AC_{NULL}) in calm close following situations.

8.6.1 Drivers Acceleration Noise In Constant Speed Situations:

Two subjects aged 30 to 40 years were instructed to drive the IV in a constant speed on a level section of motorway (ie M27 near New Forest, Southampton). The experiment was conducted when the driver had a clear road and the nearest vehicle in front was at least 300m. The speed and acceleration time series were isolated from the data files and analysed. Figures 8.30 and 8.31 show samples of the speed and acceleration time series. The data analysis has revealed that both drivers have a normal distributed acceleration noise of [m=0.0 , Std=0.07 m/s^2], and this was expected. Figure 8.32 present a frequency histogram for the acceleration for the two test drivers. Nevertheless, by examining the distribution fitting deviation graph Figure 8.33, it becomes clear that the data is not normally distributed when acceleration level becomes over twice the standard deviation ±0.14m/s^2 , and hence, this value was chosen to be the minimum controllable deceleration / acceleration for a driver.

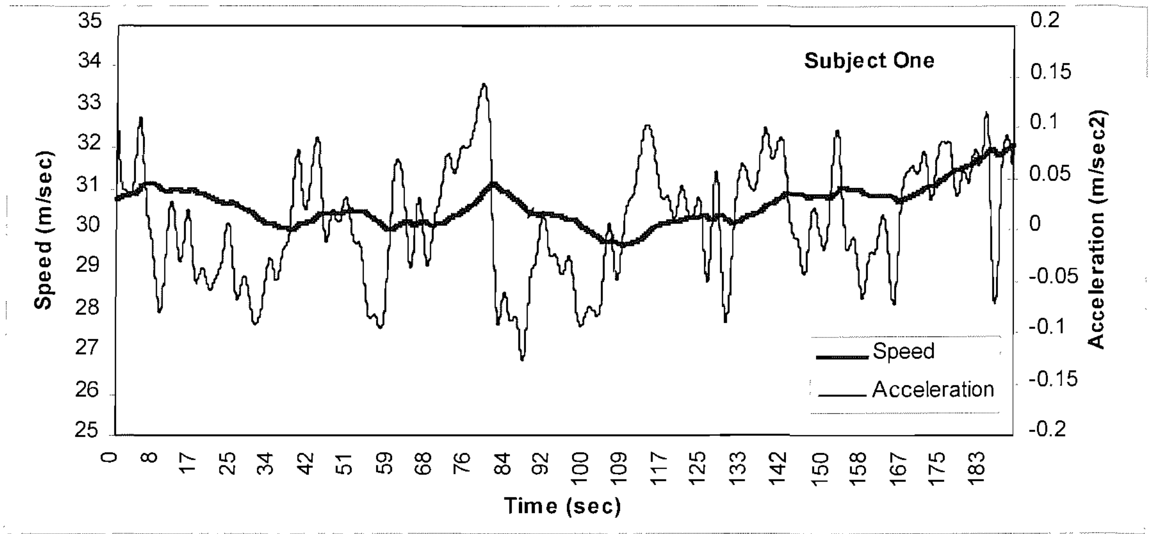


Figure 8.30: A sample of the speed and acceleration time series for Subject one.

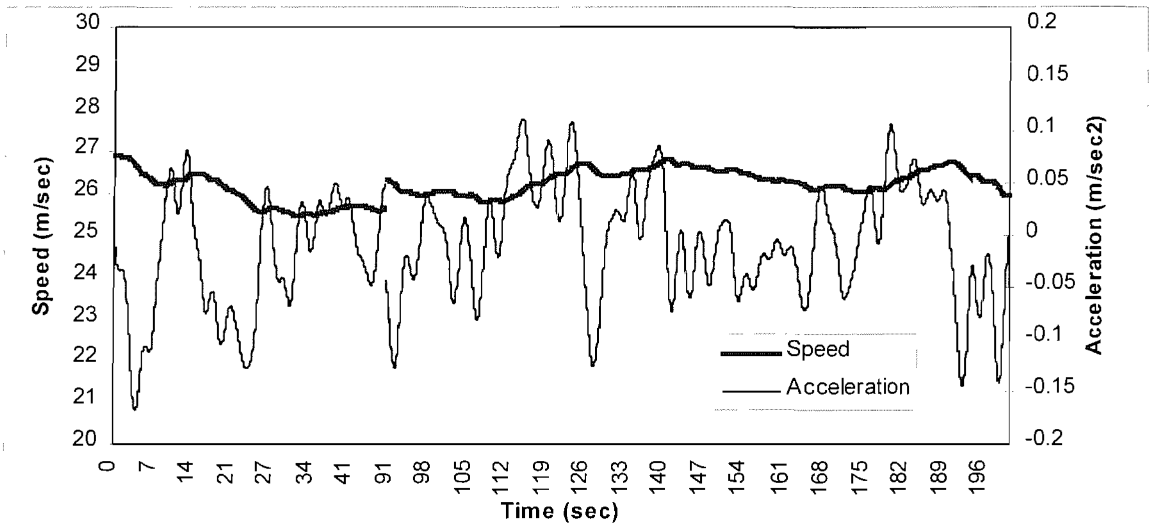


Figure 8.31: A sample of the speed and acceleration time series for Subject Two.

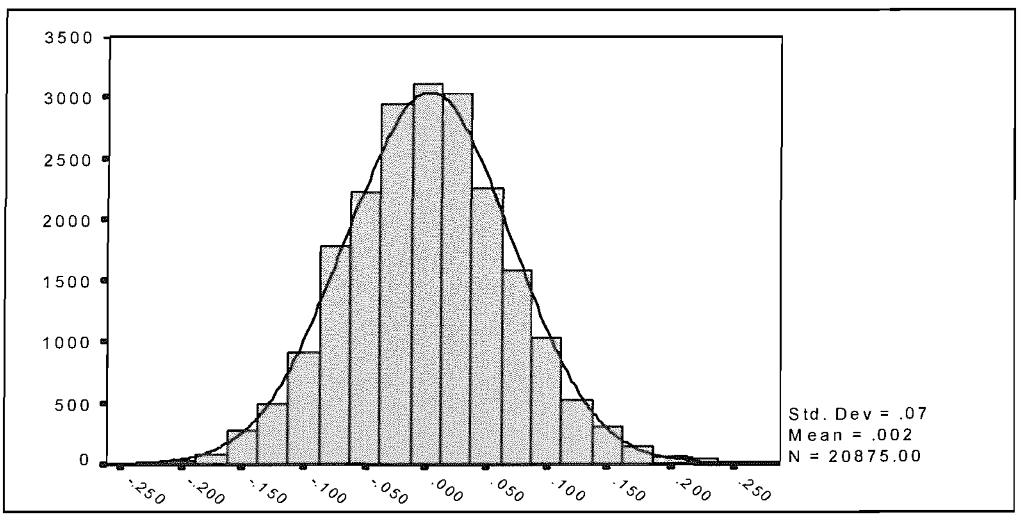


Figure 8.32: The Normal distribution and frequency histogram for drivers' acceleration noise in a constant speed

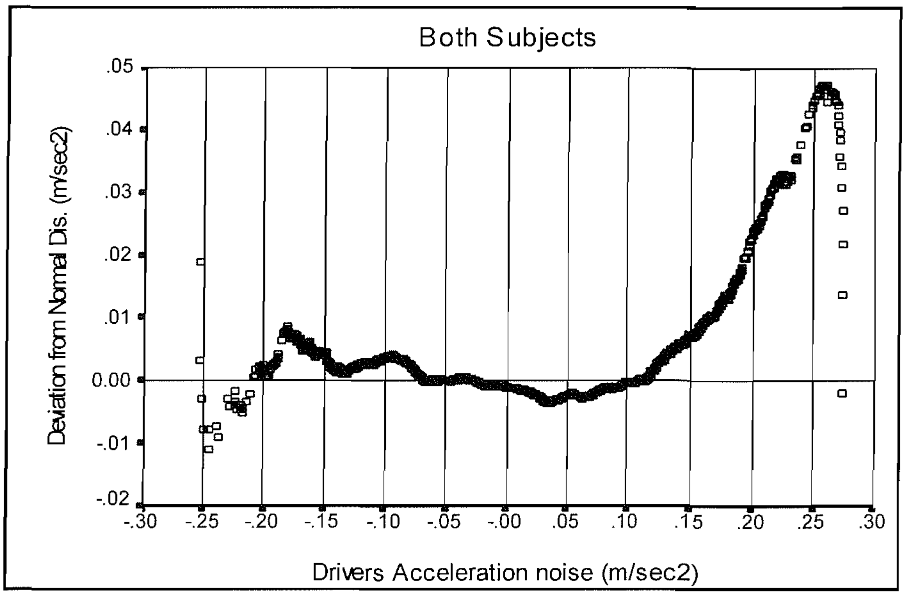


Figure 8.33: The Deviation from the Normal Distribution for the drivers' acceleration noise in a constant speed.

This experiment also, offers the opportunity to investigate drivers' sensitivity to change in speed level. This is especially useful in modelling drivers' behaviour in achieving their desired speeds. The procedure was to calculate the average speed for each series of data, and then the deviation from this average was computed for every reading. It was found that this deviation had a normal distribution with parameters: Subject (1) [m: 0.002 & std: 0.56 m/s], Subject (2) [m: -0.01 & std: 0.58 m/s] and both Subjects [m: -0.006 & std: 0.57 m/s], the distribution fitting is presented in Figures 8.34, 8.35 and 8.36.

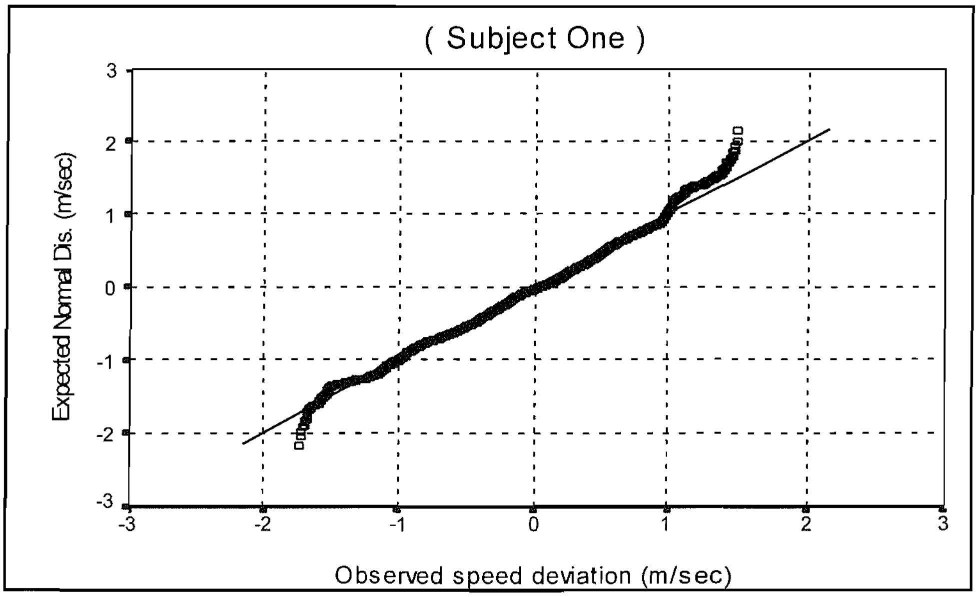


Figure 8.34: Normal distribution fitting plot of the speed deviation (Subject one).

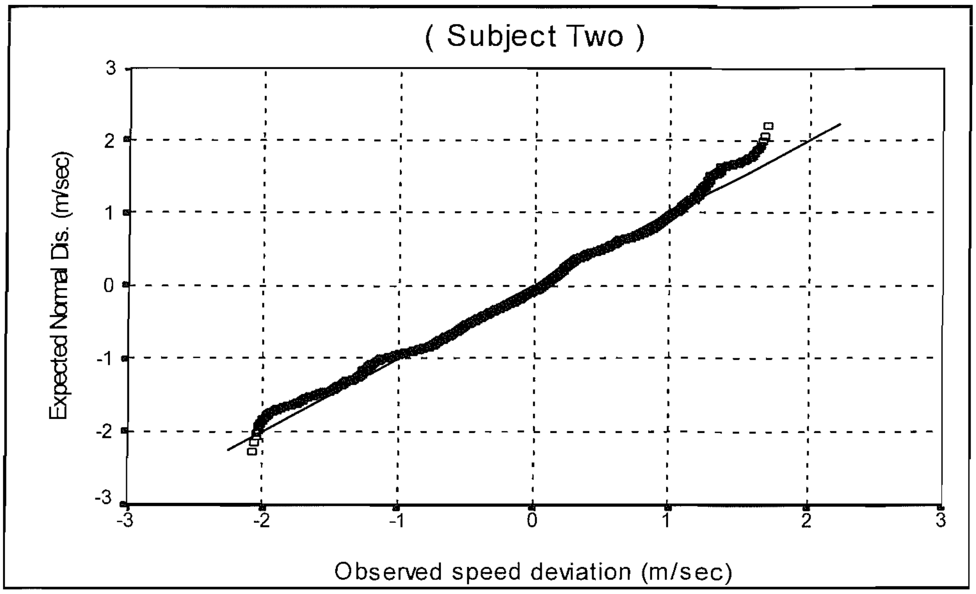


Figure 8.35: Normal distribution fitting plot of the speed deviation (Subject Two).

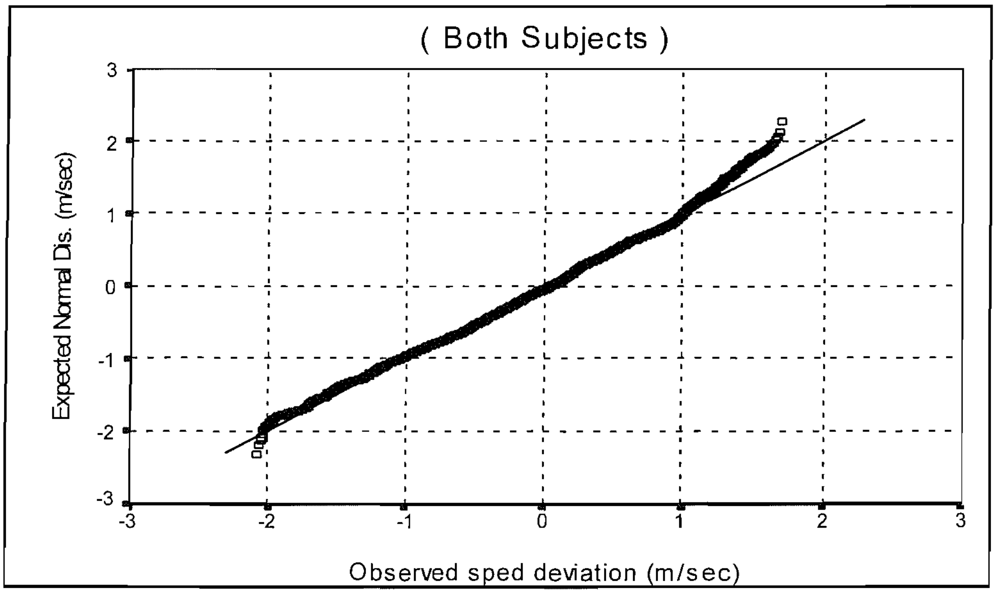


Figure 8.36: Normal distribution fitting plot of the speed deviation (both Subjects).

However, as for the acceleration noise, a value of double the standard-deviation (±1.14 m/s) was chosen to be the drivers sensitivity threshold to a constant speed.

8.6.2 Drivers Acceleration / Deceleration Levels In Calm Following Process:

The previous experiment had helped in identifying the minimum controllable deceleration or acceleration. However, this did not clarify the level of deceleration or acceleration adopted by drivers in calm close following. In order to measure this deceleration / acceleration the subject had to drive the IV and follow another vehicle, while the Radar has to be placed in front to measure the relative speed and distance. (This did not affect the data shape in the output file and the way in which the data filtering and smoothing was conducted). Several subjects were asked to follow a white van on a motorway in real traffic, while the van driver was instructed to drive in a constant speed. The data analysis procedure is similar to that of the CLDV and OPDV thresholds:

1- By assessing the simi-spirals when DV < 0 (DV > 0) the maximum level of deceleration DC (acceleration AC), the relative speed [absolute values] and the corespondent relative distance were recorded in an output file.

2- Then, every reading where DC > -0.14 m/s^2 or AC <0.14 m/s^2 was discarded, because drivers can not control their decelerations / accelerations under these levels (Result from the previous experiment).

The data then was analysed in three different aspects:

- **The Distributions :**

This approach followed that of WIEDEMANN and REITER (WIEDEMANN, R. & REITER, U. 1992)[28], although, the data did not show the same normally distributed results. It was indicated from the frequency histogram (Figures 8.37 and 8.38), that the deceleration had a uniform distribution over the range [-0.14 ... -0.45 m/s^2], the acceleration had a rather clear trend in frequency. The cumulative percentage was used to model these distribution. These results do not reveal whether or not the relative speed level or the (DX/DV) ratio level influenced deceleration or acceleration. This leads to the second test.

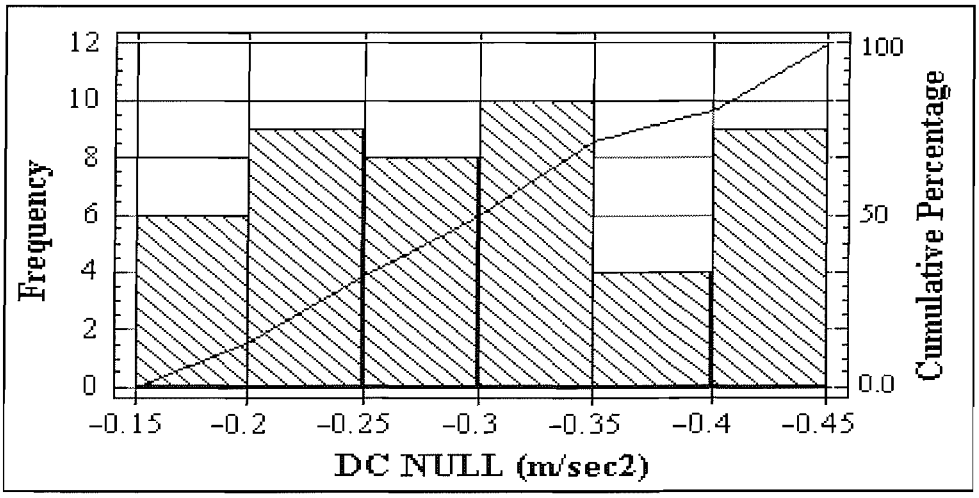


Figure 8.37: The frequency histogram of deceleration in calm close following.

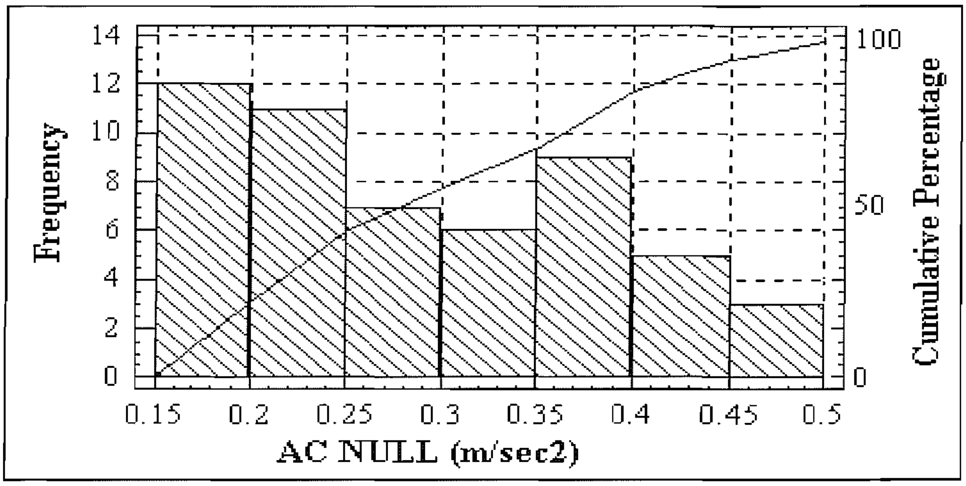


Figure 8.38: The frequency histogram of Acceleration in calm close following.

- **The Influence Of Relative Speed :**

By plotting the deceleration data against the relative speed DV, the influence becomes clear (Figure 8.39). The data shows a threshold that increases the level of deceleration with an increase in relative speed. The data is scattered above this threshold. A linear equation 8.18 was fitted to this threshold, which can be used in modelling this particular influence. Figure 8.39 presents the relation between DV and DC and the suggested threshold.

$$DC = 0.1762 * DV - 0.0883 \qquad R^2 = 94\% \qquad ....8.18$$

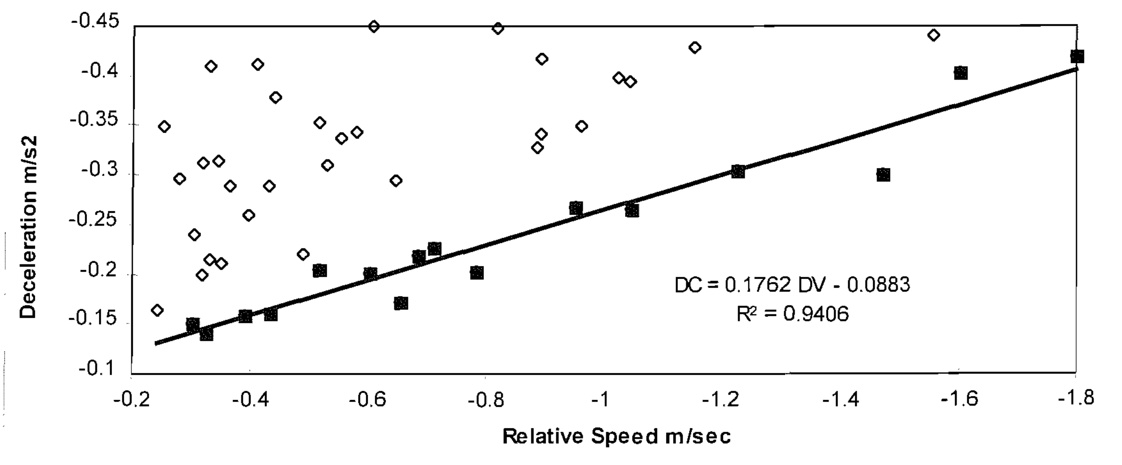


Figure 8.39: the relation between the deceleration and the relative speed in calm close following

There was no clear evidence that the relative speed had an influence on acceleration. An interpretation could be that drivers are not concerned with the relative speed level when the following situation is opening, although in closing situations concern is apparent because of potential collisions. Figure 8.40 presents the relationship between Ac and DV.

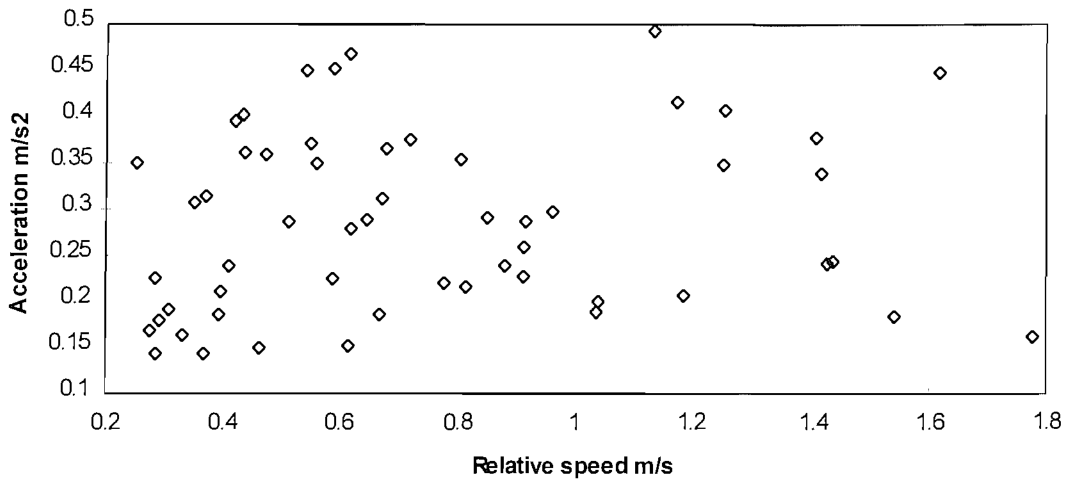


Figure 8.40: the relation between the Acceleration and the relative speed in calm close following

- **The (DX/DV) Ratio Influence :**

The (DX/DV) ratio as indicated earlier plays a crucial part in drivers perception, and effectively defines the time to collision in the closing situation (DV<0), but does not hold any physical meaning in the opening one (DV>0). Figures 8.41 and 8.42 show the relationship between the (DX/DV) ratio and the deceleration and acceleration respectively. In the closing situation a threshold was fitted to the data with an equation:

If TTC <= 65 $\Rightarrow$ DC_{th} =-8.876E-05* TTC^2 + 0.011939*TTC -0.55051
If TTC > 65 $\Rightarrow$ DC_{th} =-0.145 m/s^2 ....8.19

However, two data points were discarded perhaps because they represented driver misperceiving of the actual situation at that moment. The data is scattered above this threshold and the cumulative percentage formula, which was obtained before, may be used to represent the deceleration distribution over the proposed limits (equation 8.19).

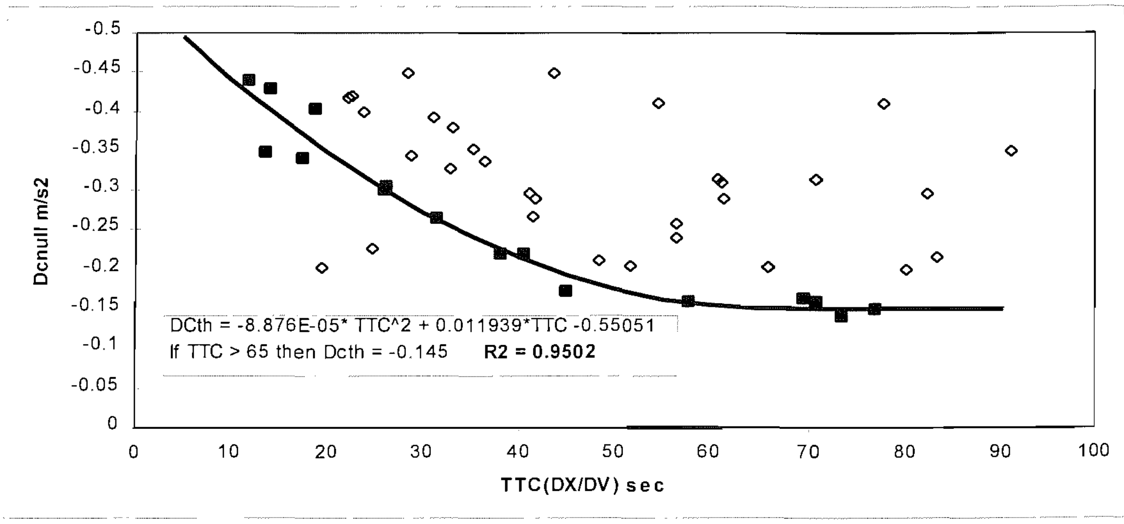


Figure 8.41: The (DX/DV) ratio and the deceleration in calm close following.

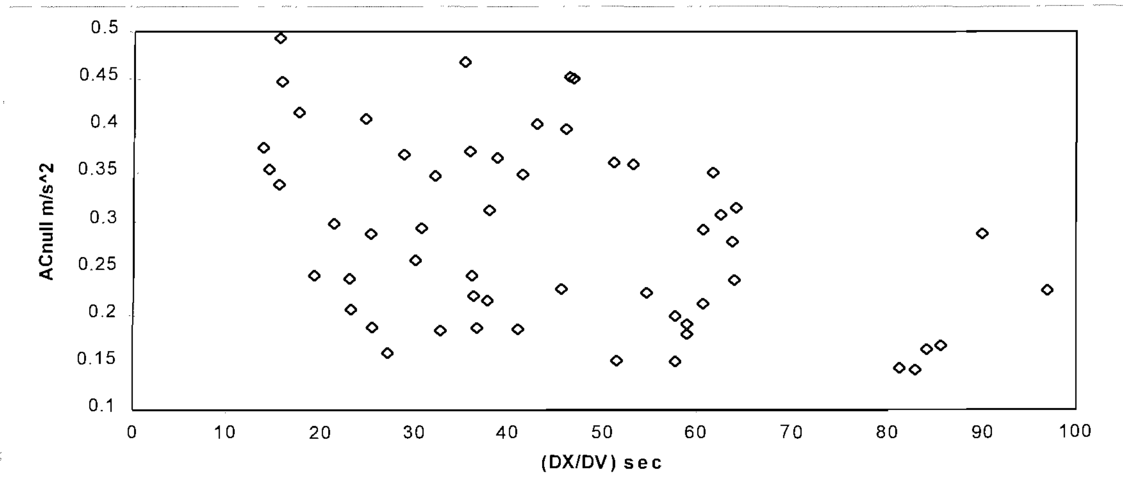


Figure 8.42: The (DX/DV) ratio and the acceleration in calm close following.

8.6.3 Conclusion:

The results from the previous two experiments can be summarised in five points:

1- Drivers cannot sustain a constant speed and their deceleration or acceleration is normally distributed [mean = 0.0, STD = 0.07 m/s^2].

2- Drivers' sensitivity limits to a desired constant speed is about [±1.2 m/s].

3- The drivers' minimum deceleration in calm close following increases in the absolute value as the TTC value increases (equation 8.19). However, this deceleration is uniformly distributed over the range [minimum value (-0.45 m/s^2)].

4- The drivers' acceleration in calm close following is distributed according to the cumulative percentage formula :

$$AC = 1.972E\text{-}05R^2 + 0.0019087R + 0.099295 \qquad ...8.20$$

Where:
R : is a random number between [0...100].
AC: is the acceleration in calm close following its value within the range [0.14...0.5m/s^2].

5- The ratio (DX/DV) does not have any influence on the drivers' acceleration in calm close following.

These results are very useful for the model in this research, although further research with a larger sample of subjects is needed.

8.7 The Improved Car Following Model:

The improved model has to be more efficient and clearer about driver behaviour during the following process. The mechanism in the MISSION model generates high speed and acceleration variations among the vehicles in a platoon. To improve the model, several modifications were made.

8.7.1 The Perceived Deceleration or Acceleration of the vehicle ahead:

It was assumed in the model that drivers will perceive the acceleration / deceleration of the lead vehicle when they are above the perceptual threshold [Closing or Opening]. In order to determine this perception, two situations were considered:

- Firstly, when a driver is in closing situation [$DV < DV_{close}$] he or she will perceive only the deceleration of the lead vehicle, and the model will use the equation 8.21 to calculate its value. This equation was presented in (Section 7.4.1, Chapter 7) of this thesis :

$$DC_L = \frac{-(DTC + t)}{TC_0 + DTC + 0.5 * t} * \frac{DV_0}{t} + DC_F \qquad ....8.21$$

Where :
DV : Is the relative Speed at moment T + t .
DV_0 : Is the relative Speed at moment T .
DC_L : Is the leader deceleration at moment T.
DC_F : Is the Follower deceleration at moment T.
TC_0 : Is the time to collision at the moment T (TC = $-DV_0$ / DV_0).
DTC : Is the change in the time to collision during the Epoch t (DTC = TC - TC_0).

t : Is the time interval.

In this equation the model uses the estimated time to collision ETTC instead of TTC, and the estimated relative speed EDV instead of DV :

$$EDV = DX / ETTC \qquad .....8.22$$

Where :

EDV : The estimated relative speed m/sec.

DX : The relative Distance m.

ETTC : The estimated time to collision calculated from the equation 8.23, which was discussed previously in (Section 7.3.23, Chapter 7):

$$ETTC = -0.18 + 0.767 * TTC \qquad .....8.23$$

In the following process, where the spacing distance is less than ABX, the concept of the underestimation of TTC has not been used in estimating the deceleration of the lead vehicle, and the model assumes that the driver will perceive the real value from the lead vehicle. This is because of two reasons:

1- All the experiments that have been undertaken to estimates time to collision, used the concept of the approach process, where the relative distance is large [more than 30m]. Therefore, we cannot relay on this result in the emergency situations.

2- Research into estimating time to collision gave very varying results due to the complexity of the estimating process and the factors that effect it. Consequently, using this idea will lead to a high rate of braking in the emergency process.

- Secondly, when a driver is in opening situation [$DV > DV_{open}$], he or she will perceive only the acceleration of the lead vehicle. The model will consider the real value of the acceleration, because the concept of TTC or (DX/DV) is not clear in these circumstances.

There are limits to deceleration / acceleration perception. The absence of these limits will lead to high variation in the relative speed in a platoon of vehicles, and the model will be very sensitive to any 'vibration' in the platoon. These limits were assumed to be the maximum deceleration and acceleration in calm close following observed in previous experiment (Section 8.6.2):

- (Following Process) $+0.5 > DC_L > -0.5$ m/sec^2

However, for the emergency process where driver beyond ABX threshold the limits become:

- (Emergency Process) $+0.25 > DC_L > -0.25$ m/sec^2

Finally, a driver will not react until he or she crosses the relative speed threshold CLDV or OPDV. This enables the model to simulate the variation in reaction time between drivers. Figure 8.43 presents the procedure of how the perceived deceleration or acceleration is determined by the model.

8.7.2 The Maximum Deceleration [DC_{max}]:

When a lead vehicle is decelerating the driver should not decelerate harder than a maximum level DC_{max} , which should be determined according to the current relative distance, the relative speed and the absolute speed. This will produce a smooth transition from the following process to the emergency process.

To calculate DC_{max}, the model will use the same equations of the Approach Process [Section 7.4.2, Chapter 7]. However, a value of DC_{NULL} will be added in order to achieve some uncertainty in the driver's behaviour:

$$DC_{\max} = \frac{DC_L * V_F^2}{V_L^2 - 2 * DC_L * V_F * \Delta T} + DC_{NULL} \qquad \ldots.8.24$$

Where:

DC_L : is the deceleration of the lead vehicle.
V_L : is the Speed of the lead vehicle [$V_L = EDV + V_F$].
V_F :is the speed of the current vehicle.
ΔT :is the available safe time :

$\Delta T = (DX + DV*(\text{reaction time}) + 0.5*DC_L *(\text{reaction time})^2 - AX) / V_F$

- Where [$DX \geq ABX$], If [$(DC_{max}) > (0.5*EDV^2 /(AX-DX))$], the driver will adopt the value : [$(DC_{max}) = (0.5*EDV^2 /(AX-DX))$].

- Where [$DX < ABX$], If [$(DC_{max}) > (0.65*EDV^2 /(AX-DX))$], the driver will adopt the value : [$(DC_{max}) = (0.65*EDV^2 /(AX-DX))$]. Another DC_{NULL} will be added to the value if the spacing distance is less than [(ABX - AX)/2].

- The maximum level of deceleration that can be achieved by a driver was considered to be -9 m/sec^2 (WOOTTON & JEFFREYS 1990)[29]. Thus, (DC_{max}) should not be less than or equal to -9 m/sec^2 . Figure 8.44 presents the flow chart of the DC_{max} procedure in the model.

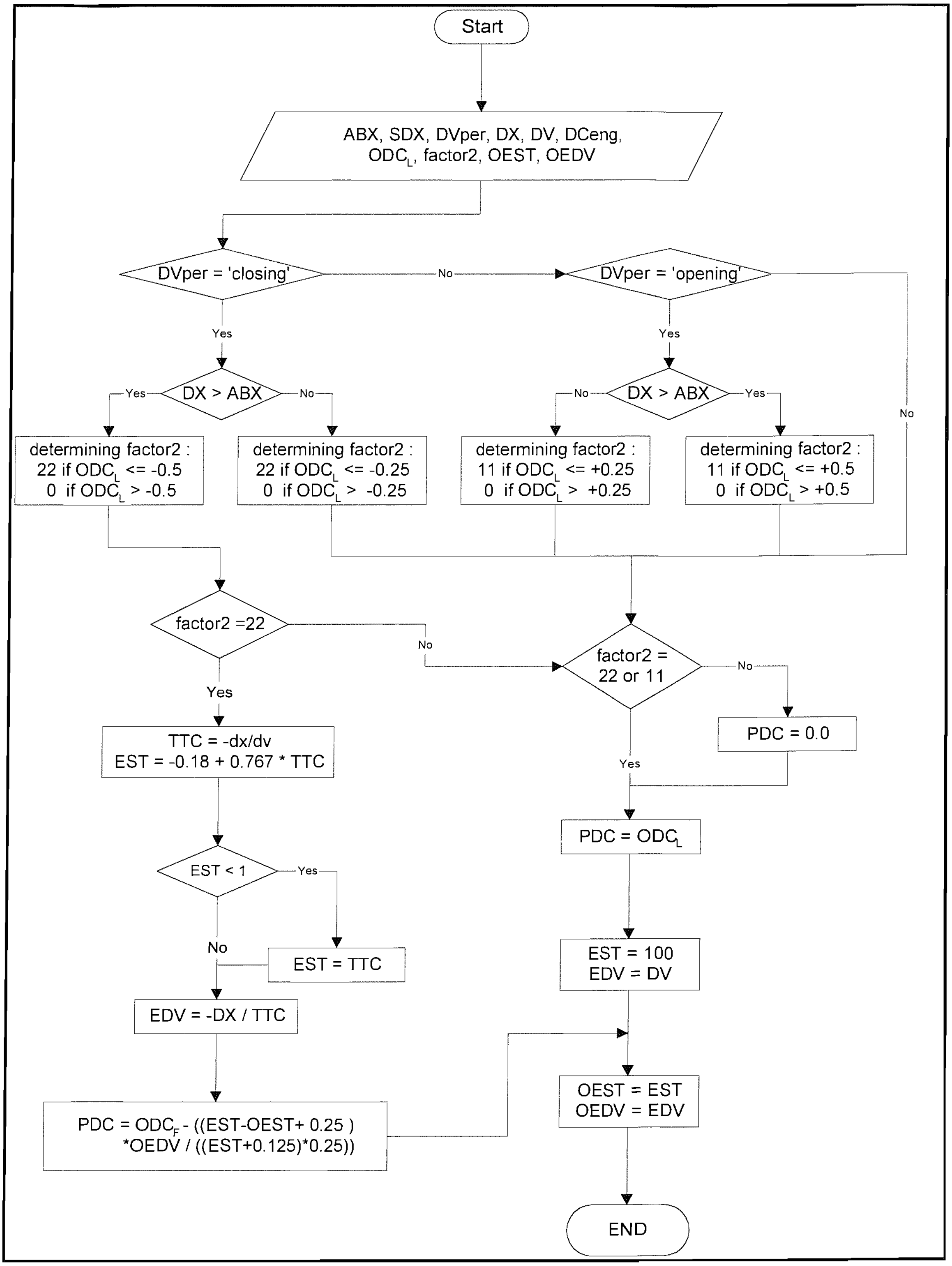


Figure 8.43: The Flow chart for determining the perceived deceleration/acceleration.

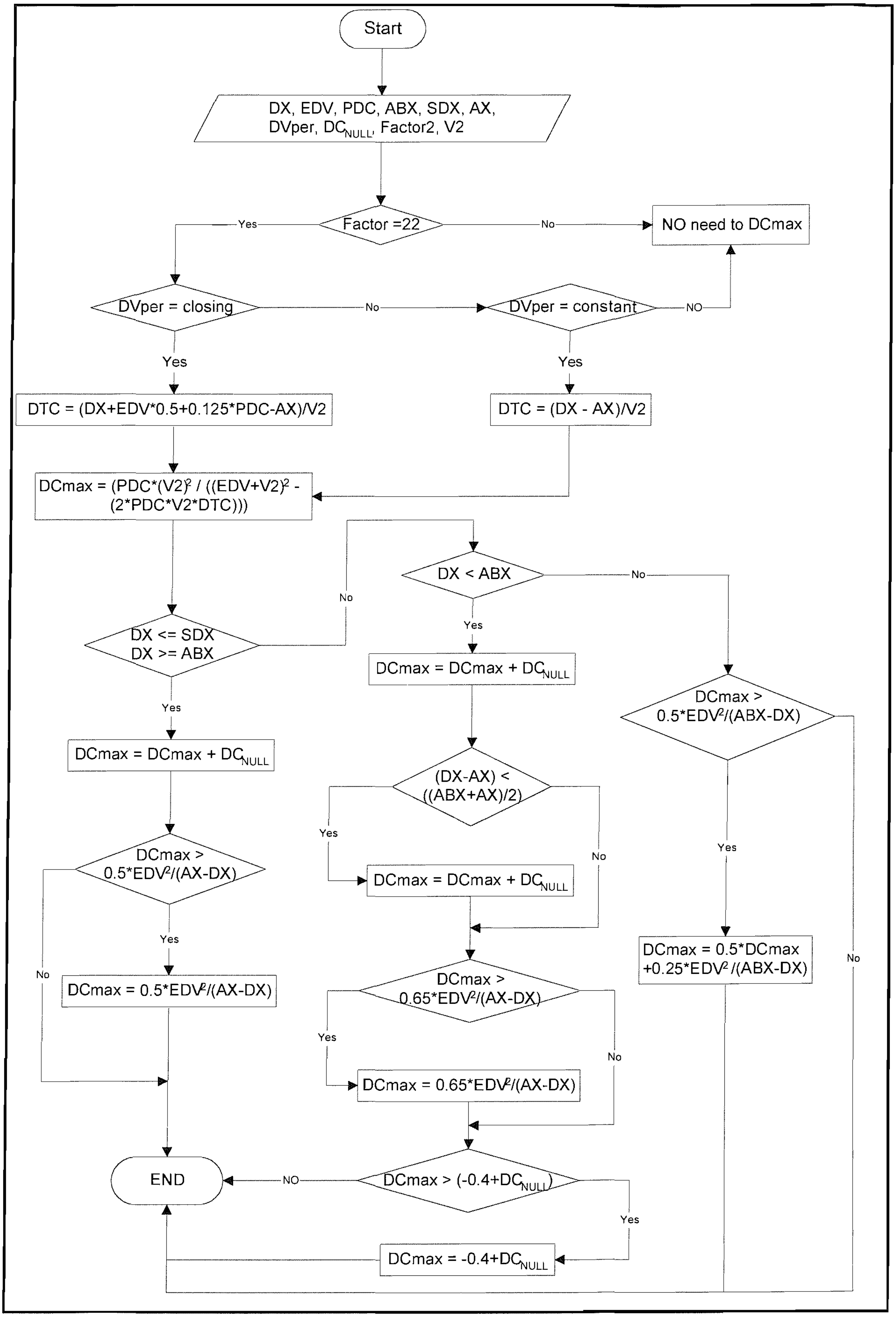


Figure 8.44: The procedure of determining DC_{max} in the Model.

8.7.3 The Influence of the vehicles ahead of the lead one:

It is usual to assume that the First Vehicle Ahead (FVA) has the major influence in the following process, although in practice, it is likely that drivers perceive information about the traffic ahead of the lead vehicle.

What type of information do drivers perceive from the vehicles ahead?

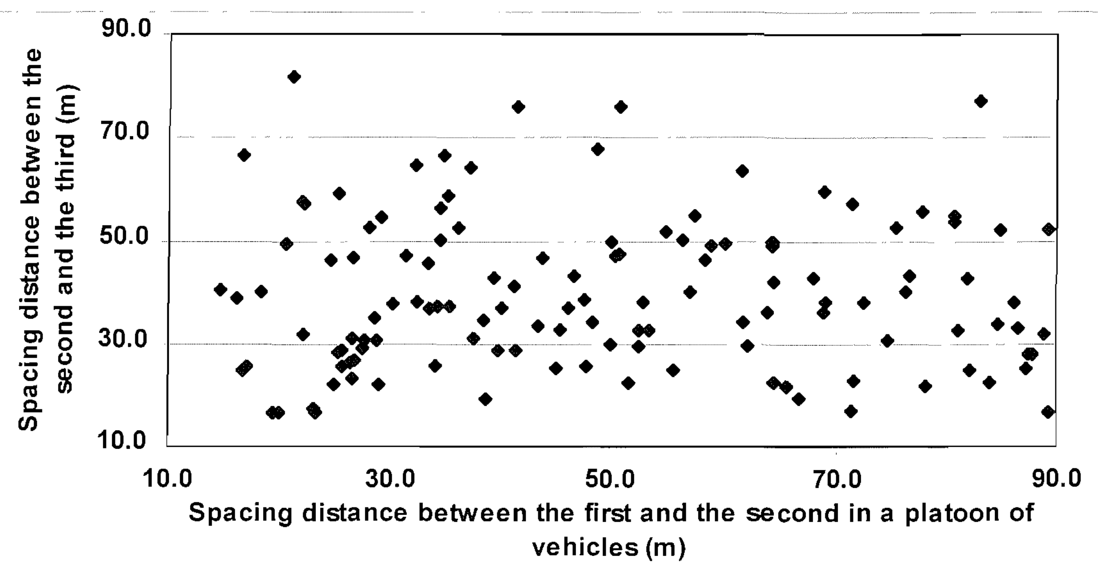


Figure 8.45: the relation between the relative distance of the leader and the one of the Follower.

Drivers may perceive much information from the vehicle ahead of the lead one. This may included vehicle type, direction, braking lights situation relative distance. Calibrating these factors for a model is not easy. However, an attempt was made to investigate any relationship between the following distance of the leader and the following distance of the follower in a platoon of vehicles. A video camera fixed on pedestrian bridge over a motorway was used in the experiment, then the time, speed and vehicle type were recorded as each passed a virtual detector point on large TV screen. Unfortunately, the data could not show relationship between the following distance of the leader and the follower in a platoon of vehicles. Figure 8.45 shows the scattered points for this relationship. However, it was decided that the ability of the model to present the situation in which the brake lights of the leading vehicle could trigger a response in the second following vehicle.

How would a driver react to the brake light?

In order to examine this factor, a platoon of at least three instrumented vehicles is needed to perform the proper experiment and produce acceptable results. This is not possible at the current time, and only simple assumptions were used in the model to simulate the driver reaction for the brake light of the vehicle ahead of the one he or she follows. These are:

1- If the Brake lights of the vehicle farther ahead are ACTIVE and the lead vehicle is accelerating the model will use 60% of this acceleration to determine the driver decision.

2- If the Brake lights of the vehicle farther ahead are NOT ACTIVE and the lead vehicle is decelerating the model will use 80% of this deceleration to determine the driver decision.

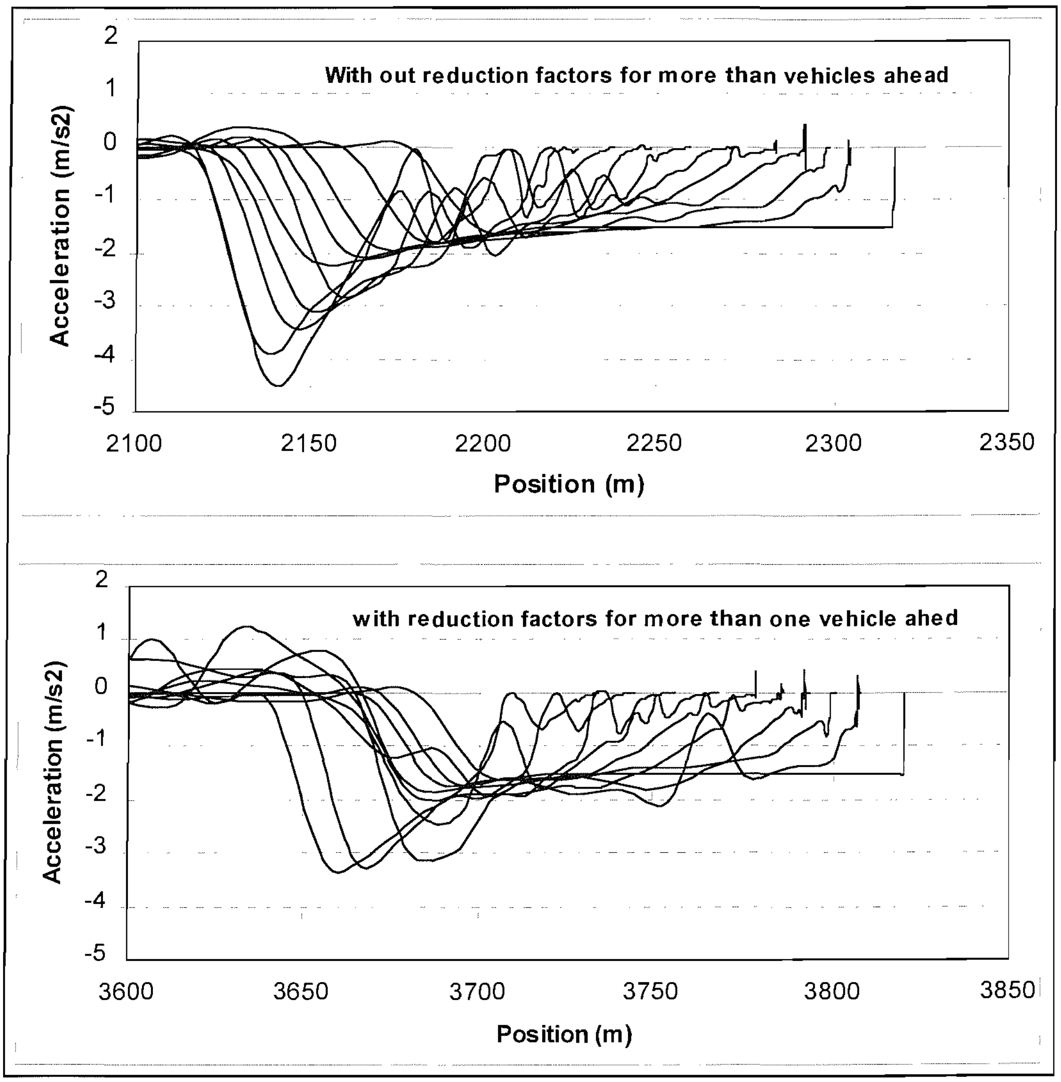


Figure 8.46: The model output with and without the reduction factors for more than one vehicle ahead

Justifying the two previous reduction factors is not easy, because the model contains so many dynamic variable and factors that influence its output. However, adding such logic to the model is good on the microscopic level where it may give some more realistic behaviour. Figure 8.46 presents two trails for a platoon of vehicles in which the leader decelerates to stop at -1.5 m/s^2. The first is without the reduction factors, while the second is with them. Clearly, in this example, the influence of the reduction factors seems to be obvious, although the initial situation is different (i.e. the spacing distances and the relative speeds between the vehicles in the platoon), as the sole influence of the reduction factors is very difficult to be achieved in such probabilistic model. Thus, whilst, the presence of the reduction factors add another feature to the model, a complete justification needs much better data to support or clarify this phenomenon.

8.7.4 The Emergency Process:

The emergency process takes place when the relative distance to the front vehicle becomes less than the ABX threshold. The driver will try to decelerate in order to regain the lost spacing. A minimum value of deceleration was adopted as [$-0.4 + DC_{NULL}$]. This value is close to the maximum level of deceleration measured in calm close following (Section 8.6.2), and the value of DC_{NULL} will generate the variation among drivers in this process. However, different circumstances were considered in this process:

- If the relative speed is less than DV_{close} , which means that the driver is closing rapidly, the model will calculate the deceleration by the equation:

$$DC_F = F1 * DV^2 / (AX - DX) + DC_L \qquad .....8.25$$

Where:

DC_L : is the real value of the lead vehicle deceleration or acceleration.

AX : The stopping distance adopted by the driver :
$AX = 1.5 + 1.5 * \text{Random}(0,1)$.

F1 : A safety factor which determines the avoidance level of a collision course, this was discussed by Lee where he reported that the driver will not be on a collision course if the value of F1 was more or equal to (0.5) (Lee D. 1976)[54]. The model uses a value of F1 = 0.65 to simulate drivers intention to avoid emergency situation as quickly as possible. However, the F1 factor will take a value of (0.75), if the spacing distance becomes less than [(ABX - AX)/2], ie harder deceleration occurs.

- In other circumstances the model will use the minimum deceleration plus the real value of deceleration / acceleration of the leader.

Finally, the concept of the maximum braking (equation 8.24) is still applied, but adding another DC_{NULL} to its value if the spacing distance becomes less than [(ABX - AX)/2]. The flow chart for this process will be illustrated in the next paragraph.

8.7.5 The Model Structure:

The flow charts for the procedures to determine the driver decision process during the following process is presented in Figures 8.47, 8.48 and 8.49. These incorporate The thresholds and features described above.

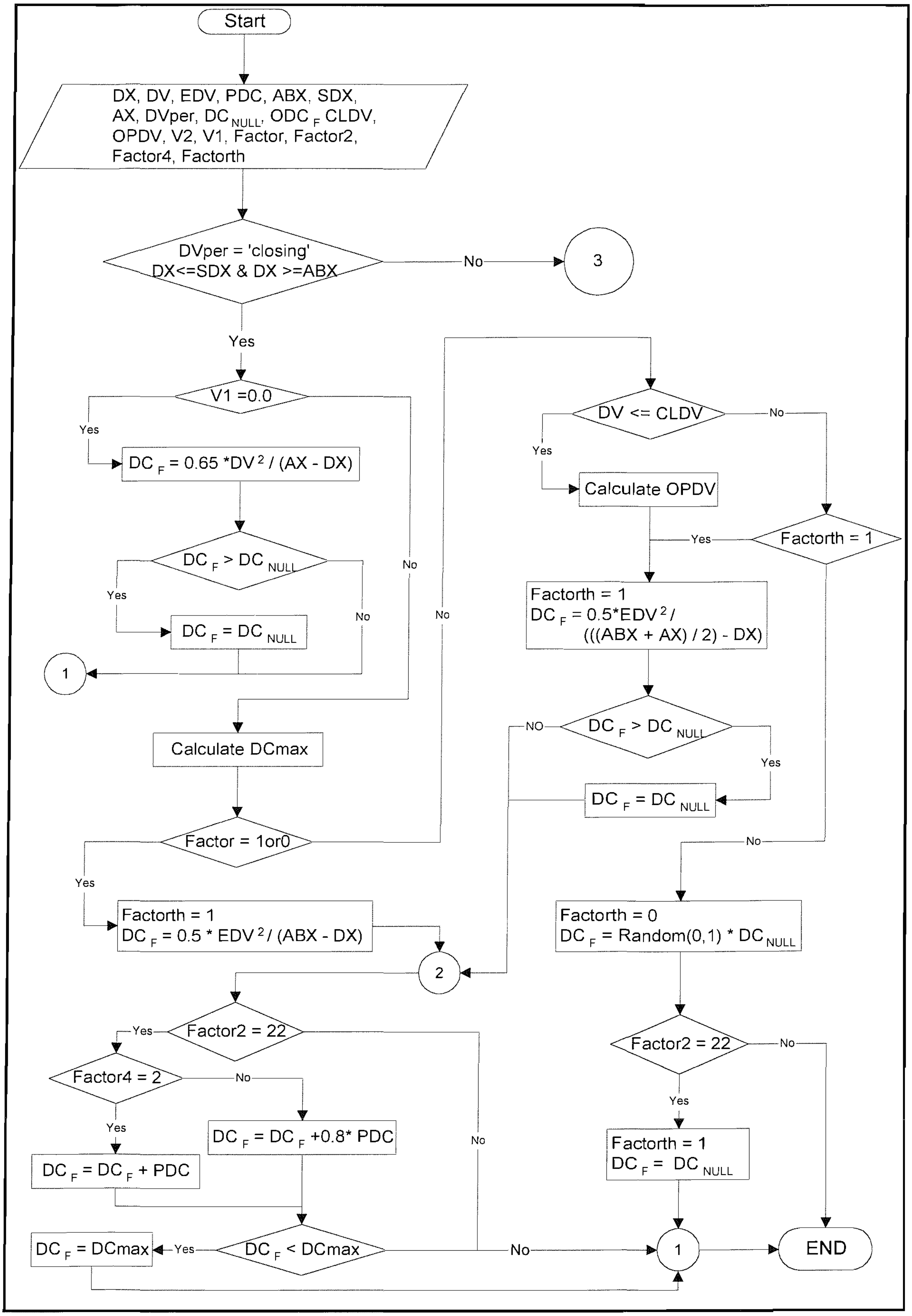


Figure 8.47: The Car following procedure Part (1).

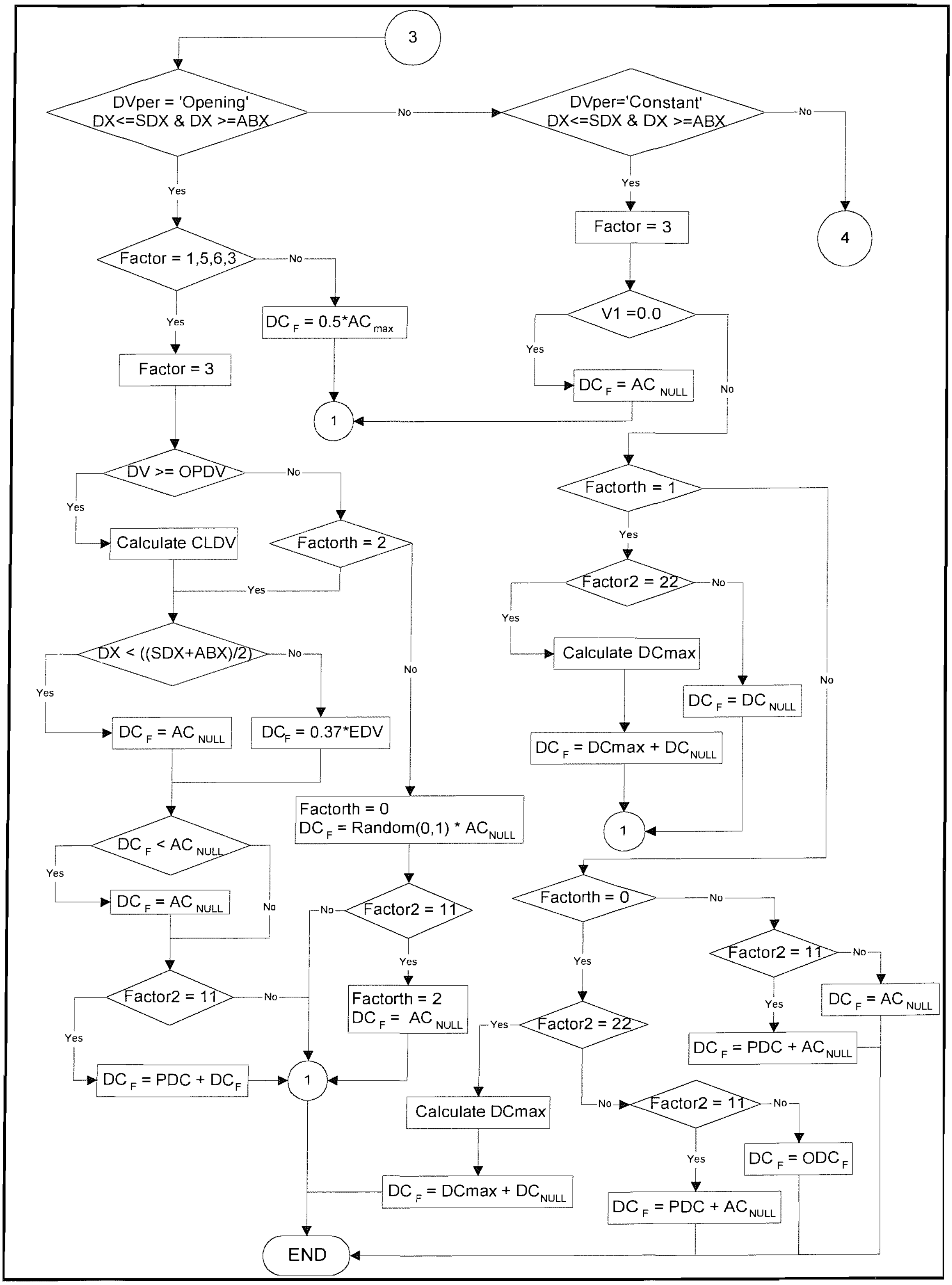


Figure 8.48: The Car following Procedure Part (2)

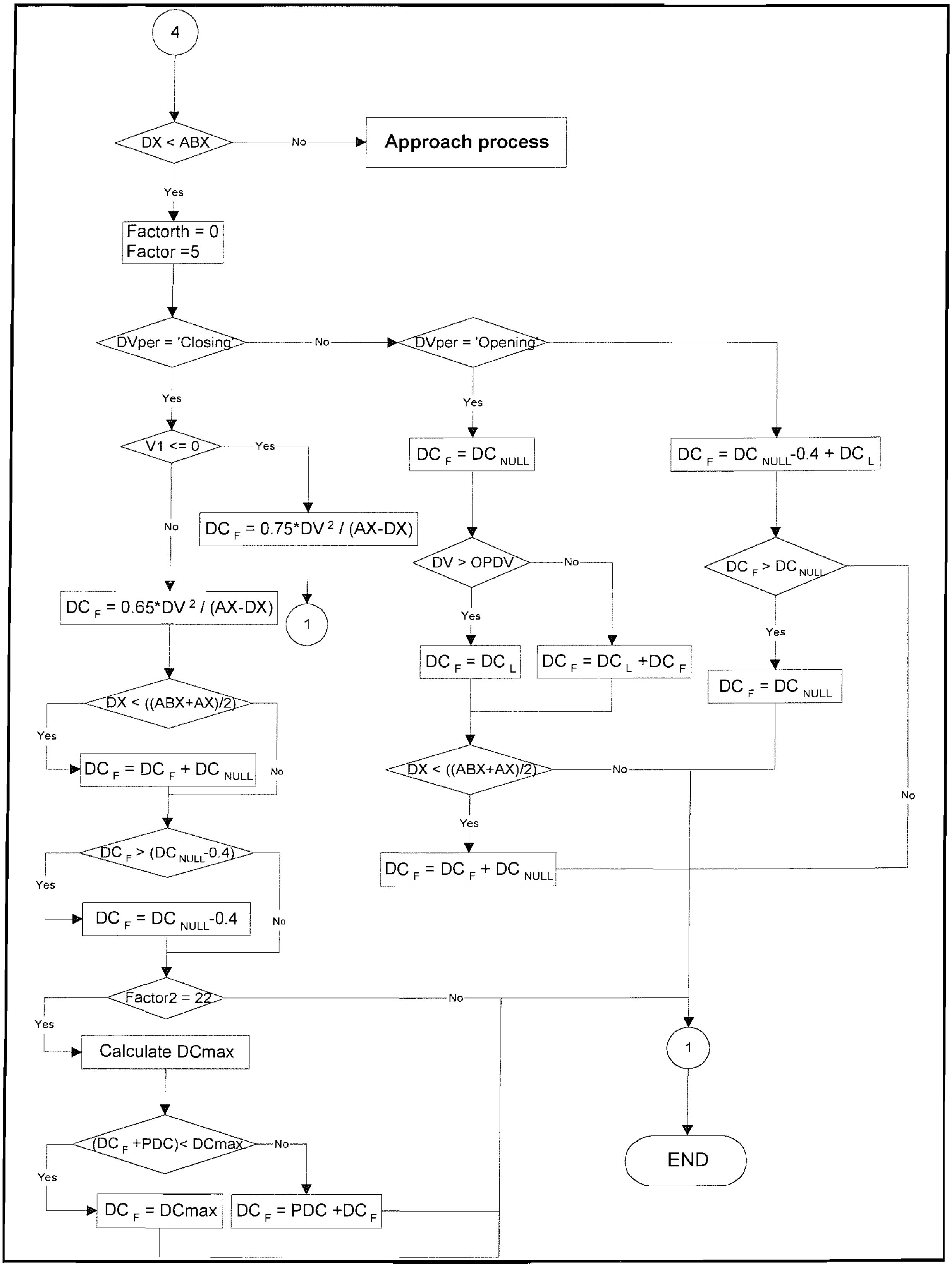


Figure 8.49: The Car following procedure Part (3).

8.7.6 The Test Program:

In order to inspect the operation of the model, a program was written to simulate the movement of a platoon of vehicles (3 to 50vehicles). This program is a key instrument to observe the model microscopically, and determine and characterise any 'odd' behaviour which may occur. In fact, this program was used to determine precisely what are the proper procedures in the model structure. Figures 8.50, 8.51, 8.52 and 8.53 present the time series of the spacing, speed, acceleration and the relative for a platoon of 15 vehicles in which the platoon leader decelerated at a constant rate of -1.5 m/sec^2 to stop. Figures 8.54, 8.55 and 8.56 show the spacing series of the acceleration, relative speed and speed for the same experiment.

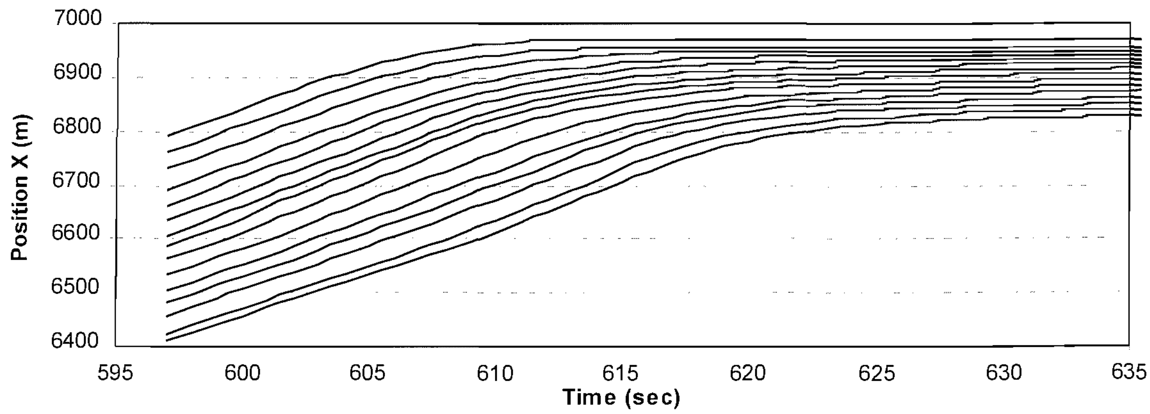


Figure 8.50: the spacing time series for a platoon of 15 vehicles when the lead vehicle decelerates in a constant deceleration -1.5 m/s^2 until stops.

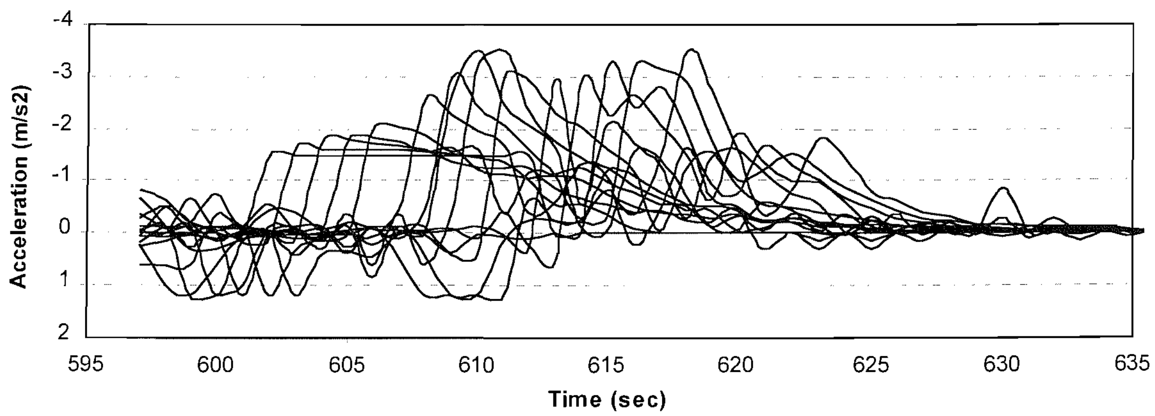


Figure 8.51: the acceleration time series for a platoon of 15 vehicles when the lead vehicle decelerates in a constant deceleration -1.5 m/s^2 until stops.

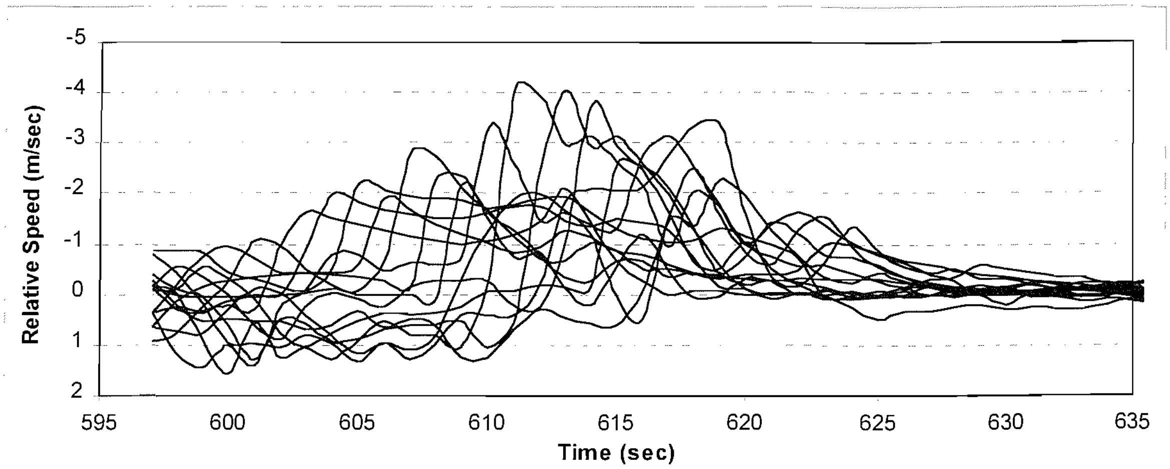


Figure 8.52: the relative speed time series for a platoon of 15 vehicles when the lead vehicle decelerates in a constant deceleration -1.5 m/s^2 until stops.

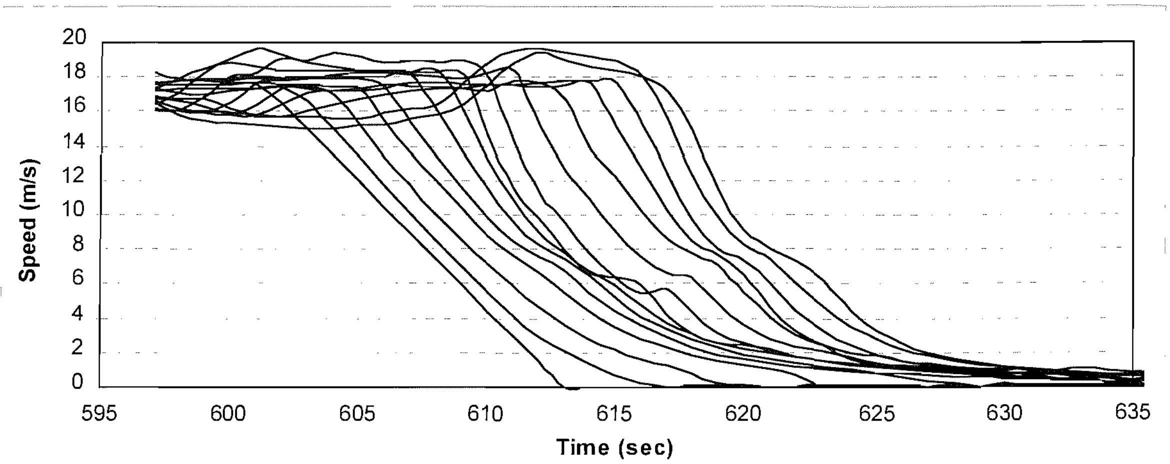


Figure 8.53: the speed time series for a platoon of 15 vehicles when the lead vehicle decelerates in a constant deceleration -1.5 m/s^2 until stops.

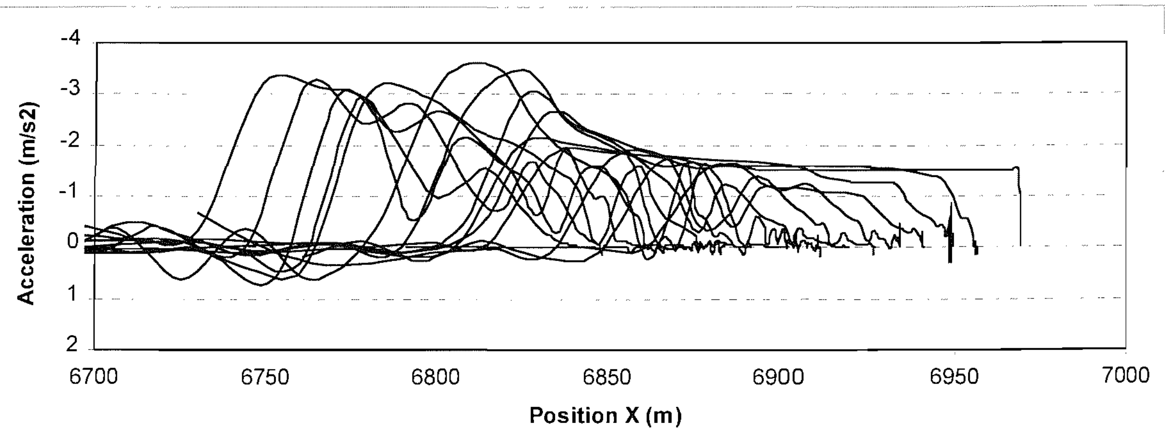


Figure 8.54: the acceleration spacing series for a platoon of 15 vehicles when the lead vehicle decelerates in a constant deceleration -1.5 m/s^2 until stops.

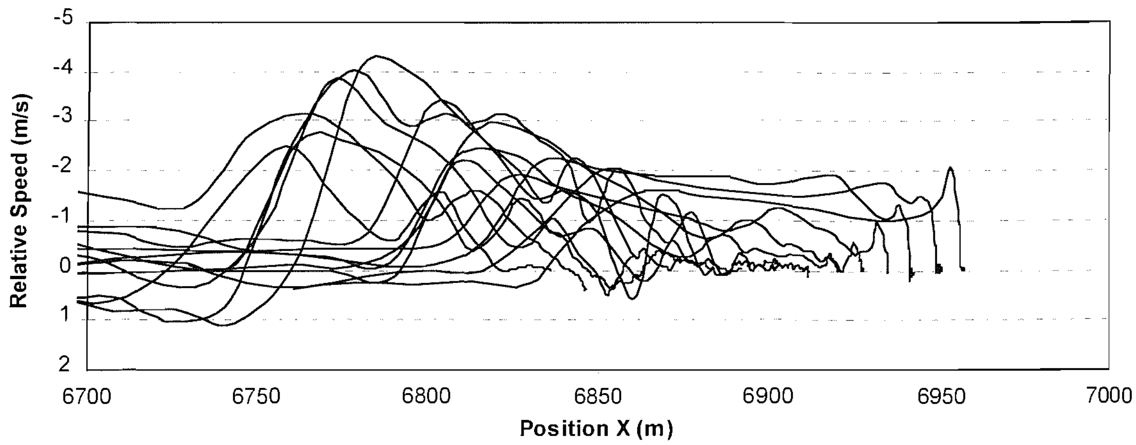


Figure 8.55: the relative speed spacing series for a platoon of 15 vehicles when the lead vehicle decelerates in a constant deceleration -1.5 m/s^2 until stops.

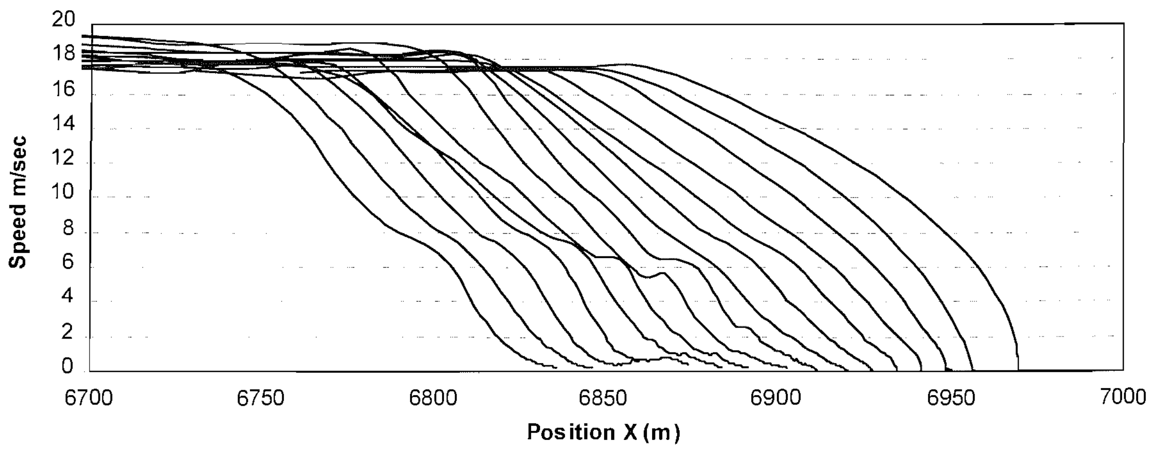


Figure 8.56: the speed spacing series for a platoon of 15 vehicles when the lead vehicle decelerates in a constant deceleration -1.5 m/s^2 until stops.

A further experiment was undertaken to compare the improved model and MISSION model. A platoon of ten vehicles was assumed in which the lead vehicle has a constant speed of 25 m/sec, from which it decelerates at -1.5m/sec^2 to a stop. Two, procedure were used one for MISSION and the other for the improved model. Figures 8.57 and 8.58 present the acceleration time series from the MISSION model and the improved one respectively. Also, Figures 8.59 and 8.60 present the relative speed time series from the MISSION model and the improved one. Clearly the improve model has reduced the variation in the deceleration for the vehicles in the platoon, and maintained an acceptable relative speed variation. This is likely to make it more realistic.

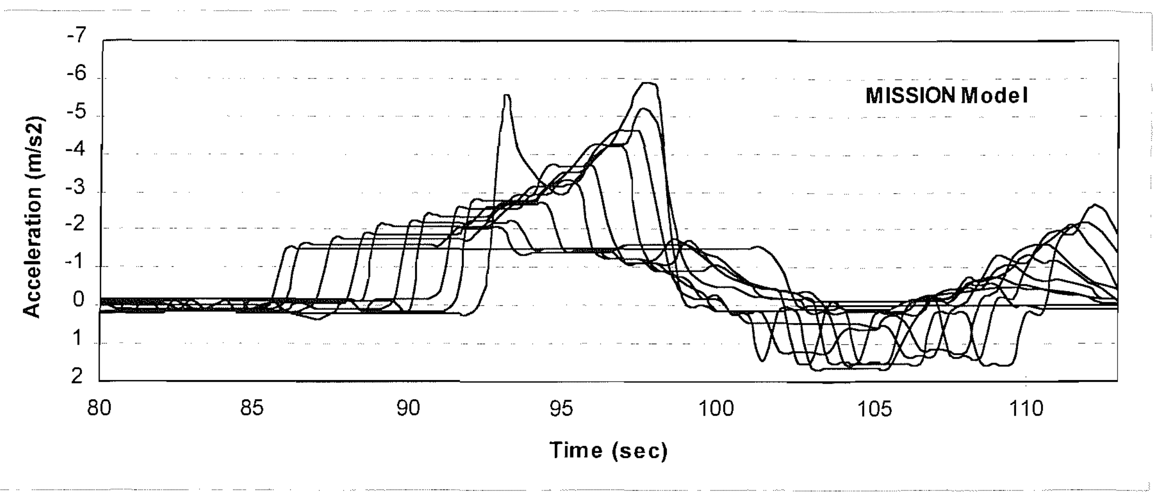


Figure 8.57: the acceleration time series for a platoon of 10 vehicles when the lead vehicle decelerates in a constant deceleration -1.5 m/s^2 until stops. [MISSION model]

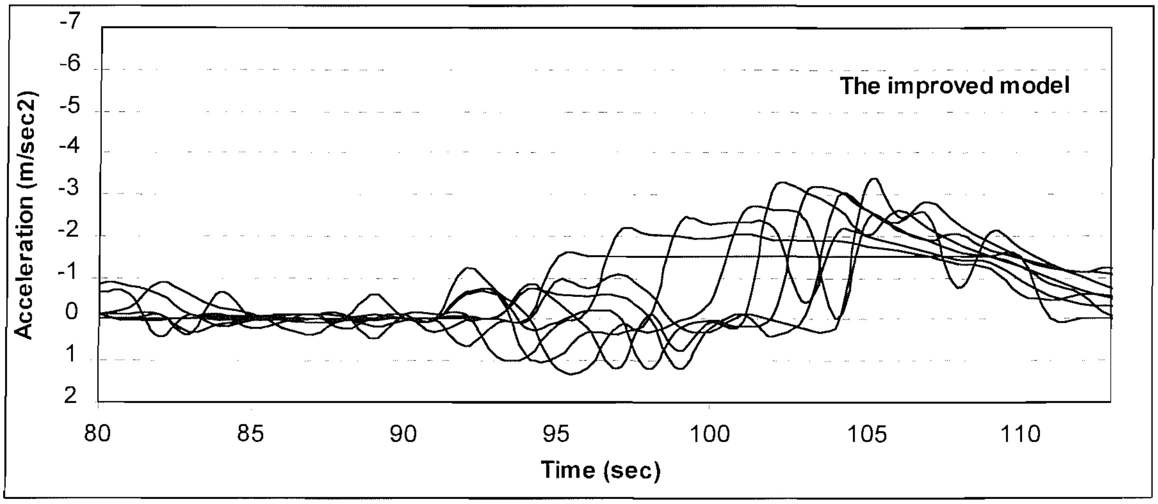


Figure 8.58: the acceleration time-series for a platoon of 10 vehicles when the lead vehicle decelerates in a constant deceleration -1.5 m/s^2 until stops. [the improved model]

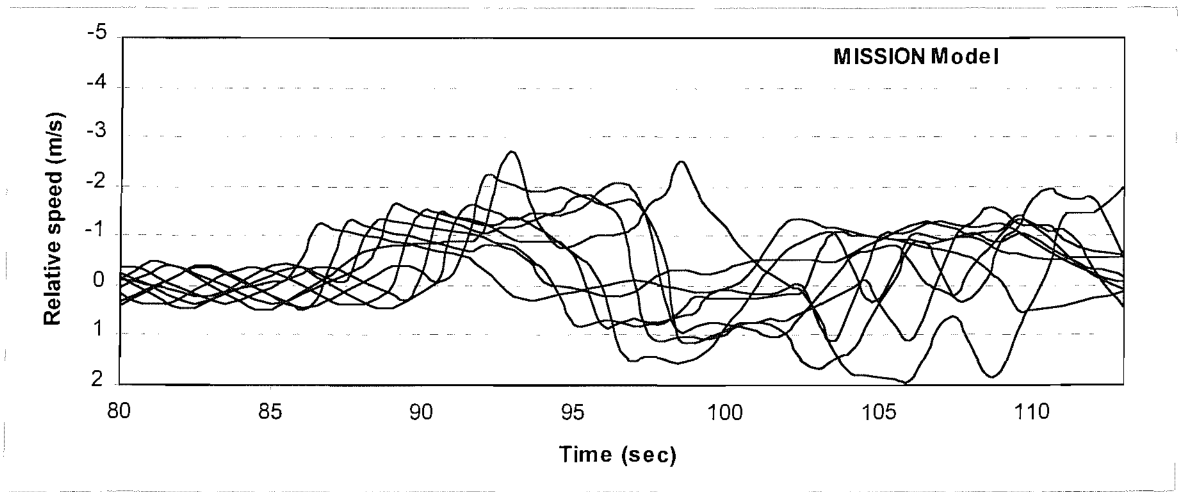


Figure 8.59: the Relative speed time-series for a platoon of 10 vehicles when the lead vehicle decelerates in a constant deceleration -1.5 m/s^2 until stops. [MISSION model]

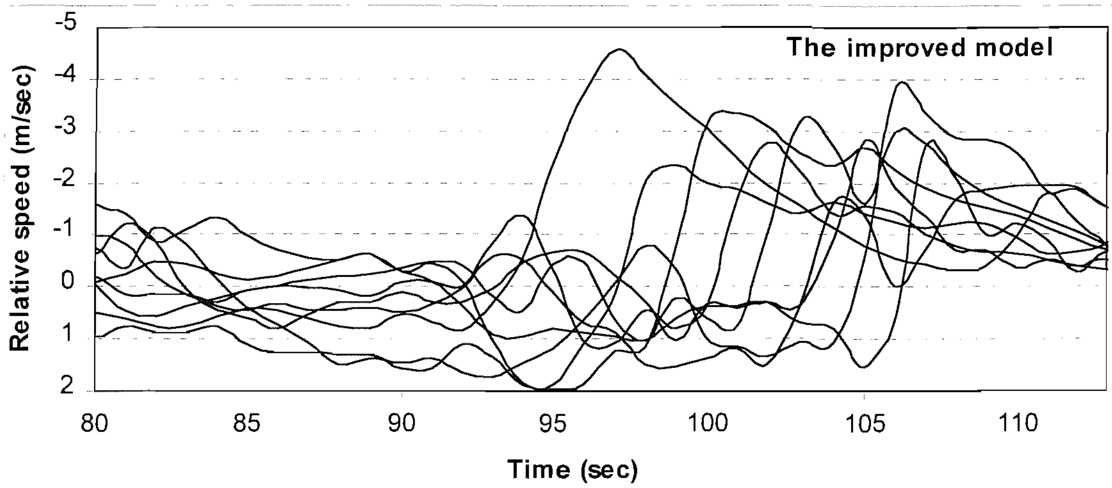


Figure 8.60: the Relative speed time-series for a platoon of 10 vehicles when the lead vehicle decelerates in a constant deceleration -1.5 m/s^2 until stops. [the improved model]

8.8 SUMMARY:

In this chapter, the thresholds in the model were determined and calibrated using microscopic data from the IV. Various experiments were undertaken to examine drivers' behaviour in different circumstances. The model structure was built with the help of a computer test program that observes the model microscopically and uses graphics to show the operation of the model. Nevertheless, a good lane-changing model is important to co-operate with the car following model to allow the possibility to validate the model macroscopically and perform some application. This is the subject of the next chapter.

CHAPTER 9

MODELLING THE LANE CHANGING PROCESS

9.1 INTODUCTION:

Unlike the car following process, lane changing plays an important part in lane utilisation and traffic distribution on a multilane motorway. Drivers usually change lane on motorway either to overtake slower vehicle or to allow other faster vehicles to overtake. It is a very complex process and it is not surprising that most microscopic models have been based purely on a number of well held beliefs about lane changing decisions, with little experimental calibration. (See Chapter 2, Section 2.4). Most of the previous studies collected data to validate their existing models. This is not to suggest that their procedures are necessarily incorrect, but it is more proper to analyse the data to initiate the model assumptions.

Theoretically, the critical time gap was used as a method of approach to model the lane changing behaviour. This was used mainly in priority inter-section studies, because it is easy to measure gaps accepted and rejected by drivers. Consequently, the critical gap would be the point at which the probability of rejection is equal to the acceptance (SALTER and HOUNSELL 1989)[78], (Figure 9.1). However, in the motorway situation it is impossible to measure the rejected gap on what may be a very long section of highway. Also, previous researches had indicated that relative speed has an influence on the driver decisions to change lane (LEUTZBACH W. & WIEDEMANN R. 1986)[21].

This Chapter presents an analysis of microscopic data that was collected in 1993 using video cameras mounted on fixed sights off the (M27) motorway. The results were used to develop a lane-changing model. Several sensitivity tests were carried out to find the parameter values that have major influences on the model performance.

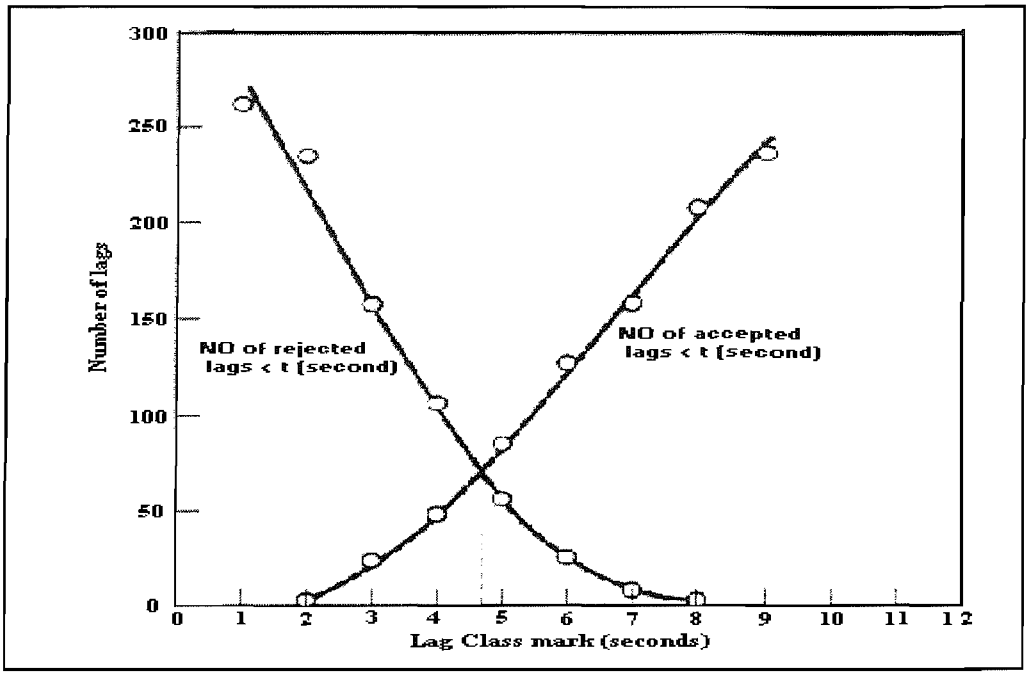


Figure 9.1: Graphical determination of the critical gap (SALTER and HOUNSELL 1989)[78].

9.2 DATA COLLECTION:

In 1993 TRG undertook a major survey to investigate lane usage and driver behaviour on motorways. A series of eight video cameras was set up on a high embankment to monitor vehicle movements as they passed along a stretch of motorway [M27] (McDonald M., Brackstone M. & Jeffery D. 1994)[77]. The spacing between every two cameras was 50m and that enabled a total coverage of 500m of the motorway, enabling vehicles to be tracked, so measuring speed and separation. Figure 9.2 illustrates the experiment's site and the video cameras positions.

One of the survey objectives was to monitor lane-changing manoeuvres and record all the possible related information that may influence drivers' decision before and during the manoeuvre.

The data measurements for every lane-changing event were:

1- Event No. 2- Vehicle speed. 3- Vehicle position. 4- Vehicle type. 5- Old lane. 6- New lane

7- Old front vehicle Speed 8- Old front vehicle Space. 9- Old front vehicle type.

10- Old rear vehicle Speed. 11- Old rear vehicle Space. 12- Old rear vehicle type

13- New front vehicle Speed. 14- New front vehicle Space. 15- New front vehicle type.

16- New rear vehicle Speed. 17- New rear vehicle Space. 18- New rear vehicle type.

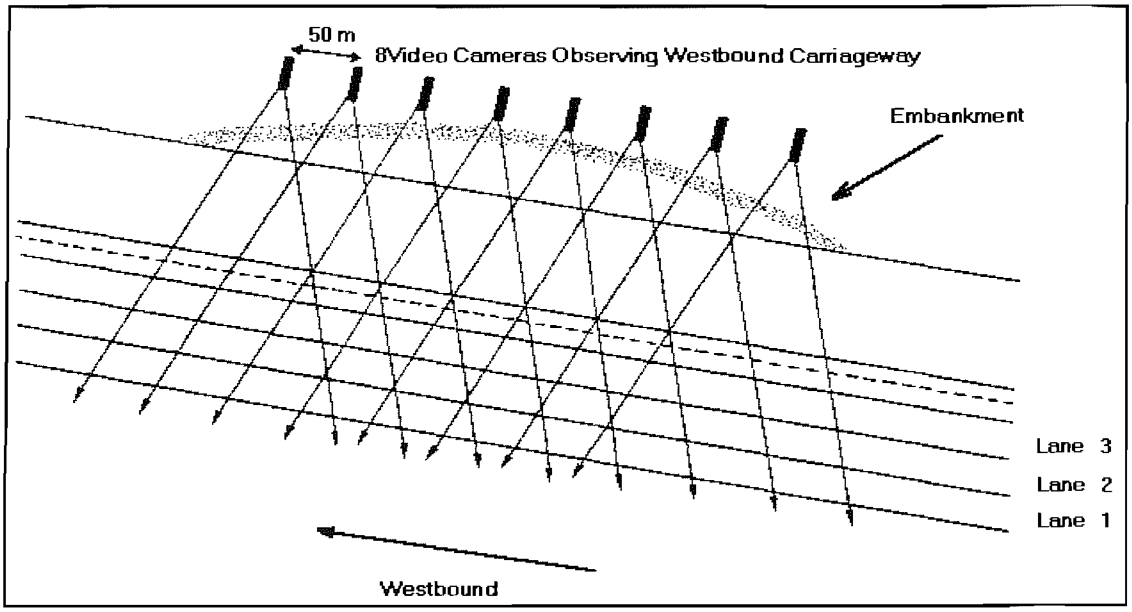


Figure 9.2: The data collection experiment sight and the video cameras positions to cover 500m of the M27 motorway. (McDonald, Brackstone & Jeffery 1994)[77]

Figure 9.3 shows a simple graph for all the data measurements and parameters that were covered by the lane changing survey. These data enabled the researcher to investigate drivers' lane changing decisions and estimate the gap acceptance thresholds under which they are able to perform their manoeuvre. However, the survey site was about 1.5 mile before a major junction on the M27 Motorway, which may have biased the data of lane changing to the left events.

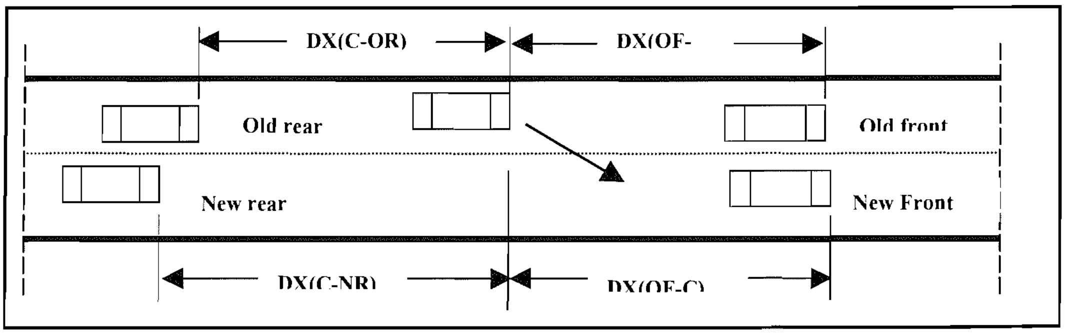


Figure 9.3: the data record of LC manoeuvre to the right (faster lane) by the survey.

9.3 DATA ANALYSIS:

The data was received in Spread-Sheet format (MS.Excel), in which every row presented a lane changing event with all related measures described in the previous paragraph (Section 9.2). Therefore, the data analysis task was to sort the data, distinguish between different types of events then consider the relative speeds and distances related to each lane changing situation. Two main types of lane changing were distinguished:

1- Lane changing to the right (to the faster lane).

2- Lane changing to the left (to the slower lane).

Further situations were specified for each of the mentioned type this will help to develop the lane changing logic in the model.

9.3.1 Lane Changing Manoeuvres To The Right [Faster Lane]:

Drivers change lane to the right (faster lane) in order to overtake slower vehicle in front, or to allow another vehicle to merge from a slip road, or to avoid perceived future delay. The analyses performed as part of this research will focus on the first situation, because the objective is to study drivers behaviour on motorways not influenced by any other type of traffic (eg merging traffic). Also, the available data did not include merging behaviour. (Future work could follow to model the merging process and complete the lane-changing model for all situations).

A driver decision to change lane to the right may be considered to be influenced by three vehicles, they are:

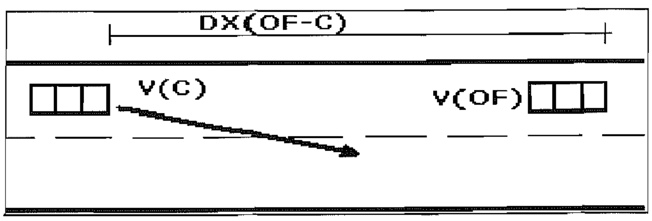
1-) The lead vehicle in the old lane [Veh_{OF}].

2-) The lead vehicle in the new lane [Veh_{NF}].

3-) The rear vehicle in the new lane [Veh_{NR}].

Although, in the real world, drivers will probably assess the situation in more holistic way, the process must be described in a sufficiently simple way so as to form a lane-changing model.

9.3.1.1 The influence of the lead vehicle in the current lane [Veh_{OF}]:



The vehicle in the old lane [Veh_{OF}] plays an essential part in detecting the desire to change lane to the right. In order to assess the influence of [Veh_{OF}], the data was filtered to consider only the situation where there were no vehicles registered in the faster lane. Thus the only two parameters, which could be contributed to influence the drivers' decision, are: the relative speed $DV_{(OF-C)}$ and the spacing distance $DX_{(OF-C)}$ with the lead vehicle. Obviously, the speed of the lead vehicle has to be less than the driver's desired speed. Figure 9.4 presents the relationship between $DV_{(OF-C)}$ and $DX_{(OF-C)}$ for this situation. It is possible to suggest a quadratic equation between $DV_{(OF-C)}$ and $DX_{(OF-C)}$ that could simulate limits for this situation. The equation was found to be :

$$DV_{(OF-C)} = -0.0019 * DX_{(OF-C)}^2 + 0.0817 * DX_{(OF-C)} - 2.5318 \qquad \ldots\ldots 9.1$$

This threshold supports the idea that drivers change lane farther behind the lead vehicle, when the relative-speed with the lead vehicle higher. (Figure 9.4)

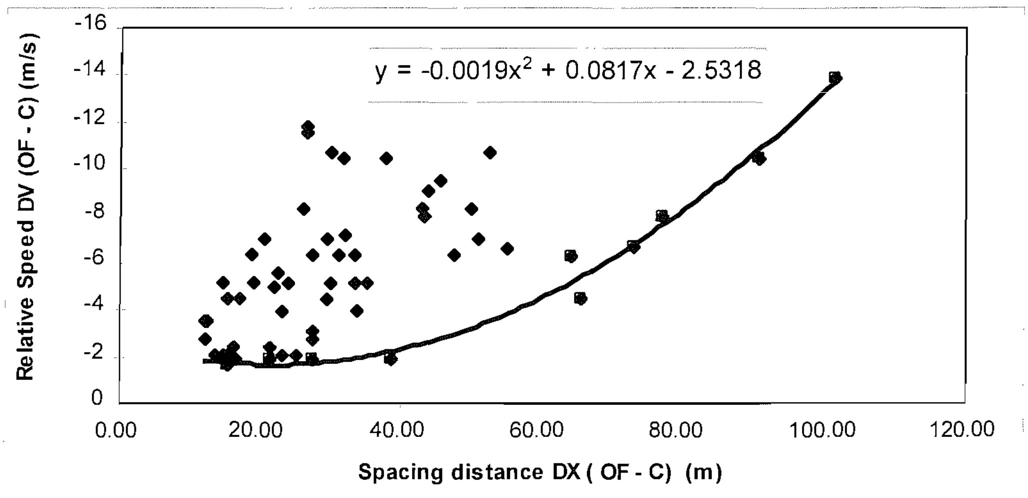


Figure 9.4: The relative speed against relative distance with respect to the lead vehicle in LC manoeuvre to the Right.

However, by examining the time to collision [TTC = DX/DV], the data showed that drivers tended to achieve short TTC before they change lane to the right when there is no traffic in the faster lane. The data shows the TTC values has a Lognormal distribution with the parameters: Mean = 6.599 sec, Std = 3.504 sec, K-S = 0.876. Figure 9.5 shows the frequency histogram with the fitted Lognormal distribution for the TTC values in this situation. Introducing such phenomena inside the simulation model is very difficult, because it generates a very dangerous situation that can not be handled by the present car following model, unless it is changed fundamentally. Nevertheless, whether drivers do lane change before achieving such TTC levels or not, the journey time probably will not be affected. Therefore, it was concluded that the equation 9.1 would be sufficient to present the points at which drivers change lanes to the right when the outside lane is empty.

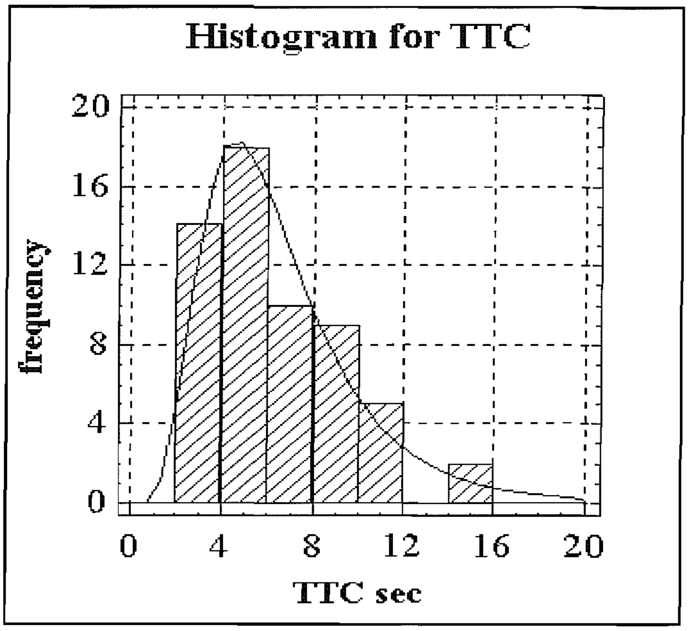
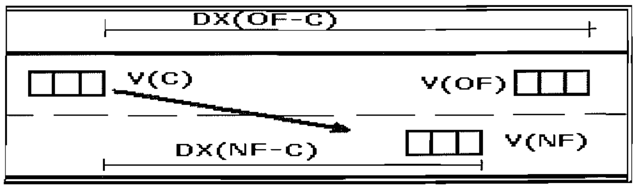


Figure 9.5: The frequency histogram with the lognormal distribution fit for the TTC values when drivers change lane to the right and the outside lane is empty.



A further situation was considered in which there is a new front vehicle in the faster lane. This situation is totally different from the previous one, and there are many possible interpretations of driver behaviour. The data was filtered to present this situation, and Figure 9.6 illustrates that it is very difficult to draw precise conclusions about drivers' decisions in such complex circumstances.

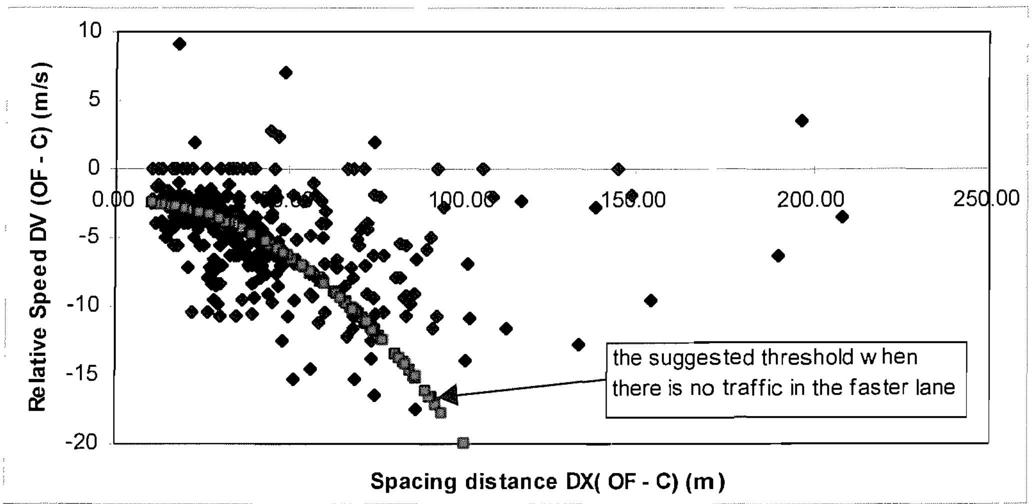


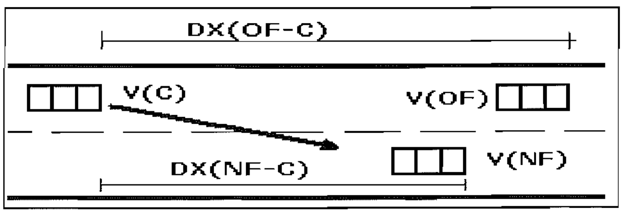
Figure 9.6: The relative speed against the relative distance with respect to the lead vehicle [Veh.OF] and the existence of new lead vehicle [Veh.NF] for the LC events to the right.

As can be seen from Figure 9.6, there are some points where the relative speed of the lead vehicle is positive. This can be explained if the lead vehicle is also performing a lane changing manoeuvre to the faster lane. Also, there are many points under the suggested threshold, Equation 9.1. This probably indicates that such drivers were in the following processes immediately before lane changes. Finally, some points have a very large relative distance between the driver and the lead vehicle. This probably means that the driver is predicting from the surrounding traffic situation that there is an opportunity to perform a lane change which may not be available in the foreseen future.

In conclusion, this data is insufficient to fully describe all the subtle influences as when a driver will want to change lane to the right. Common sense will guide us to the perception threshold (Chapter 7, Section 7.3.1), above which a driver will perceive whether he or she is closing on a slower vehicle in front or not. Therefore, the model will use equation 9.1 and the perception threshold to detect the need for a lane changing. (ie if a driver's desired speed is above the threshold (equation 9.1) then he or she will detect the necessity to change lane, otherwise, the perception threshold will be used). This will be explained later in the lane changing detection to the right.

9.3.1.2 The influence of the lead vehicle in the new lane [Veh.$_{NF}$]:

To analyse the influence of the new lead vehicle [Veh.$_{NF}$] in the faster lane on the driver decision to change lane to the right, two restrictions are suggested:



1- The spacing distance with the [Veh.$_{NF}$] has to be less than the spacing distance with the [Veh.$_{OF}$], otherwise the driver may be able to complete his or her manoeuvre before catching the [Veh.$_{NF}$].

2- The [Veh.$_{NF}$] has to be faster than the old lead vehicle [Veh.$_{OF}$]. Otherwise the driver will not change lane unless the [Veh.$_{NF}$] is far away from [Veh.$_{OF}$] (more than 200m).

Regarding the previous restrains, two main situations have been considered to evaluate the influence of the [Veh.$_{NF}$], they are:

The driver's speed is more than the speed of the [Veh.$_{NF}$]to the right:

The data was filtered and prepared in order to calibrate this situation. The speed and position of the lead vehicle in the right lane have to meet the measurement restraints.; So, there has to be no rear vehicle in the right lane. After that, the relative speed and distance relationship with the new lead vehicle [Veh.$_{NF}$] was examined. Figure 9.7 presents this relationship, where a possible threshold can be proposed with a quadratic relation between $DV_{(NF-C)}$ and $DX_{(NF-C)}$ [equation (9.2)] under which drivers are able to change lane .

$$DV_{(NF-C)} = -0.001 * DX^2_{(NF-C)} + 0.0015 * DX_{(NF-C)} - 2.1317 \quad \text{....9.2}$$

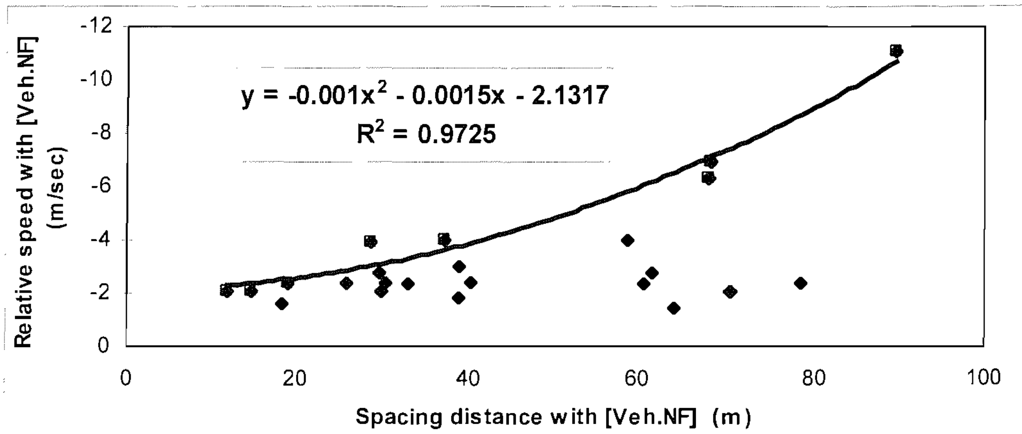


Figure 9.7: The relative speed against relative distance with respect to the [Veh.$_{NF}$] for the LC manoeuvre to the right (with the absence of the new rear vehicle [Veh.$_{NR}$]).

This threshold was examined again in the situation in which the new rear vehicle was not far away. The data shows a minor change and the proposed threshold has the same characteristic (quadratic relationship) with a very insignificant change in the equation's parameters:

$$DV_{(NF-C)} = -0.0011 * DX^2_{(NF-C)} + 0.0105 * DX_{(NF-C)} - 2.6741 \quad \text{...9.3}$$

Figure 9.8 presents the data and the modified threshold when considering the new rear vehicle in the right lane. The equation (9.3) will be used to describe this situation in the model. However, the spacing distance with the [Veh.$_{NF}$] was however checked against the speed of the drivers. The data shows no relation between the two parameters, (Figure 9.9). This leads to the conclusion that whenever a driver is under the DV threshold (equation 9.3) and has a safe spacing with the [Veh.$_{NF}$] (determined by the distribution of ABX) he or she is able to change lane with respect to the [Veh.$_{NF}$].

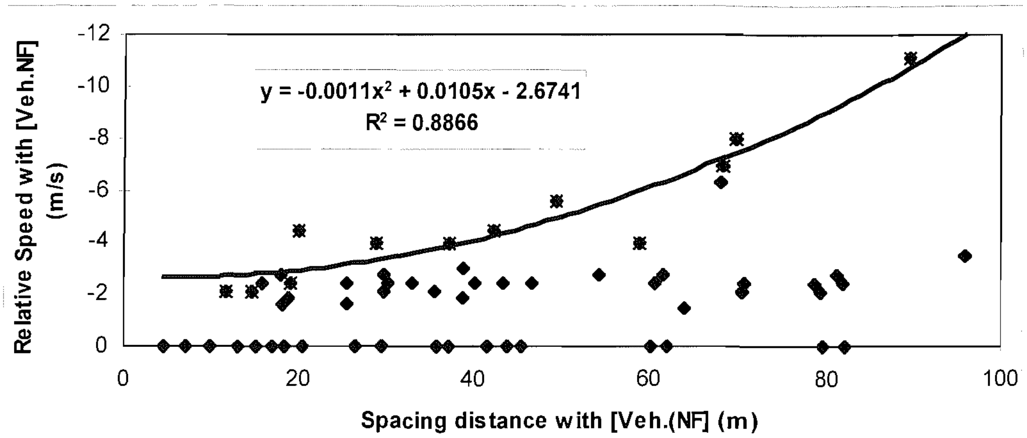


Figure 9.8: The relative speed against relative distance with respect to the [Veh.$_{NF}$] for the LC manoeuvre to the right (with the new rear vehicle [Veh.$_{NR}$] present).

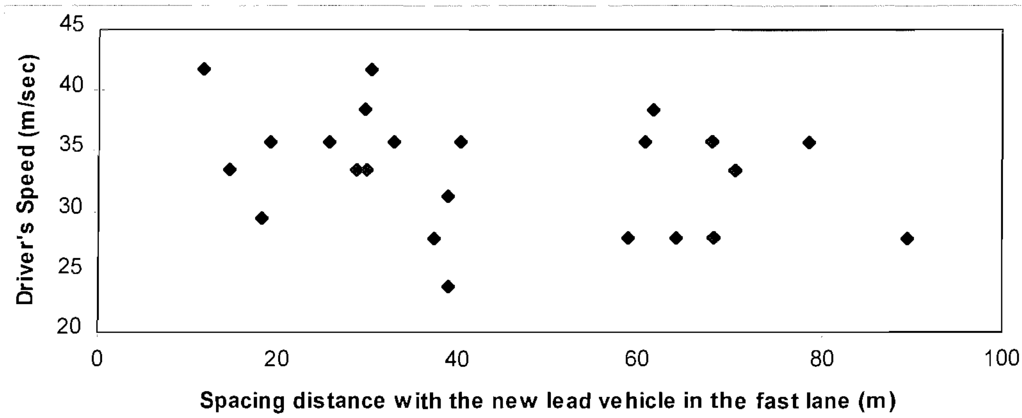


Figure 9.9: The relative distance to the new lead vehicle against the driver's speed.

The driver's speed is less than the speed of the [Veh.$_{NF}$] to the right:

In these circumstances it is expected that drivers will not consider the new lead vehicle in the lane changing decision as long as it is faster than their current speed. The only restriction is the availability of safe spacing distance for the following process. Figure 9.10 shows the result for the relative speed and relative distance relation between the driver and the [Veh.$_{NF}$] vehicle. The data was very scattered which indicates that the relative speed is independent from the relative distance. Also, the drivers' speed with respect to relative distance was examined. No correlation or relationship was found between the two mentioned parameters. (Figure 9.11).

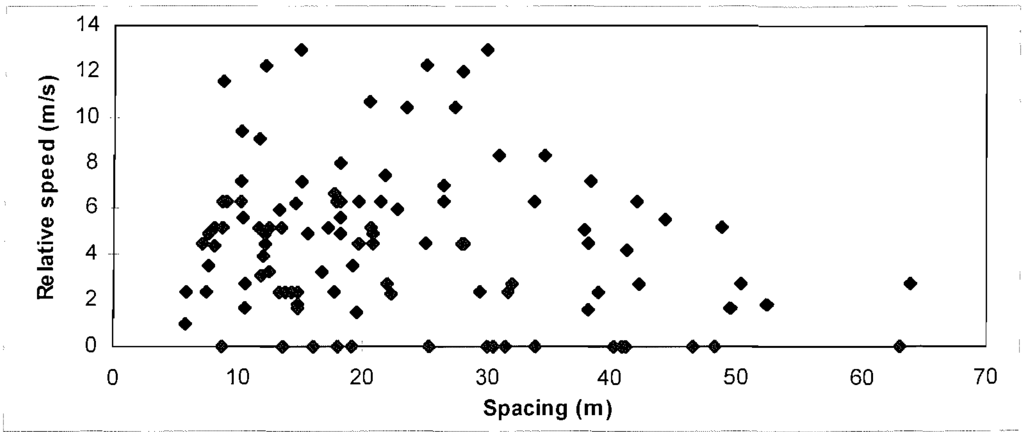


Figure 9.10: The relative speed against the spacing distance between the driver and the [Veh.NF], (the [Veh.NF] is faster)

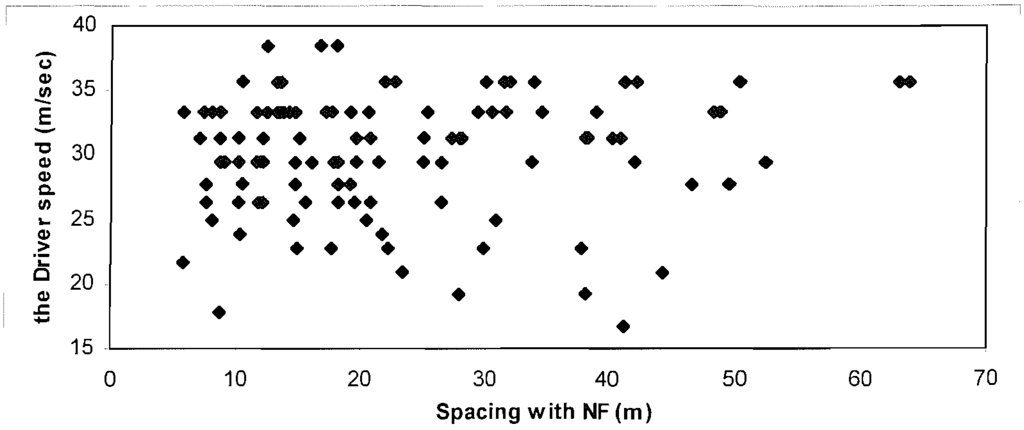
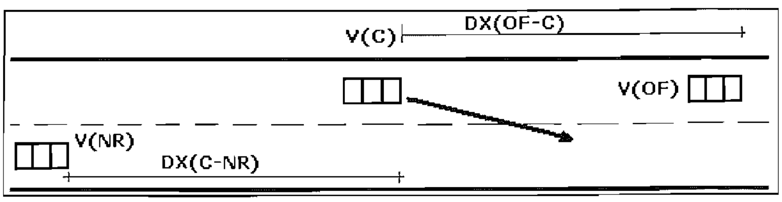


Figure 9.11: The driver's speed against the spacing to the [Veh.NF] vehicle, (the [Veh.NF] is faster)

9.3.1.3 The influence of the rear vehicle in the new lane [Veh.NR]:



The same procedure was followed to assess how the rear vehicle in the new right lane [Veh.NR] may influences drivers' decision to change lane to the right. The [Veh.NR] has to be faster than the driver's speed and close enough to affect his decision [DX(C-NR) < 200m). The 200m, which is at least equal to four seconds headway, was considered to be the maximum distance for perception (WIEDEMANN 1986)[21]]. Also, . the [Veh.NF] has to be far away and out of the survey range distance. However, because the survey was undertaken on a three lane motorway, it may be argued that the decision to change from lane one to lane two could be different from the decision to change lane from lane two to lane three. This is because if the current lane is lane one, the driver may consider that the [Veh.NR] will be able to change lane from two to three to avoid him or her. In order to avoid such a conflict in the analysis, The manoeuvres from lane two to lane three were considered first. Figure 9.12 presents the relative speed and relative distance data points with the new rear vehicle in the right lane [$Veh._{NR}$] (LC from lane two to three). Another proposed threshold was considered to determine the level of relative speed and distance under which a driver is able to lane change to the right with the present of the [$Veh._{NR}$]. The equation was found to be:

$$DV_{(C\text{-}NR)} = -2.4078 * Ln(DX_{(C\text{-}NR)}) + 3.8711 \qquad \ldots\ldots 9.4$$

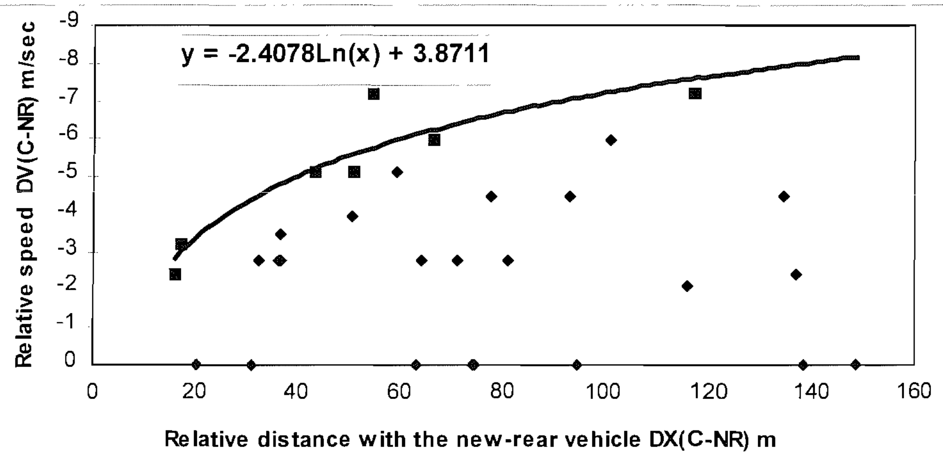


Figure 9.12: The relative speed against the spacing distance to the new-rear vehicle in the right lane (LC from lane Two to Three).

The second step was to look at the data with considering three lanes (LC from One to Two) and (LC from Two to Three). As expected, drivers appear to take more risk with the [$Veh._{NR}$] vehicle when changing lane from One to Two. However, only four points were over the previously suggested threshold (equation 9.4), (Figure 9.13). Thus, it can be argued that Equation 9.4 could be used to characterise drivers' behaviour with respect to the rear new vehicle [$Veh._{NR}$] when they change lane to the right. Nevertheless, some sort of logic inside the model has to be developed to consider the situation where drivers anticipate that the [$Veh._{NR}$] is able to move to a farther lane safely.

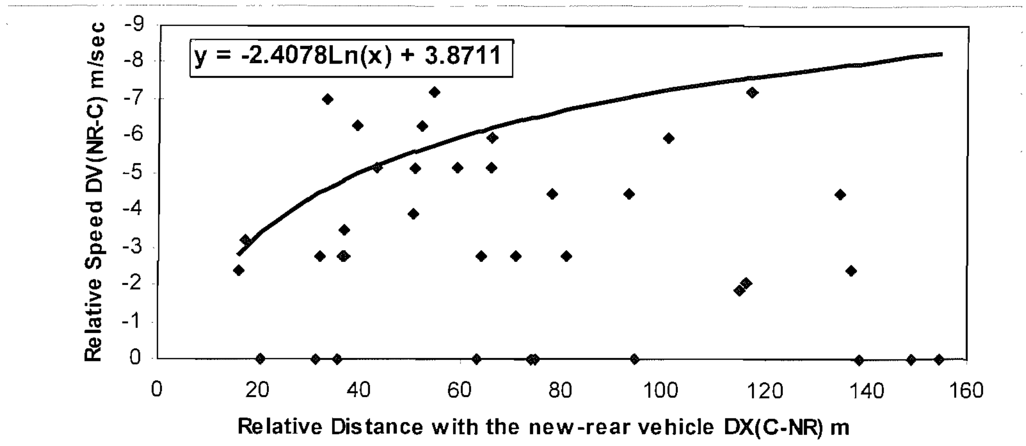
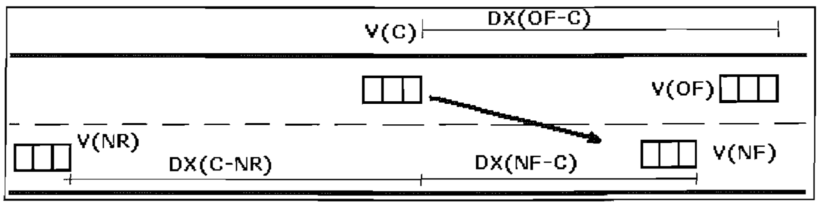


Figure 9.13: The relative speed against the spacing distance to the new-rear vehicle in the right lane [(LC from lane One to Two)&(LC from Two to Three)].

Finally, the data was analysed again with respect to the condition with a new front vehicle in the right lane. This situation describes the circumstances where the level of service is low, and the opportunity to change lane is less and drivers become more aggressive. The same procedure was followed, with the lane changing events from lane two to three examined first. The data showed that drivers appear to become more aggressive and are willing to take more risk in their manoeuvres. Another threshold was proposed for the acceptable relative speed with respect to the spacing distance range (Figure 9.13). The threshold equation was found to be:



$$DV_{(C\text{-}NR)} = -4.0034 * Ln(DX_{(C\text{-}NR)}) + 8.4239 \qquad ....9.5$$

However, when the lane changing events from lane One to Two were considered the difference was more significant, and the drivers' apparent risk taking far greater. This may be because drivers anticipate when the new rear vehicle is able to change lane from lane Two to Three. Figure 9.14 shows the relative speed against relative distance with the new rear vehicle for the Lane changing events from lane One to Two with the present of the new front vehicle. The new proposed threshold has become:

$$DV_{(C\text{-}NR)} = -5.6037 * Ln(DX_{(C\text{-}NR)}) + 12.051 \qquad ....9.6$$

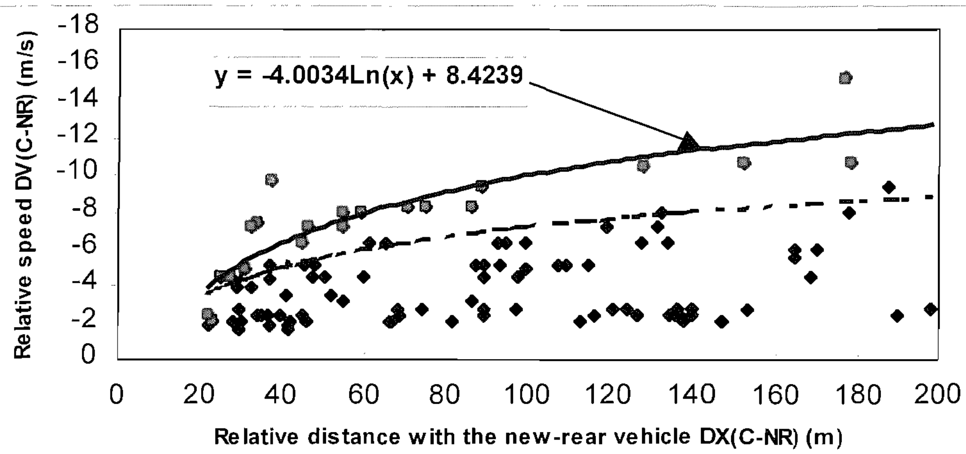


Figure 9.14: The relative speed against Relative distance to the new-rear vehicle (the new-front vehicle is not absent) [LC from lane Two to Three]

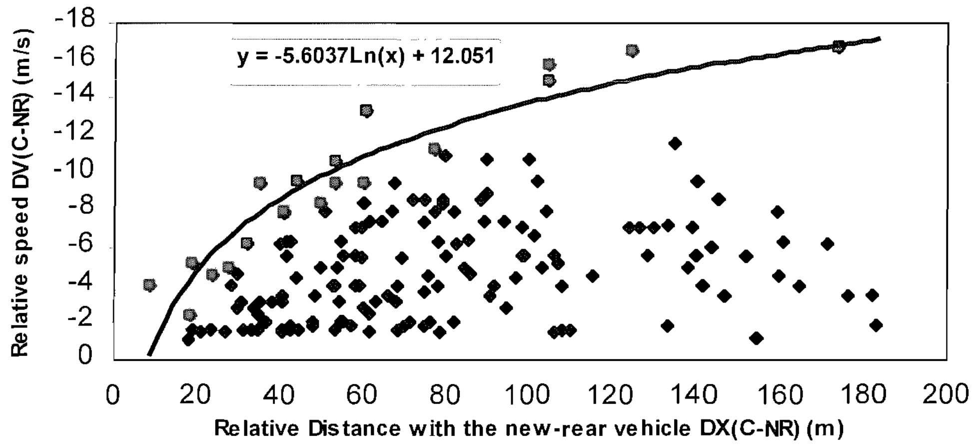


Figure 9.15: The relative speed against Relative distance to the new rear vehicle (the New Front vehicle is not absent) [LC from lane One to Two]

The difference between the original proposed threshold (Equation 9.4) and the later thresholds (Equations 9.5 and 9.6) can be used inside the model logic to replicate drivers aggressiveness which builds up when they are not able to perform their desired manoeuvre.

The above thresholds are the maximum measured relative speeds achieved by some observed drivers. It is difficult to believe that all drivers will commit themselves to such limits during their manoeuvres. For modelling purposes, a random number with a normal distribution N(mean=70% and Std=30%) was introduced to simulate the differences between drivers' lane changing thresholds.

9.3.2 Lane Changing Manoeuvres To The Left [Slower lane]:

Unlike lane changing to the right, drivers change lane to the left (slower lane) in order to allow faster vehicle to pass. (Although, drivers are instructed in the Highways Code to return to the left hand lane after they finished overtaking, not everyone follows the rule (YOUSIF S. 1995)[30]). Finally, drivers may change lane to the left to prepare for an exit manoeuvre. The driver decision to change lane to the left is considered to be influenced by three vehicles, they are:

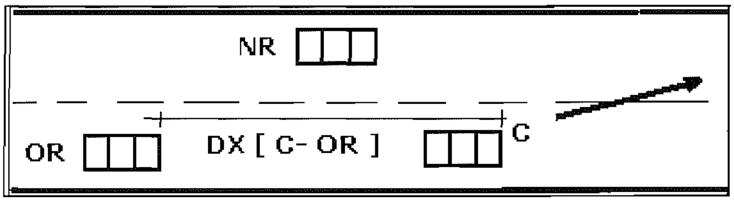
1-) The rear vehicle in the current lane [Veh.$_{OR}$].

2-) The lead vehicle in the new lane [Veh.$_{NF}$].

3-) The rear vehicle in the new lane [Veh.$_{NR}$].

Whilst it is likely that drivers make a complex integrated decision on all the information available the data analysis has had to be simplified because the model is deterministic.

9.3.2.1 The influence of the rear vehicle in the current lane [Veh.$_{OR}$]:



This vehicle encourages the driver to change lane to the left and give the way. However, it may be argued that a driver may not give the way if he or she anticipates that the [Veh.$_{OR}$] is able to move to an outside lane. Consequently, in the analysis, only the lane changing events from lane three to lane two were considered. Figure 9.16 shows the data of the relative speed against the relative distance with the [Veh.$_{OR}$] for lane changing events to the left. Clearly, there are many points where the relative speed has a positive value. This indicates that drivers were faster than the rear vehicle on the old lane and their desire to change lane was to return to the left hand lane as instructed in the Highways Code.

The data was filtered again to assess only those situations where the relative speed was less than or equal to zero. However, the result did not change and no obvious threshold was evident (Figure 9.17).

In order to complete the analysis, the headway distribution of the rear following vehicle [Veh.$_{OR}$] was studied. The lognormal distribution gave [mean = 1.29 sec and Std = 0.867 and K-S_{test} =0.14> 0.05], (Figure 9.18)

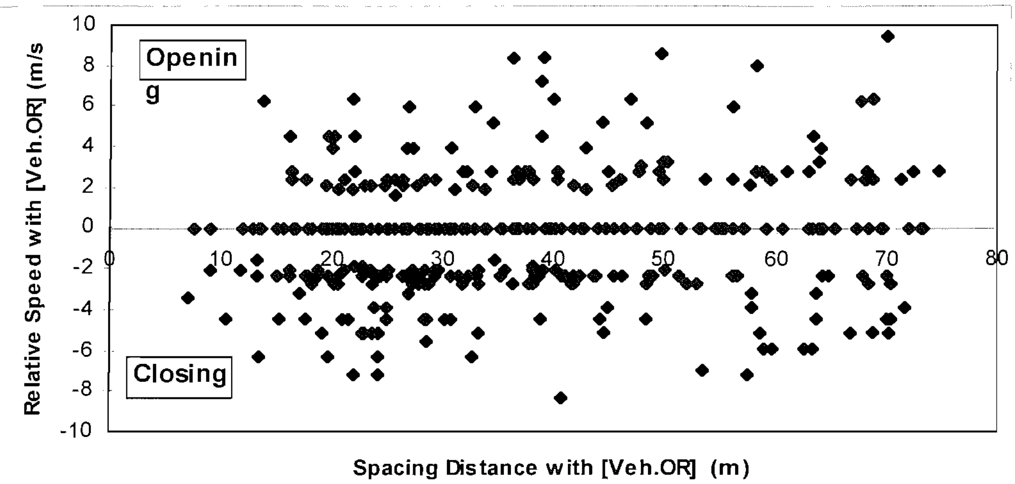


Figure 9.16: the [$Veh._{OR}$] relative speed against relative distance for the lane changing events to the left.

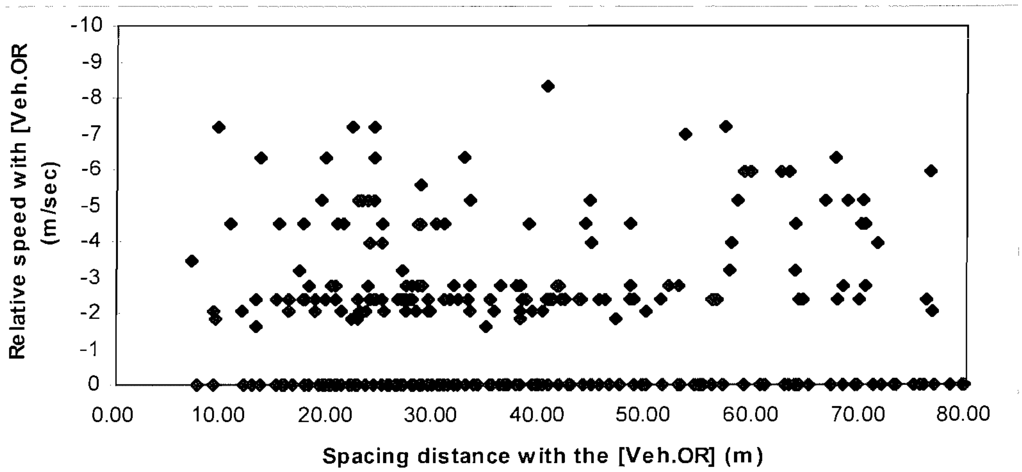


Figure 9.17: the [$Veh._{OR}$] relative speed against relative distance for the lane changing events to the left (the [Veh.OR] is faster than driver speed)

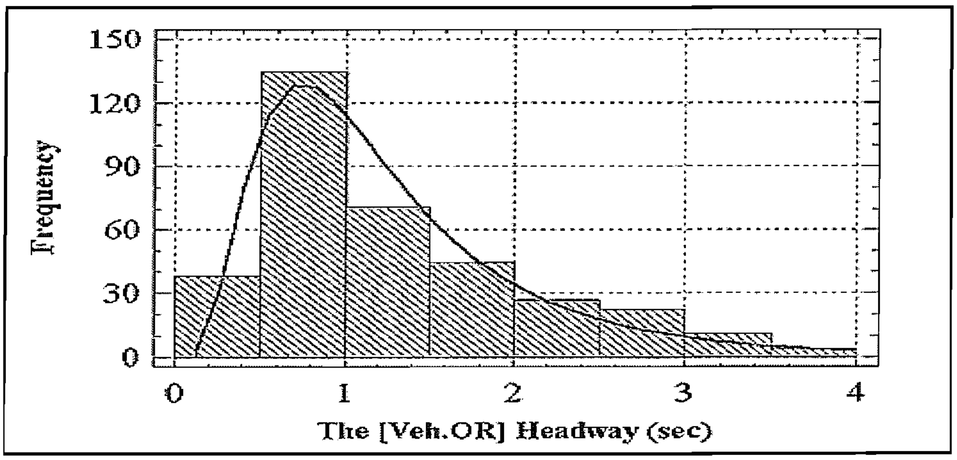


Figure 9.18: The frequency histogram for the [$Veh._{OR}$] Headway for lane changing to the left.

Although, the headway distribution of the rear following vehicle [Veh.$_{OR}$] has a specific distribution, this does not help in the modelling process, where such a distribution should be an output from the model rather than an input. In order to solve the problem of determining the starting point of the rear following vehicle's influence, the SDX_{max} equation 8.13 was used [$SDX_{max} = 2.0687*V_{OR}^{0.6944}$]. Though, the driver stimulus to change lane to the left will rise with a linear proportion to reach its maximum at the point where the rear vehicle following distance equal to the ABX_{min} equation 8.10 [$ABX_{min} = 3.2485*V_{OR}^{0.2118}$]. Whilst this theoretical assumption was proposed without supporting evidence, the final output from the model will show whether or not it is valid. Figure 9.19 presents the relative distance with respect to the speed to determine the minimum and maximum influence of the rear following vehicle.

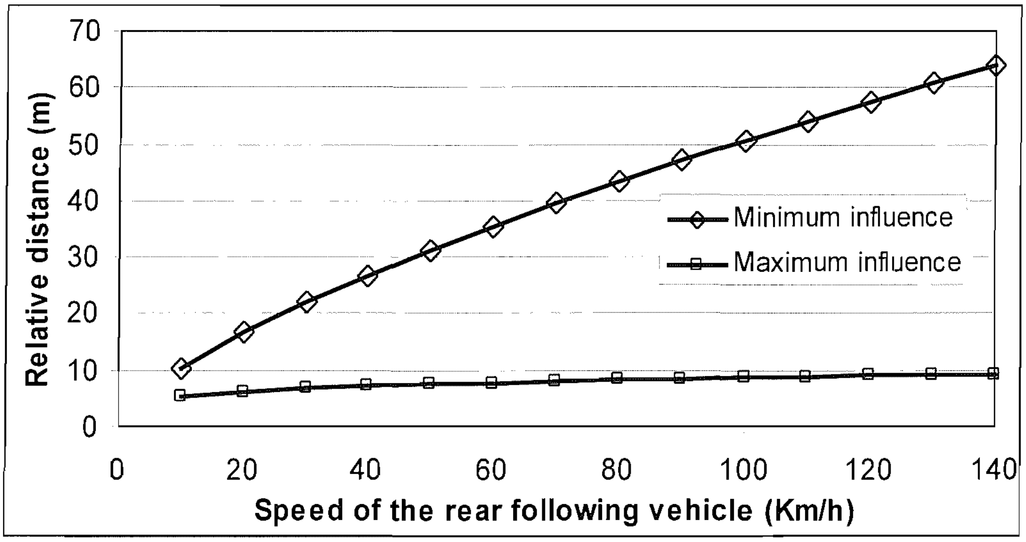
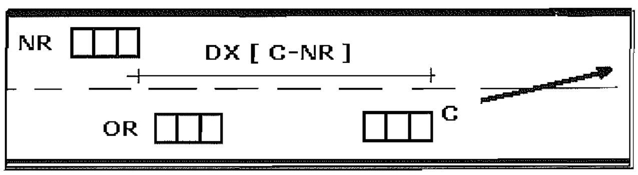


Figure 9.19: The relative distance to the [Veh.$_{OR}$] with respect to the speed to determine the minimum and maximum influence on LC to the left.

9.3.2.2 The influence of the rear vehicle in the new lane [Veh.$_{NR}$]:

The present of the rear vehicle in the new lane is expected not to affect the driver decision to change lane to the left unless the rear vehicle speed is faster than the driver's speed. However, drivers need to have a safe margin of headway before they perform the manoeuvre. In order to investigate the influence of the [Veh.NR] two situations were distinguished:



1- The driver's speed is less than the speed of the [Veh.$_{NR}$] to the left:

It is not common on motorways to have the speed of the rear vehicle in the left hand lane to be faster, unless the traffic flow is very high. Though, the data survey included such lane changing events, the sample was small. Figure 9.20 presents the lane changing events that describe this situation. Another threshold was also suggested to limit the relative speed levels with respect to the spacing distance. As can be seen from Figure 9.20, three events were seemed irrelevant and were discounted, these can be explained either by assuming the drivers have an emergency stopping or there where an error in measuring the speed. The threshold equation was found to be:

$$DV_{(C\text{-}NR)} = -1.8531 * Ln(DX_{(C\text{-}NR)}) + 3.4245 \qquad ......9.7$$

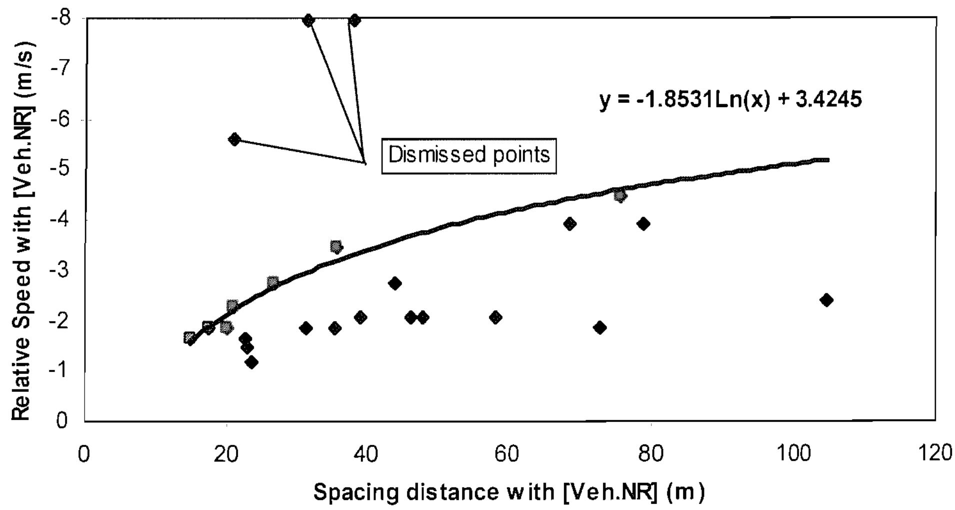


Figure 9.20: The relative speed Vs the relative distance with respect to the new-rear vehicle [Veh.$_{NR}$] (Lane changing to the left) (the [Veh.$_{NR}$] is faster than the driver)

2- The driver's speed is greater than the speed of the [Veh.$_{NR}$] to the left:

This is likely to be the most common situation, and it may be argued that the rear vehicle in the left lane [Veh.$_{NR}$] does not influence the drivers' decision to change lane to the left as long as they are faster than the [Veh.$_{NR}$]. The data has supported this argument for the relative speed against relative distance relationship, and no obvious threshold can be determined (Figure 9.21). Nevertheless, drivers have to achieve a certain headway before they can undertake their manoeuvre. In order to examine the headway distribution two situations were considered:

1- The rear vehicle in the old lane is present and the drivers are under pressure to move to the left (Following headway is less than 1.5 sec).

2- The rear vehicle in the old lane is absent and the drivers are not under pressure to move to the left

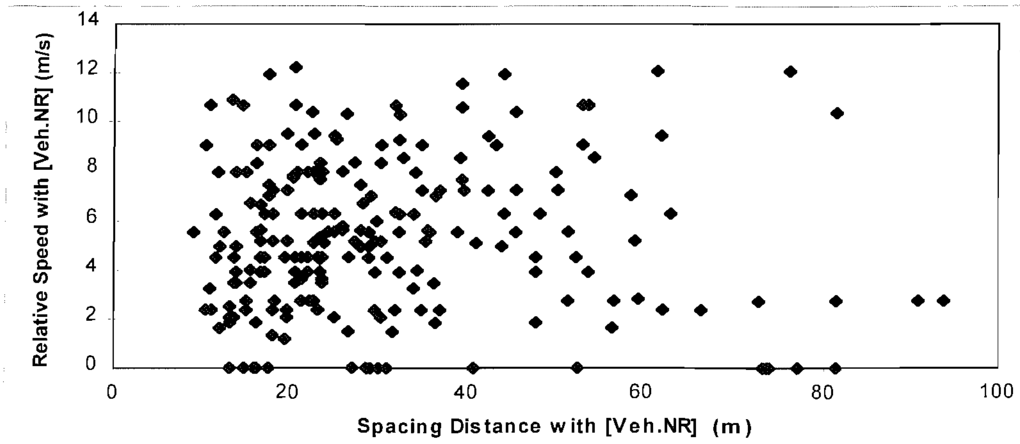


Figure 9.21: the relative speed against relative distance to the [$Veh._{NR}$] (Lane changing to the left and the driver speed is more than the [$Veh._{NR}$] Speed)

The Kolmogorov-Smirnov test to compare between the two sets of headway shows that there is no statistically significant difference between the two headway distributions (K-S = 0.59). Figure 9.22 presents the proportion plot for each situation. the lognormal distribution was found to be the best fit for the data with the parameters [mean=1.118 sec , Std = 0.651 sec , K-S = 0.257], (Figure 9.23). In conclusion, it is useful to find that there is a distribution for the headway between the new rear vehicle and the vehicle that change lane to the left. However, it is difficult to apply this result in the model, because, the dominant factor in the drivers' decision as to whether or not change lane to the left, is not the new rear vehicle. Therefore, the model will use the ABX threshold as a critical factor to judge the new rear vehicle when a driver wishes to change lane to the left.

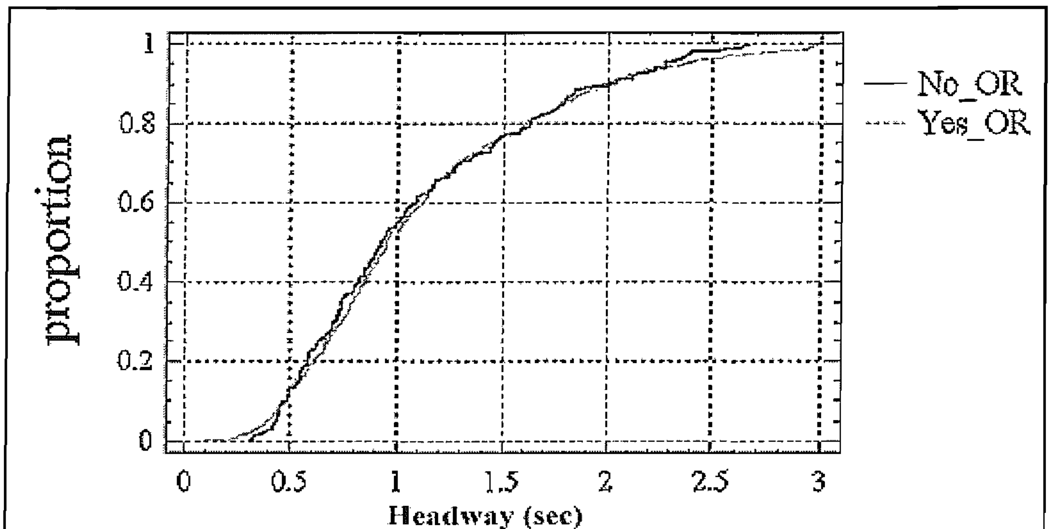


Figure 9.22: The proportion plot for the two headway series [with and without the old rear vehicle].

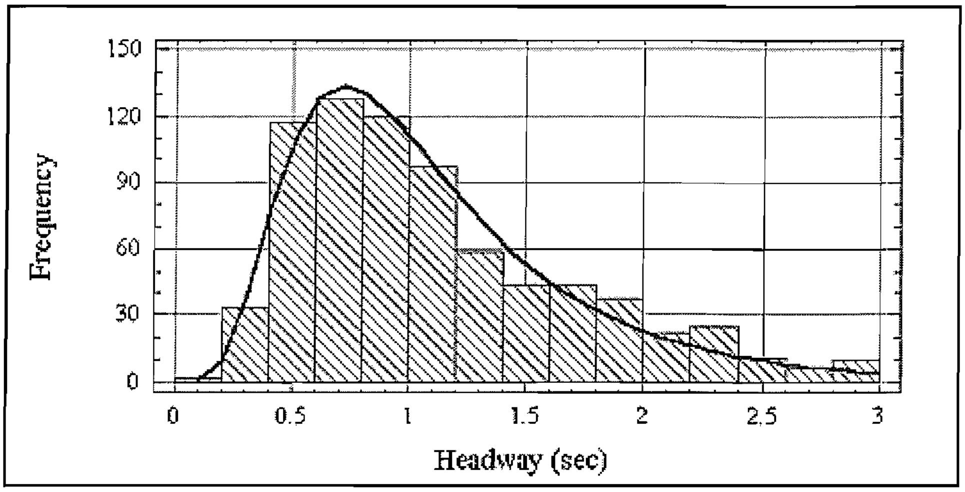
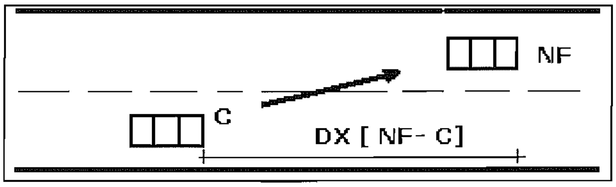


Figure (9.23) The lognormal Distribution test for the headway of the [Veh.NR] (lane-changing manoeuvre to the left)

9.3.2.3 The influence of the front vehicle in the new lane [Veh.NF]:



The presence of a front vehicle in the left lane [Veh.NF] is likely to play a critical part in a drivers' decision to change lane to the left, because, this vehicle may force the driver to change lane again to the right or reduce speed. If the [Veh.NF] is faster than the driver, [Veh.NF] is considered unlikely to affect the driver's decision to change lane to the left, unless it is very close. The filtered data did not contradict this (Figure 9.24)

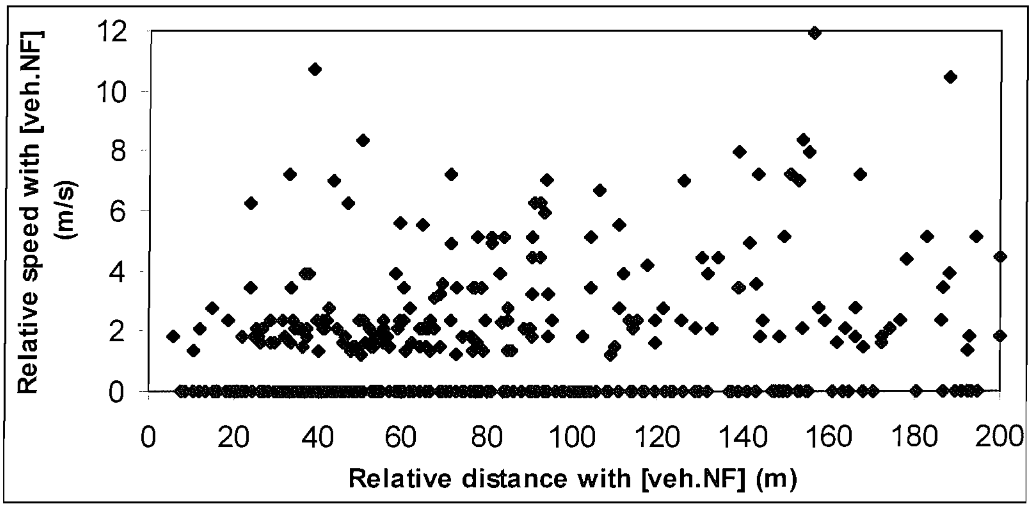
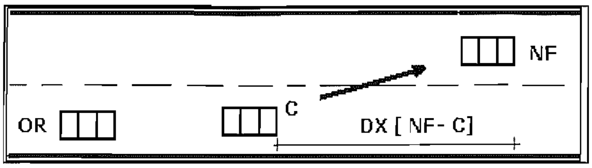


Figure 9.24: The relative speed against relative distance to the [Veh.NF] regardless of the position of the rear vehicles (new and old).

The situation is quite different when the driver is faster than the new front vehicle to the left. In order to analyse this situation two condition were considered:

1- The rear vehicle in the old lane is very close (headway less than 1.5sec), which means that the driver is under pressure to change lane change to the left. The data was filtered and analysed and Figure 9.25 presents the relative speed and relative distance relationship. As can be seen, a threshold can be defined to resemble the limits under which drivers can accept a lane change to the left. The equation was found to be :



$$DV_{(NF\text{-}C)} = -3.3729 * Ln(DX_{(NF\text{-}C)}) + 6.4687 \qquad ...9.8$$

2- The rear vehicle in the old lane is absent and the driver is not under pressure to lane change to the left. The data analysis did not show any significant difference to the previous situation and the relative speed threshold was very similar to the proposed one when the driver was followed by a close rear vehicle, (Figure 9.26).

As a result, the data was combined and final threshold was proposed for the whole situation. The equation was found to be:

$$DV_{(NF\text{-}C)} = -3.5277 * Ln(DX_{(NF\text{-}C)}) + 7.0591 \qquad ...9.9$$

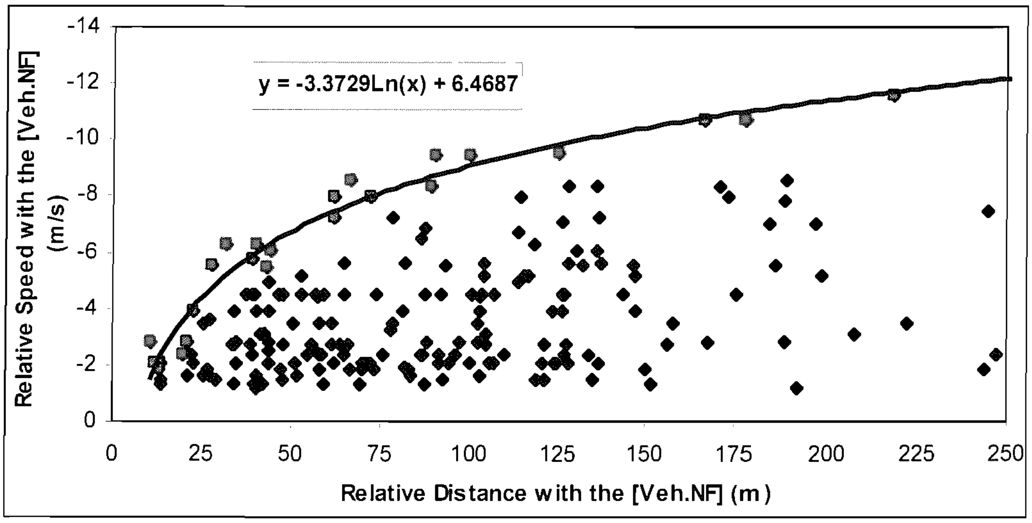


Figure 9.25: The relative speed against Relative distance to the [Veh.NF] (the rear old vehicle is very close and the new front vehicle is slower than the driver)

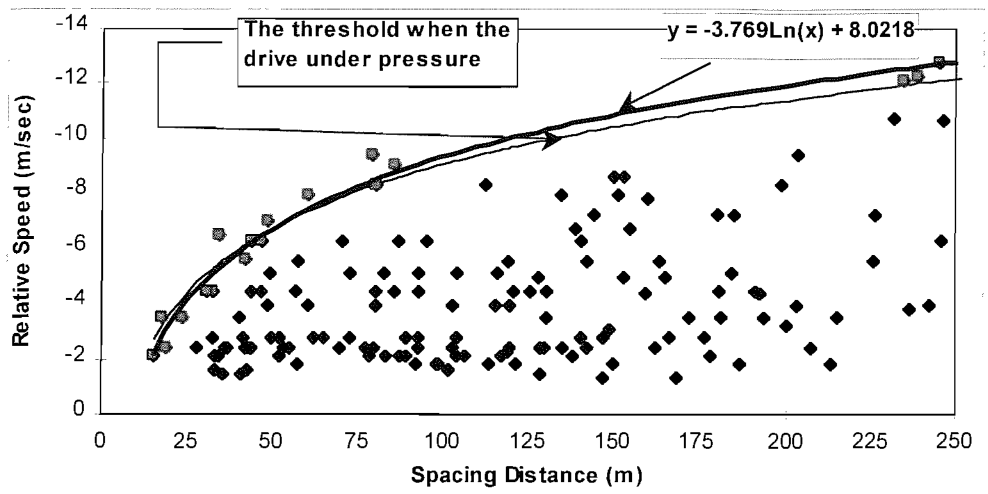


Figure 9.26: The relative speed against Relative distance to the [Veh.$_{NF}$] (the rear old vehicle is absent and the new front vehicle is slower than the driver)

Nevertheless, the proposed threshold raises questions about its relationship with the other thresholds proposed earlier in this Chapter and Chapter (7) (The approach process):

- Firstly, the threshold for lane changing to the right, which was discussed in Section 9.3.1.1 (Equation 9.1), has a conflict with the threshold of lane changing to the left (Equation 9.9). Figure 9.27 presents the two thresholds plus the relative speed Vs relative distance relationship with the new front vehicle to the left. The only explanation is that drivers in the conflict area where changing lane to the left to exit the motorway in the next junction. Therefore the model should not consider this part of the data in the lane changing process to the left unless the driver is forced to change lane to the left [lane drop or exiting the motorway].

- Secondly, in Chapter (7) (Modelling The Approach Process) it was found that driver will start their approach processes with a value of time to collision (TTC) between [7 to 17 sec], and the TTC was normally distributed with the parameters [mean = 11.7 and Std = 2.29]. Thus it is unrealistic to have drivers, who are going to change lane to the left, with a value of TTC less than his or her assigned TTC_{start} at which the approach process started. Figure 9.27 shows the TTC line with relation to the relative speed against relative distance data with the new front vehicle to the left.

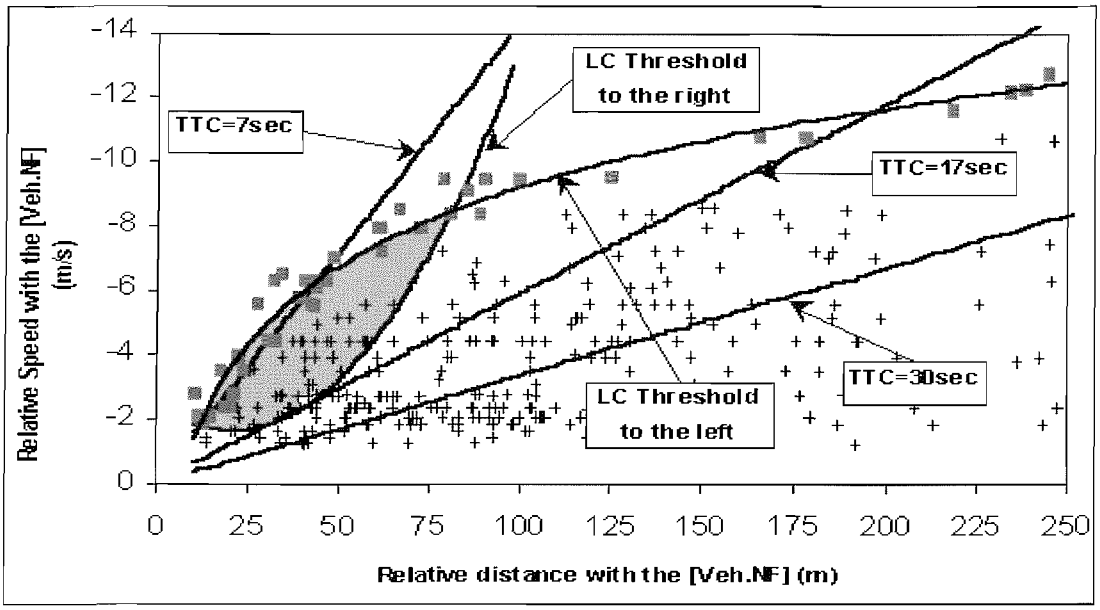


Figure 9.27: The LC threshold to the left regarding the [Veh.$_{NF}$] and the LC threshold to the right and the TTC levels for the relative speed against relative distance data to the new front vehicle to the left

In Conclusion, in order to consider the proposed threshold for changing lane to the left [Equation (9.9)], it is important to ensure that there is no conflict with the threshold to change lane to the right [Equation (9.1)] with respect to the same front vehicle to the left. The time to collision with the front vehicle to the left should be checked against the tolerated value of time to collision $TTC_{Tolerate}$.

The $TTC_{Tolerate}$ is the time to collision, with respect to the front vehicle to the left, above which the driver will accept changing lane to the left. For modelling purposes the $TTC_{Tolerate}$ was correlated with drivers' stimulus to change lane to the left [see Figure (9.18)] and its value ranged between 30 sec and the driver's TTC_{start}. The 30 sec is an arbitrary value which can be changed any time. (However, a sensitivity test was subsequently carried out to see the effect of considering alternative values).

9.4 THE LANE CHANGING LOGIC:

The lane changing logic is the decision process that the model will use to achieve a sound lane changing representation. The data analysis has developed several thresholds to guide the lane-changing model, and the overall decision structure needs to ensure that the complexity in the process does not lead to unsound results. Logically, the lane changing decision needs two stages, They are:

1- Detecting the necessity to change lane either to the right or to the left.

2- Evaluating the situation with the surrounding vehicles [ie relative speed and distance].

The lane changing decision process and the logic adopted in the model are shown in Figure 9.28.

Finally, some drivers prefer to use a certain lane, the so called the 'favourite lane', and they will not change lane as long as other vehicles can pass using a faster lane. However, when they approach a slower vehicle they will overtake it and return to the favourite lane as soon as possible. This phenomenon is called the 'lane hogging'. To simulate such behaviour the 'favourite lane' concept was introduced in Chapter (5). Every generated vehicle in the model will get a random number between 0 and 100, which will be compared with the percentage of the lane hogging drivers to determine whether or not the driver is lane hogging.

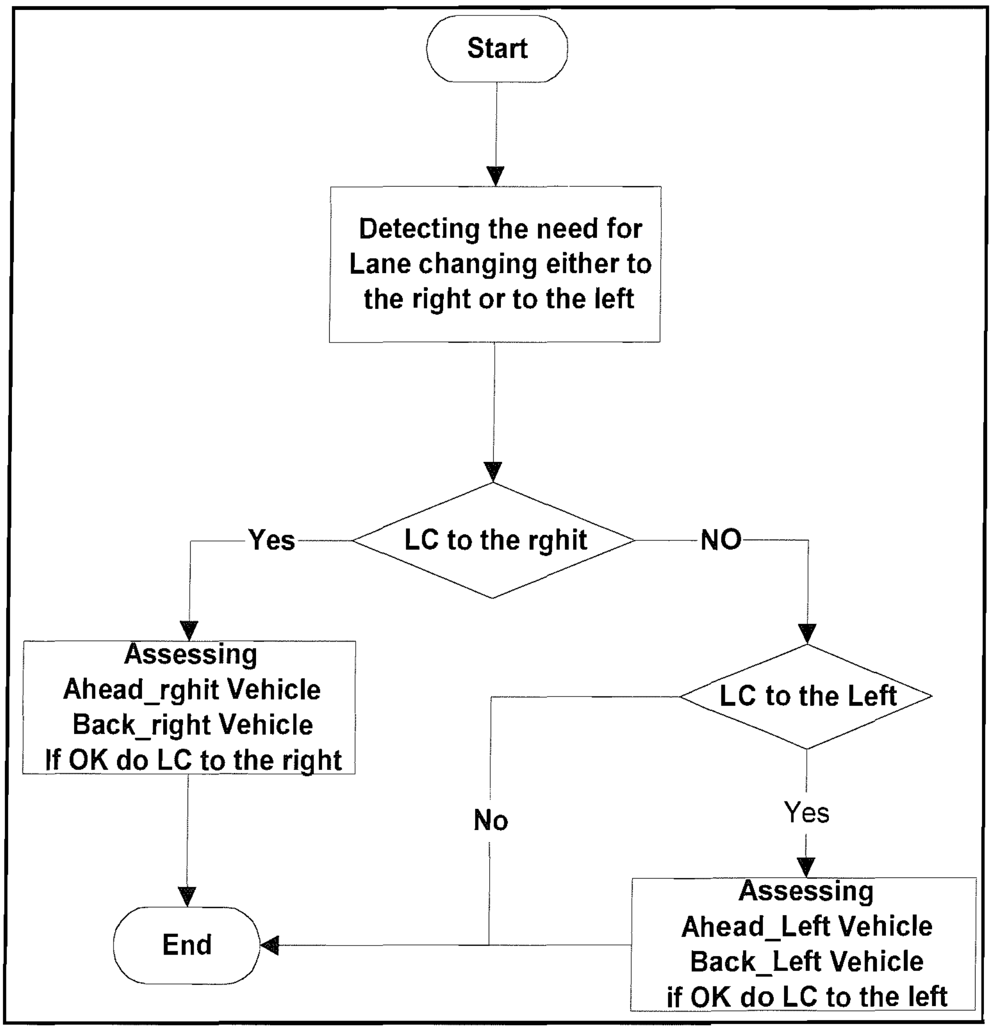


Figure 9.28: The Outline structure of the lane changing logic

9.4.1 Detecting the Lane Changing Manoeuvre:

In order to develop the logic for lane changing, two main aspects should be considered:

1- Constraints (ie the conditions to be included in the model to control certain behaviour). Such as:

- HGV's must not go into lane three as it is illegal for the HGV and Coaches to use the third lane of the motorway.
- The acceptable speed reduction that a driver will tolerate without changing lane: This was introduced to the first time by FERRARI in which he suggested that the driver speed tolerance ratio can be calculated from the equation (FERRARI P. 1989)[78]:

$$R = 1040 / Speed_{desired} \qquad (9.10)$$

Where:

R : Is the speed tolerance ratio.
$Speed_{desired}$: Is the driver desired speed km/h.

Later, this concept was used in the SISTM Model. Currently, the model uses two ratios:

a- A random number with a normal distribution N[0.05, 0.025] to apply for the heavy vehicles

b- A random number with a normal distribution N[0.1, 0.025] to apply for the light vehicles.

Thus, every vehicle generated in the model will have a random number [R6] to represent its speed tolerance according to the previous criteria. The model user can change the tolerance ratio distribution at any time. Finally, in the case where the leader is a lane hogger the driver will seek the overtaking regardless of the speed tolerance.

- The minimum duration time for the lane-changing manoeuvre: The driver will not be able to perform lane-changing manoeuvre until he or she finishes the previous one. Two conditions were proposed to determine the end of a lane changing manoeuvre:

a- The manoeuvre objective was achieved such as: overtaking certain vehicle.

b- The manoeuvre has lasted more than a certain length of time. This can be set by the user, however, the model uses a normal distribution N[mean=30sec and Std=5sec].

- The delay time for the lane-changing manoeuvre if one of the surrounding vehicles is currently changing lane. This is because drivers are constrained most by the following process and any change to the surrounding conditions will force the driver to evaluate the whole situation again, which takes a noticeable period of time. Currently, the model will use a delay of 2 sec if the new leader or follower is manoeuvring.

Other constrains can be added to the model in order to investigate its effects on the traffic volume.

2- The lane changing thresholds, which were assessed comprehensively in previous parts of this Chapter.

As was considered before there are two lane-changing situations, to the right and to the left, and consequently there should be two types of detection:

i) Detecting the lane changing manoeuvre to the right: The outcome from the decision process is a variable called Overtake1, which takes a value of one if the driver wants to change lane to the right, and zero if the driver does not want to change lane to the right.

ii) Detecting the lane changing manoeuvre to the left: This process will produce a variable called Overtake2, which takes a value of one when the driver wants to change lane to the left and zero when the driver does not.

If a driver has detected the lane changing to the right and left at the same time the priority is given to the right over the left unless the left lane is empty. finally, for modelling purposes, another restraints were added to the detecting process, They are:

- *If the current lead vehicle in the same lane is changing lane to the right the model will not allow a lane change to the right.*
- *If the current lead vehicle in the same lane is changing lane to the left the model will not allow a lane change to the left.*

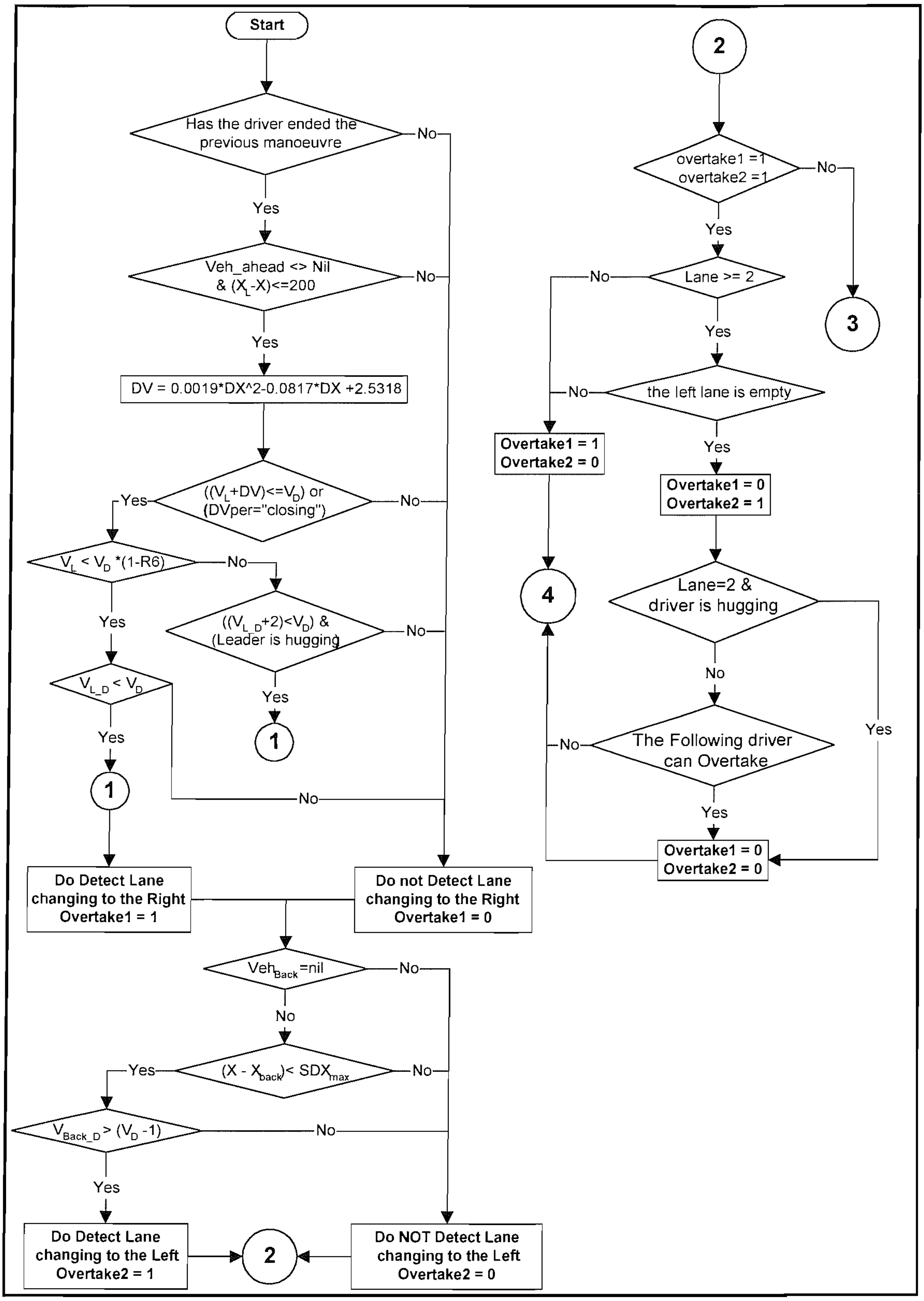


Figure 9.29a: The algorithm flowchart of detecting lane changing to the right and left.

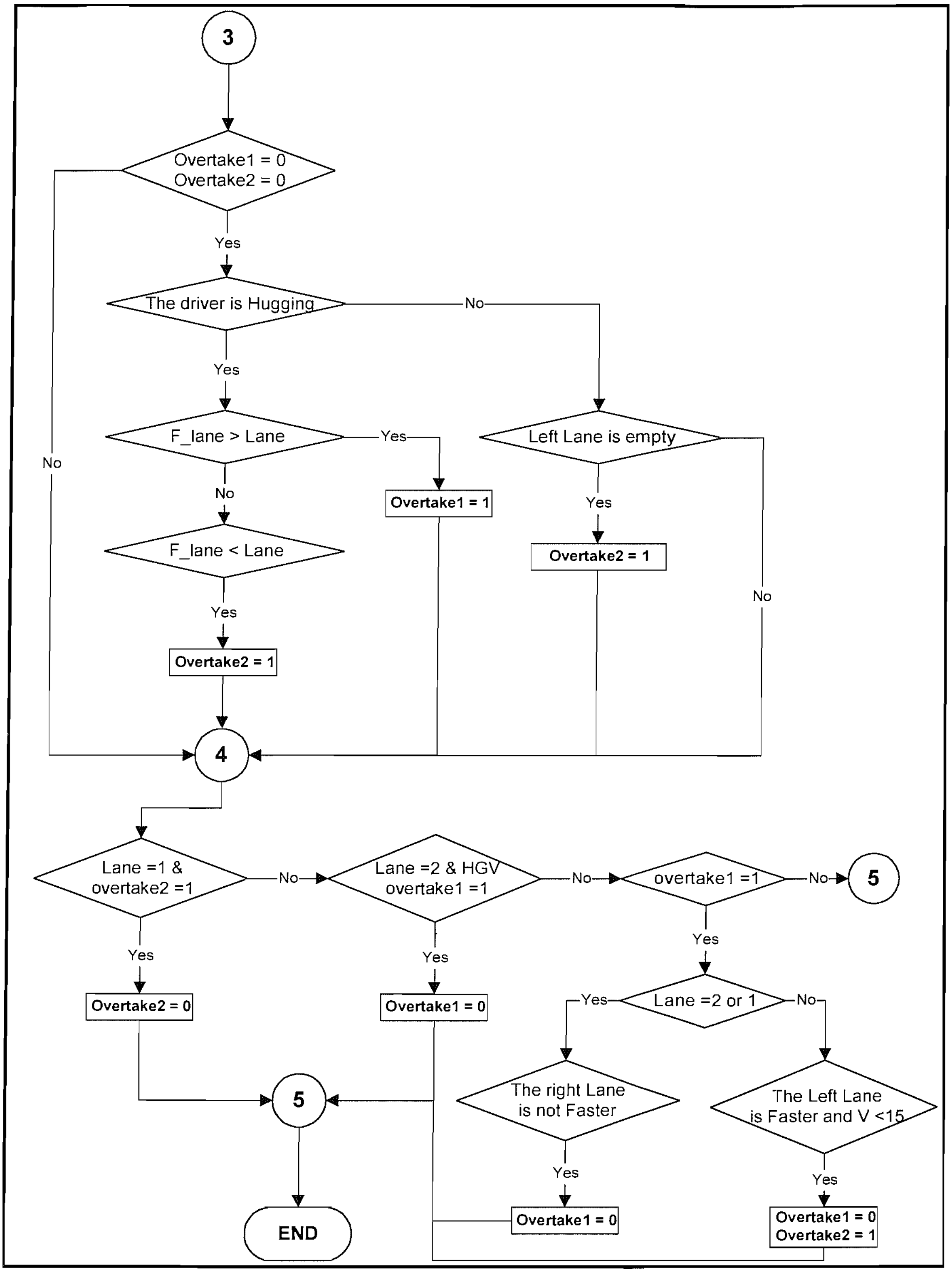


Figure 9.29b: The algorithm flowchart of detecting lane changing to the right and left.

This will help to prevent a platoon of vehicles from jumping simultaneously to the right lane or left lane. The delay between successive vehicles making the decision was assumed to be one second. The outcome from the detecting process should be one of these cases:

1- Overtake1 = 1 & Overtake2 = 0: Detect Lane changing to the right.
2- Overtake1 = 0 & Overtake2 = 1: Detect Lane changing to the Left.
3- Overtake1 = 0 & Overtake2 = 0: There is no necessity to change lane.

Figures (9.29a) and (9.29b) present the flowchart of the detecting lane changing process procedure. The next step in the model is to assess the situation with the adjacent vehicles to determine whether the lane-changing manoeuvre is possible or not.

9.4.2 Evaluating The Possibility to Change Lane To the Right:

When the results from the detecting process show that there is a need to change lane to the right (Overtake1 = 1 and Overtake2 = 0), the model will evaluate the possibility to undertake the manoeuvre. There are two assessment processes can be separated:

9.4.2.1 Evaluating The Front Vehicle In The right Lane [Veh_{NF}]:

The influence of the [Veh_{NF}] was analysed before in this Chapter [Section 9.3.1.2], and it is obvious that the current process will be cancelled if there is no lead vehicle to the right [Veh_{NF}]. In order to evaluate the front vehicle in the right lane two main situations were distinguished regarding its position, they are:

a- If ($X_{ahead\_right} <= X_{ahead}$): The model will suspend the lane changing manoeuvre in these two circumstances:

1- The relative distance with the Veh_{NF} is less than the minimum level ABX.
2- The relative speed is over the threshold presented in the equation (9.3).

b- If ($X_{ahead\_right} > X_{ahead}$), the model will suspend the lane changing manoeuvre in the circumstance when the relative distance with the Veh_{NF} is less than (200m) and the speed of the [Veh_{NF}] is less than the speed of the lead vehicle in the current lane. The (200m) limits will be discussed later in the sensitivity tests in this Chapter.

Figure 9.30 present the flowchart for evaluating the lead vehicle in the right lane to determine whether to proceed in the lane-changing manoeuvre to the right or not.

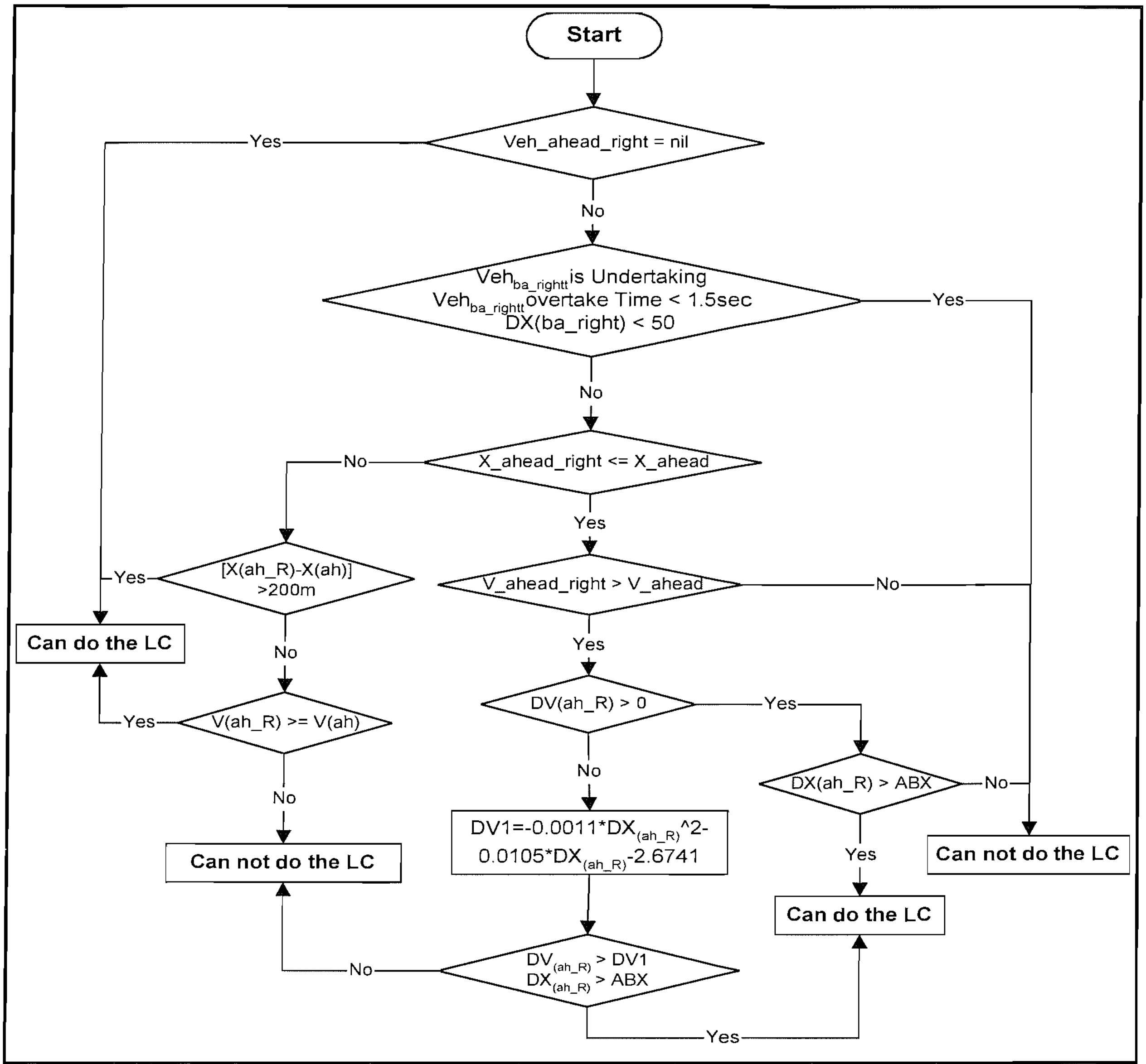


Figure 9.30: The flowchart of evaluating the lead vehicle in the right lane to determine the possibility to change lane to the right

9.4.2.2 Evaluating The Rear Vehicle In The Right Lane [Veh_{NR}]:

In paragraph 9.3.1.3 the influence of the rear vehicle in the right lane [Veh_{NR}] was analysed. Two main cases were considered to evaluate the relative situation with the Veh_{NR}. They are:

a- If ($V_{back\_right} > V_{vehicle}$): The first step is to assess the relative distance to the Veh_{NR}. It was proposed that the model should proceed in the lane-changing manoeuvre if the relative distance more than 200m. [This was considered as the maximum distance for perception]. However, if the relative distance was less than 15m the model will reject the manoeuvre. In other cases, the model will determine the critical relative speed DV_{cr} with the Veh_{NR}. A parameter called (STIMU) was presented to simulate the driver stimulus to change lane to the right. The STIMU parameter will have a value starting at Zero from the moment the driver realises the need to change lane, and build up with the time until it becomes One. (The time period lasts for one minute. However, a sensitivity test will be under taken later to check different values). In order to determine the DV_{cr}, two sub-situations were distinguished :

1- The current lane is lane Two: In this case the critical value of the relative speed DV_{cr} will be calculated from the equation:

$$DV_{cr} = RAND8*(DV_{(9.4)} + (DV_{(9.5)}-DV_{(9.4)}) * STIMU) \quad .....9.10$$

Where:

RAND8: Is a normally distributed random number N(mean=70%, Std=30%) to represents the differences between drivers

$DV_{(9.4)}$: The relative speed calculated from equation 9.4.

$DV_{(9.5)}$: The relative speed calculated from equation 9.5.

2- The current lane is Lane One: In this case, if the Veh_{NR} is able to change lane to the right the critical value of the relative speed DV_{cr} will be calculated from the equation 9.11. Otherwise, the equation 9.10 will be used to determine DV_{cr}.

$$DV_{cr} = RAND8*(DV_{(9.4)} + (DV_{(9.6)}-DV_{(9.4)}) * STIMU) \quad .....9.11$$

Where:

RAND8: Is a normally distributed random number N(mean=70%, Std=30%) to represents the differences between drivers

$DV_{(9.4)}$: The relative speed calculated from equation (9.4).

$DV_{(9.6)}$: The relative speed calculated from equation (9.6).

b- If ($V_{back\_right} \leq V_{vehicle}$): In this case, only the relative distance to the Veh_{NR} will be assessed. It was proposed that the model should proceed with the lane-changing manoeuvre, if the relative distance is more than ABX. (ie the minimum acceptable following distance by the driver). A minimum acceptable value of 15m was used.

Figure 9.31 presents the flowchart of the decision process to assess the back right vehicle in the faster lane. After the model has evaluated the front the rear vehicles in the right lane, the manoeuvre will be proceeded if there is no objection, otherwise the whole situation will be evaluated again in the next Epoch.

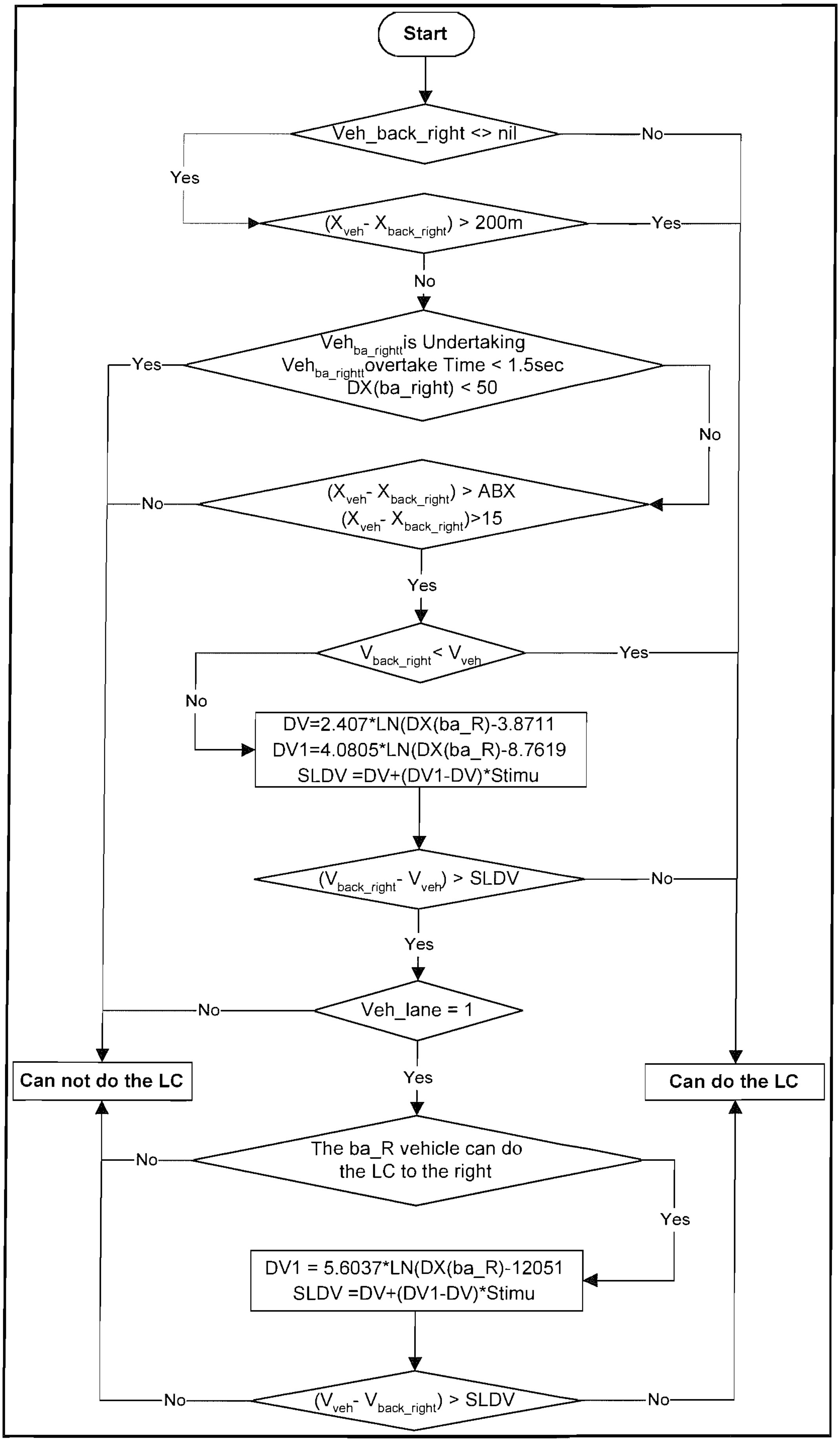


Figure 9.31: The decision process to evaluate the rear right vehicle in the faster lane.

9.4.3 Evaluating The Possibility to Change Lane To the Left:

The model will evaluate this possibility if the detecting process shows that changing lane to the left is needed (ie Overtake1 = 0 and Overtake2 = 1). The assessment will depend on the relative speed and distance with vehicles to the front and back in the left lane to achieve a decision. However, in a three lane motorway vehicles moving to the middle lane from the offside and nearside lanes at the same time may come into conflict. In order to avoid this problem, the model will prevent a lane-changing manoeuvre from the offside lane to the left if there is any vehicle is moving in from the offside lane within a safe relative distance of 50m

This evaluation process was separated into two parts, they are:

9.4.3.1 Evaluating The Front Vehicle In The Left Lane [Veh_{NF}]:

The data analysis was undertaken as in paragraph (9.3.2.3), and the conclusion was to use the value of the time to collision ($TTC_{Tolerate}$) that the driver can tolerate. The ($TTC_{Tolerate}$) can be determined from the equation:

$$TTC_{Tolerate} = 30 - (30 - TTC_{Start}) * STMU_{Left} \qquad .....9.12$$

Where:

TTC_{Start} : Is the acceptable time to collision by the driver to start his or here approach process.

$STMU_{Left}$: Is the driver stimulus to change lane to the left. The following distance of the rear vehicle governs this stimulus (see Figure 9.19).

Obviously, equation 9.12 will become pointless if the front vehicle in the left lane is faster than the lane changing vehicle and where the driver will not lose time. Nevertheless, the ABX value was used as the minimum acceptable relative distance with the Veh_{NF} to prevent any odd manoeuvre by the model. Figure 9.32 presents the flowchart of the assessment process for the lead vehicle in the left lane.

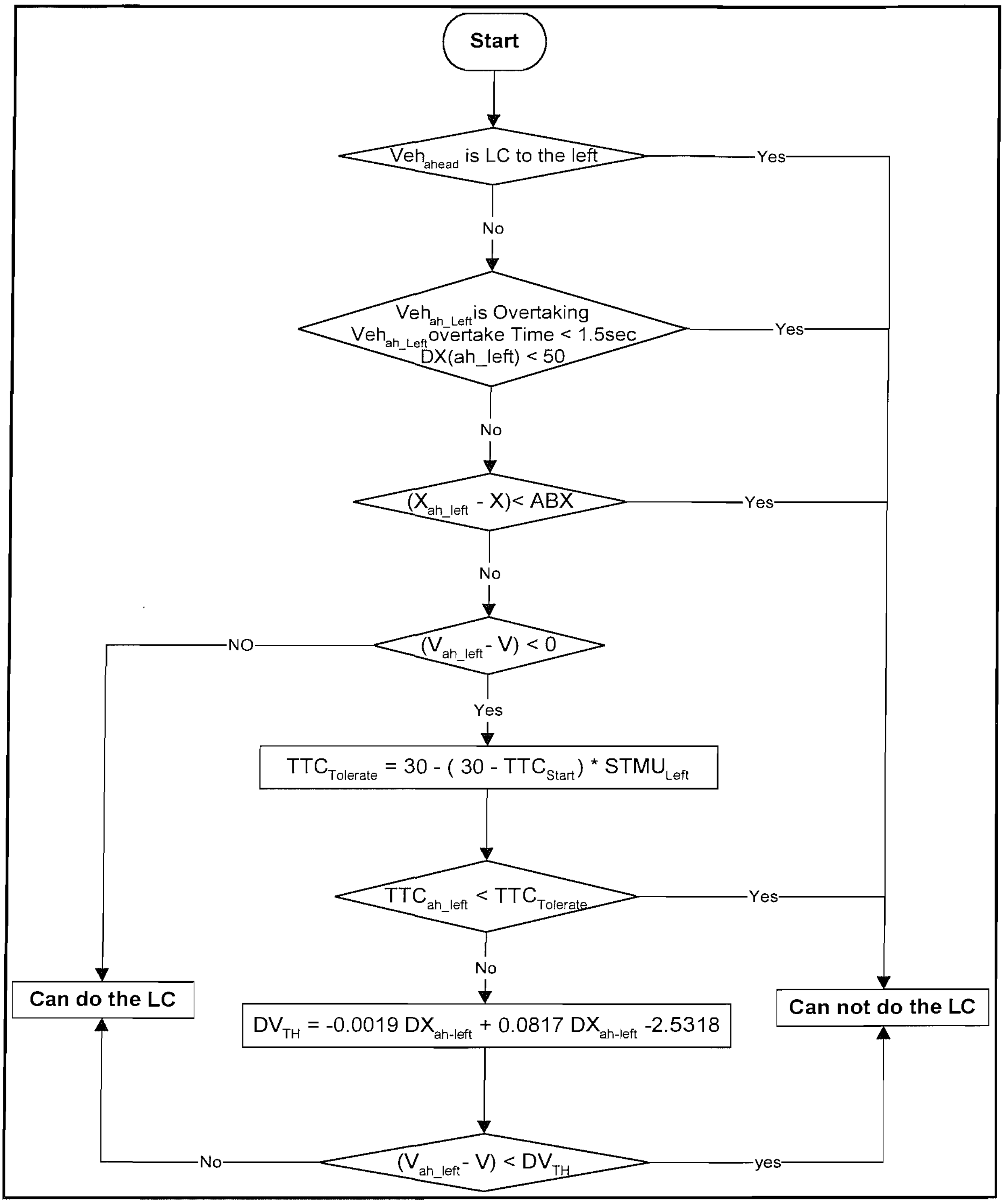


Figure 9.32: The Evaluation process of the front vehicle in the left lane

9.4.3.2 Evaluating The Rear Vehicle In The Left Lane [Veh_{NR}]:

As in the previous evaluation, ABX was used as the minimum acceptable relative distance with the Veh_{NR}. However, the rear vehicle will not influence the drive decision until it is faster than him or here and closer than 200m. In this case, the model will use the equation 9.7 to determine the acceptable relative speed with the new rear vehicle in the left lane.

Figure 9.33 shows the flowchart of the evaluation process regarding the new rear vehicle in the left lane.

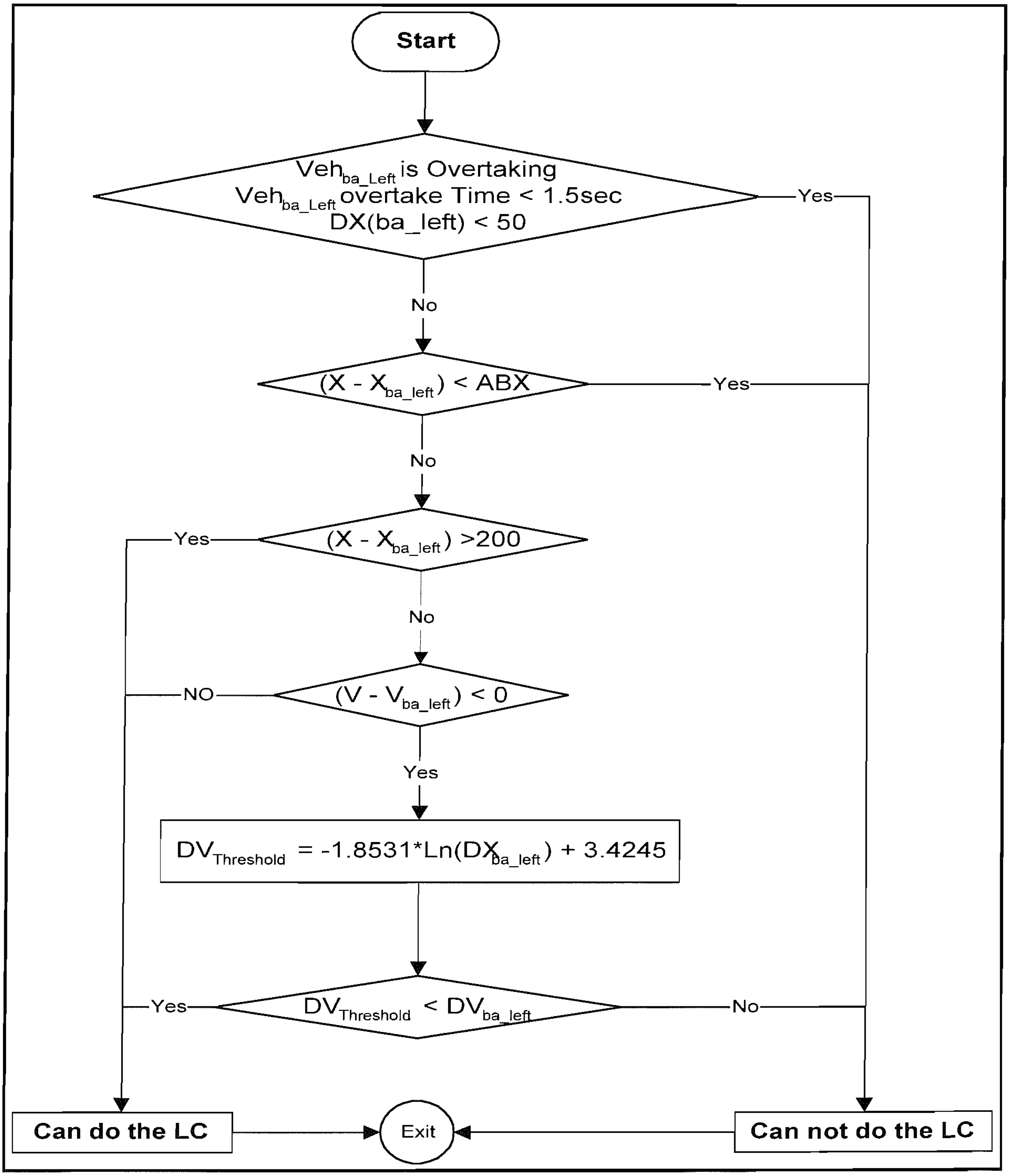


Figure 9.33: The Evaluation process of the rear vehicle in the left lane.

After the model has evaluated the front the rear vehicles in the left lane, the manoeuvre will proceed if there is no objection, otherwise the whole situation will be evaluated again in the next Epoch.

9.5 SENSITIVITY TESTS:

During the data analysis and the discussion of the lane changing logic several parameters were identified, and values were assumed according to common sense. However, this section will discuss the influence of such parameters on model results, and hence their validity. Performance was judged according to two criteria:

1- The average number of lane changing events in a one km section of the motorway: This will allow us to see if the total number of lane changing events has either increased or decreased with relation to a specific level of flow. However, for a parameter where its influence is only on a certain type of lane changing (ie: To the left or the right) only the affected lane changes will be numbered.

2- The Lane utilisation percentage between the three-lane motorway: This will allow us to see the overall effect of changing this parameter on the traffic and judge the points at which the traffic level of lane three becomes the highest.

The results were judged visually by checking the graphics for each parameter's test because of three main reasons:

1- The objective of these tests is to see whether the model is sensitive to the change in the value of its parameters or not.
2- Although statistical tests will give an accurate answer to how much the model is sensitive, it consumes time without answering the question how good is the model.
3- The lack of having a highly rated real data weakens the statistical judgements on the model's goodness.

In order to produce the above, the model was run to cover a flow range of 2000 to 6000 veh/h with a 100 veh/h flow step last for 10 minute. Also, the simulated motorway section was considered to be 10 km while the measurement section length was for one km after a 4.5 km of warm up section. For every run, the data was saved in a text file then it was transferred into an Excel file for the analysis. Finally, when the model was run to investigate a specific parameter all other parameters were set to their standard, which is:

1- Lane Hogging is 20%.
2- The stimulus time to change lane to the right is 60 sec.
3- The time to collision threshold to change lane to the left with the lead vehicle in the left lane is 30 sec.
4- The effective relative distance with the lead vehicle in the right lane is less than 200m.
5- The distribution factor for the lane changing thresholds to the right R8 is N[mean=70%, Std=25%].
6- The Lane changing duration is N[mean=30 sec, Std=5sec].

9.5.1 Evaluating The Lane Hogging Parameter:

The lane hogging parameter is basically the percentage of drivers who are lane hogging. The model was run three times with values of 0% , 20% and 50% in sequence. The results showed that the total number of lane changing events does not vary very much between the different value of the lane hogging parameter, although, there was a slight reduction in the number of LC events corresponding to an increase in the lane hogging proportion from 0% to 50%, (Figure 9.34).

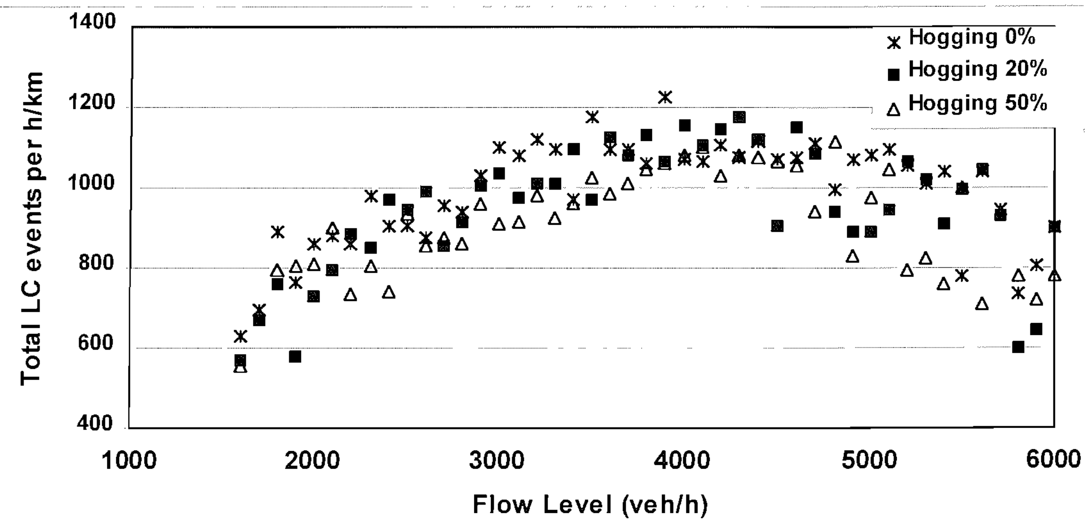


Figure 9.34: The total number of lane changing events per km for different value of lane hogging percentage [simulation output].

The next step was to check lane utilisation. The output shows that in high-flow situations of more than 4000 veh/h, the overall traffic distribution between lanes dose not change with relation to the lane hogging percentage (Figure 3.35). However, in low-flow situations of less than 3000 veh/h the distribution has changed. Clearly, the portion of traffic on lane three, (ie the fastest lane), increased with an increase in the lane hogging factor, whilst the opposite happens to lane one (ie the slowest lane). Also, it seems likely that the flow level at which the crossing point between the traffic percentage of lane three and two becomes lower as the percentage of lane hogging becomes higher. It may be concluded from this test that the presence of a low level of lane hogging gives some realistic behaviour to the model. Though, at high level the results become unrealistic at low demands.

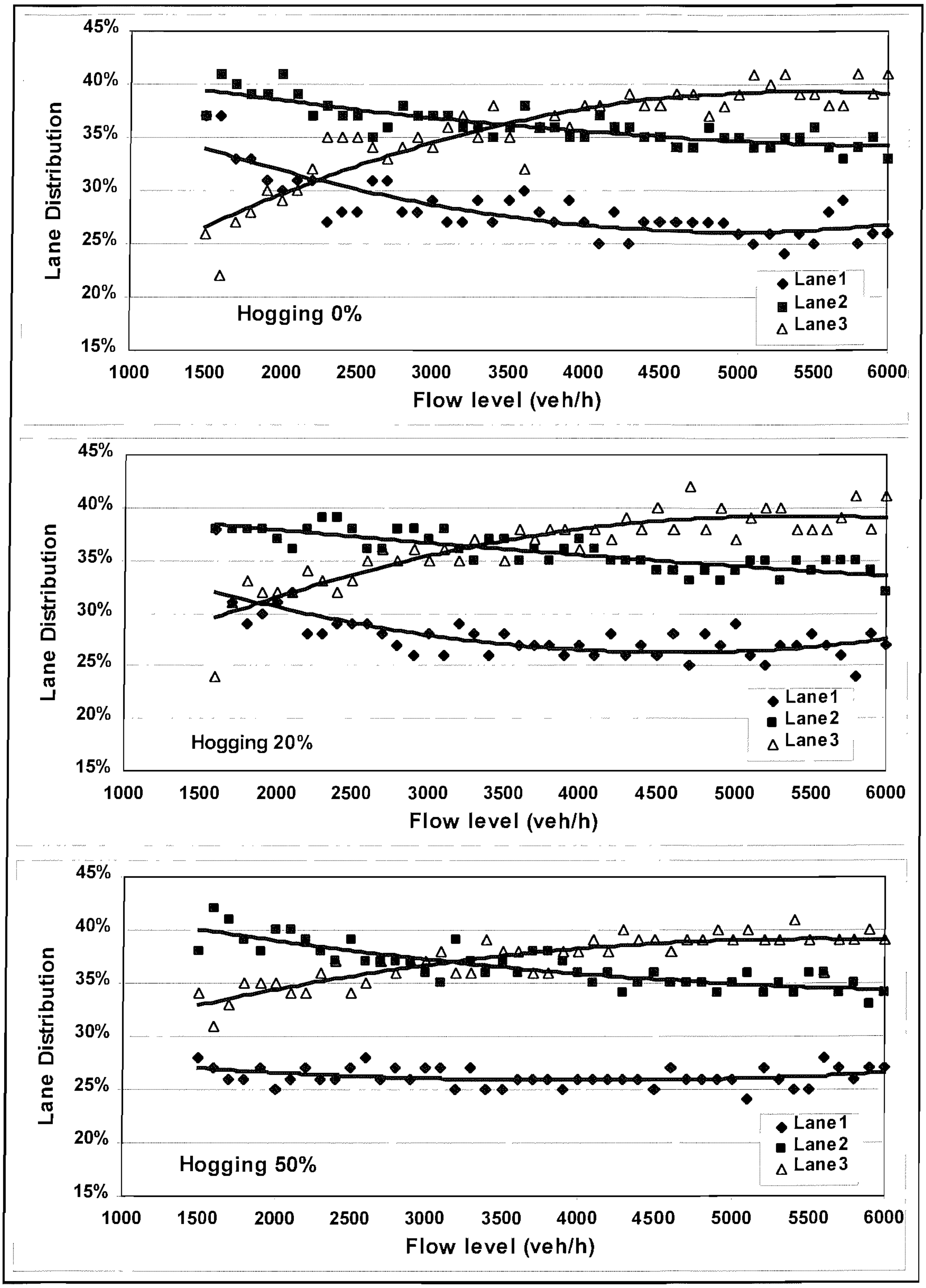


Figure 9.35: The lane utilisation over a three-lane motorway with different value of lane hugging parameter [simulation output].

9.5.2 Evaluating The Driver Stimulus Time To Change Lane To The Right:

In paragraph 9.3.1.3 three thresholds were identified from the data analysis to determine the influence of the rear vehicle in the right lane on the drivers' decision to change lane. Then relationship between these thresholds was presented in paragraph 9.4.1.2 where the lane changing logic was developed. It was suggested that the driver stimulus to change lane to the right would build up with time, and it reaches maximum after a certain time called the stimulus time. The model was run with various stimulus-times [0.5, 1 and 2 min]. Then the number of lane changing events to the right and the traffic distribution between lanes was compared for these various settings. The results indicated that number of lane changing events to the right by the model is not affected by the change in the stimulus time. Figure 9.36 shows the number of LC events over the flow levels regarding the considered situations. Also, the traffic distribution over the three lanes was checked and no significant difference was found among the studied situations too, (Figure 9.37).

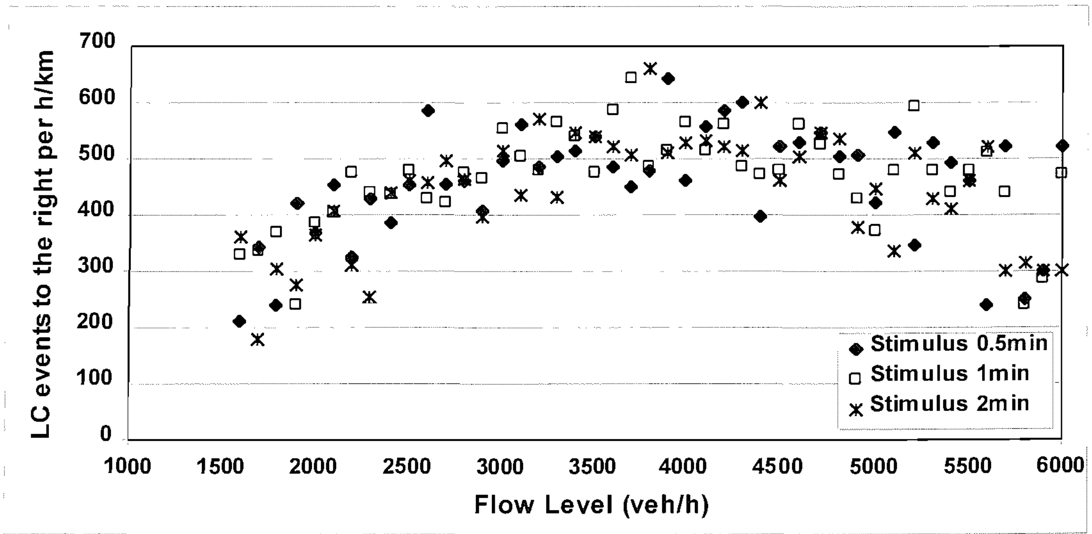


Figure 9.36: The total number of LC events to the right regarding different stimulus Time [simulation output]

9.5.3 Evaluating The TTC Threshold To Change Lane To The Left:

The time to collision threshold TTC_{Thresh} was proposed to determine the starting point at which the vehicle in the left lane would influence the drivers' decision to change lane to the left. This was discussed in paragraph 9.3.2.3, where a value of (30 sec) was suggested for the model. In order to investigate the influence of this parameters on the model performance, several values of TTC_{Thresh} were applied [25sec, 30sec and 35sec]. The model was run and the output was sorted and analysed. The results showed that the number of lane changing events to the left fell with an increase in the TTC_{Thresh} from 25 sec to 35 sec when the flow demand was in the range of 3500 to 4500 (veh/h). Though the difference appears small at other demand levels. Figure 9.37 presents the lane changing events to the left over the flow level for the three considered situations.

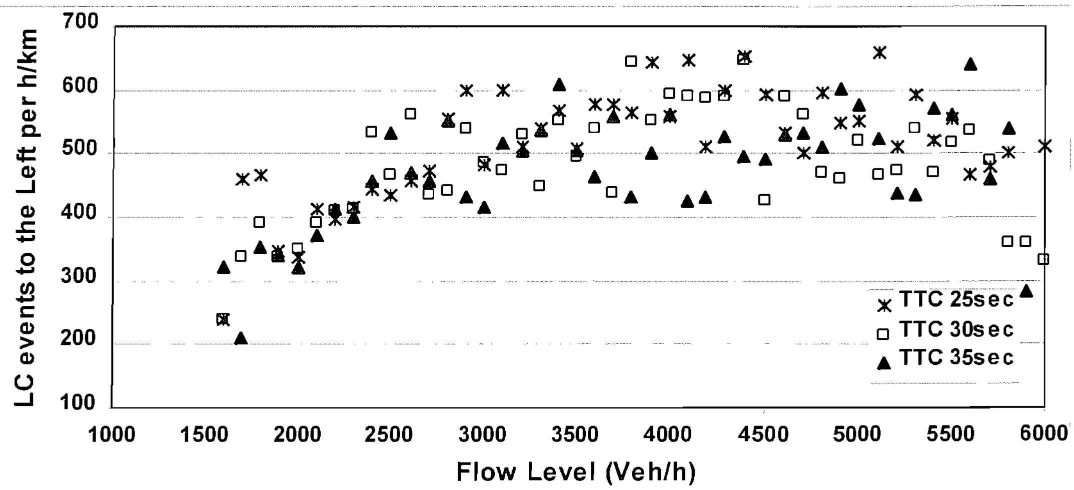


Figure 9.37: The lane changing events to the left for different value of TTC.

The influence of the TTC_{Thresh} became clearer when the traffic distribution between the lanes was investigated. The results showed that when the value of TTC_{Thresh} increased there was a drop in the flow level at which the traffic volume on lane three become equal to the lane two, (Figure 9.38).

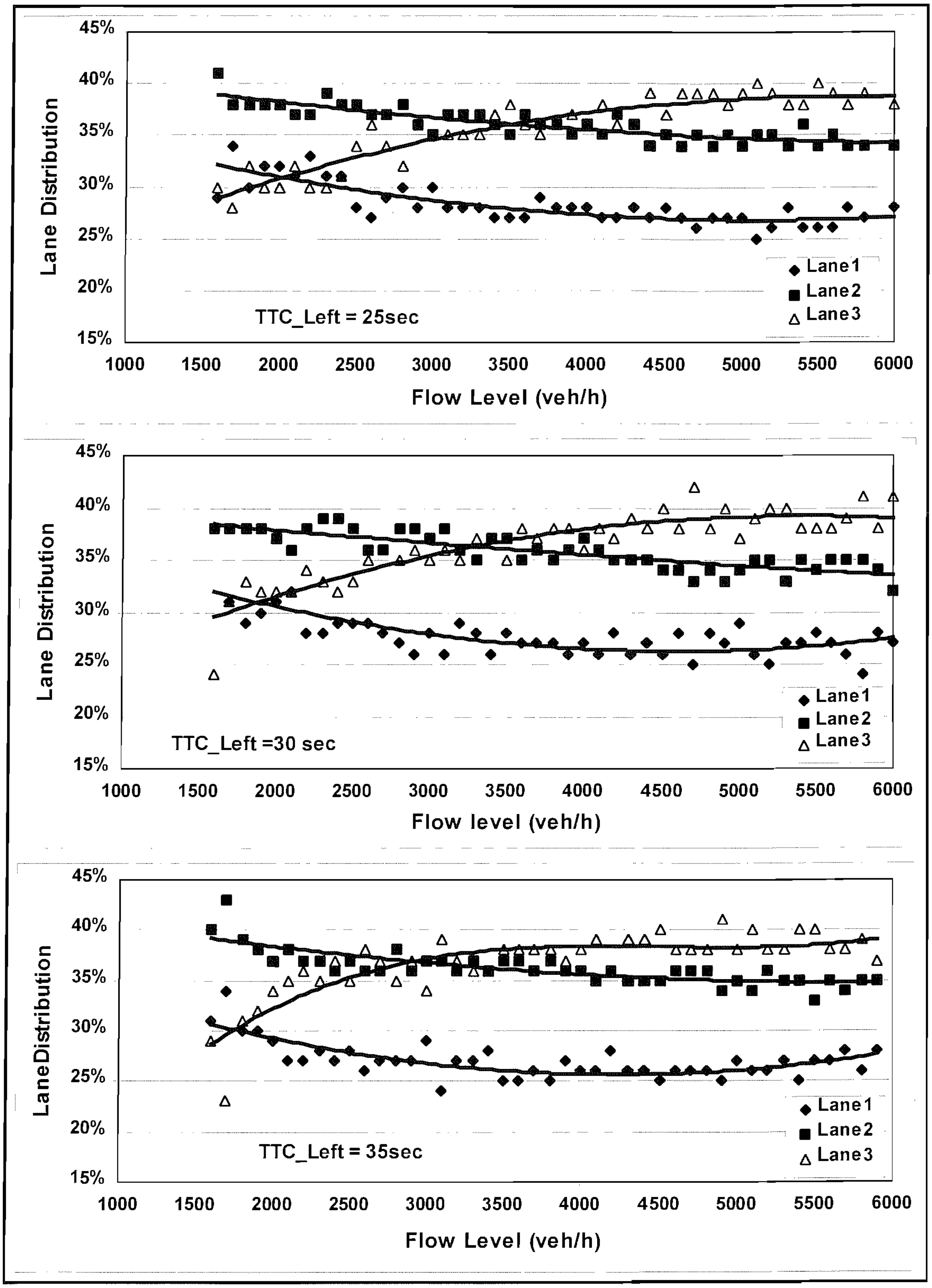


Figure 9.38: The lane utilisation for three lanes motorway regarding different TTC_{Thresh} stimulus levels [simulation output]

9.5.4 Evaluating The Effective Spacing With The Lead Vehicle In The Right Lane:

It was assumed in the lane changing logic that the driver would not consider the influence of the vehicle in the right hand lane unless the relative distance was less than 200m. This assumption was proposed to simulate the drivers' disability to perceive information such as relative speed when the relative distance is very long. However, a sensitivity test was undertaken to investigate the change on the model output if this relative distance was reduced to 100m. The results showed that there is no apparent influence on the model output, (Figure 9.39). (Most of the time, the lead vehicle in the right lane is faster than the modelled vehicle and will not influence of this vehicle).

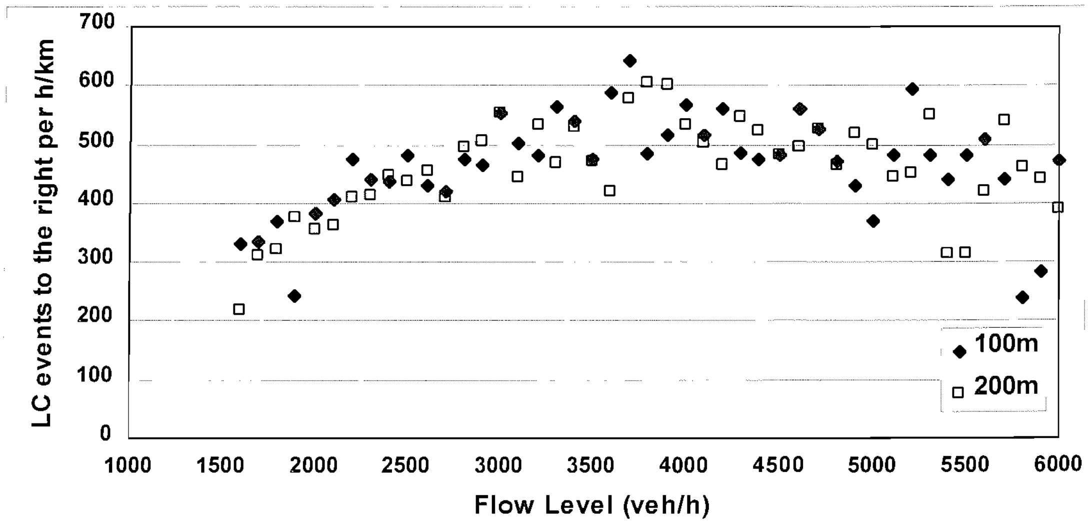


Figure 9.39: Lane changing events to the right regarding to different value of critical relative distance with the lead vehicle in right lane

9.5.5 Evaluating The Distribution Factor For The LC Threshold To The Right:

Because the suggested thresholds for the LC decision represents the most extreme measured situation, it is expected that not every driver will accept these thresholds, although identifying the real distribution for such thresholds is almost impossible. Therefore it is suggested that a factor [Rand8], with a normal distribution could be introduced to the model to achieve such a variation among drivers. This idea arose after the model generated a very high number of lane-changing manoeuvres per h/km. The test objective was to assess the influence of Rand8 on the model by using different values for the normal distribution parameters (mean and Std), they are: N[80%, 15%] , N[60%, 25%] and N[75%, 25%]. As the number of lane-changing events fell with decrease in the mean value of Rand8, especially when the flow level was more than 3500 vel/h. Although, there is a large difference between the 80% and 60% value in the number and the shape of the outcome, the difference is much less between 80% and 75% . Figure 9.40 presents the average lane changing number to the right over the various flow levels.

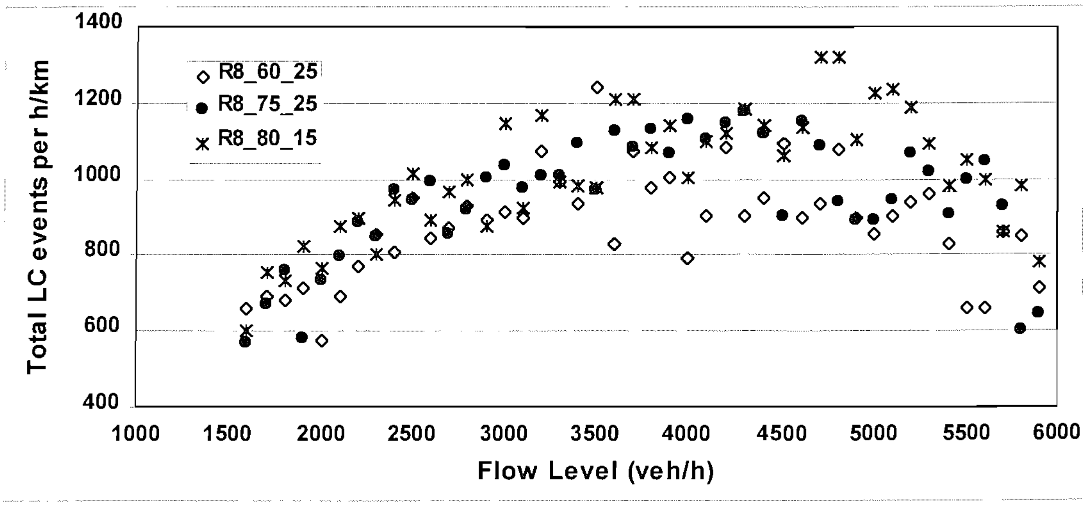


Figure 9.40: The lane changing events to the right regarding to different value for the distribution of the Rand8 factor

From consideration of the traffic distribution over the lanes, the influence becomes obvious in high-flow situations. The reduction in the value of the distribution's mean leads to a smaller difference between lanes two and three. However, the traffic proportion of lane one increases with the decrease of the mean value of Rand8. This result was expected as the ability to change lane to the right reduced with reduction in the mean value of Rand8 (Figure 9.41).

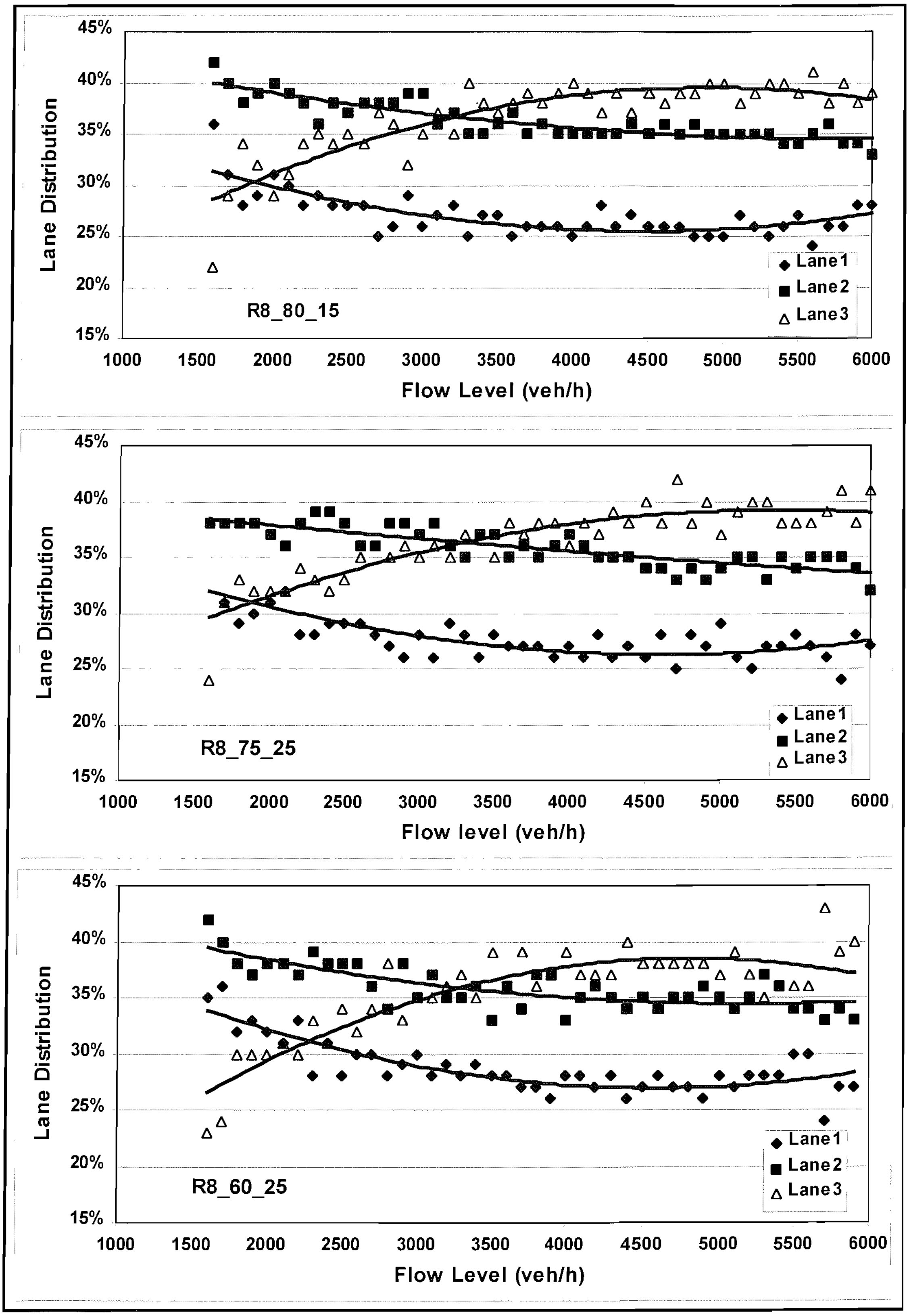


Figure 9.41: The lane utilisation for three lanes motorway regarding different distribution for Rand8 factor [simulation output]

9.5.6 Evaluating The Influence Of Different Speed Tolerance:

In practice, it is difficult to believe all driver will always attempt to overtake every slower vehicle in front, whatever the speed difference is. (the speed tolerance concept, was discussed in paragraph 9.4.1.). This test was conducted to check the effect of using different speed tolerance rate on the model output. Three values were used [8%, 10% and 12% for cars]. The results showed that number of lane changing manoeuvres to the right did not vary with the change in the speed tolerance rate, (Figure 9.42). The interpretation is that drivers migrate to lane three due to the need to overtake the lead vehicle and thereby constrain the number of further lane changing events. This interpretation is supported by lane distribution results, where it was clear that the traffic on lane three fell with increase in speed tolerance (Figure 9.43).

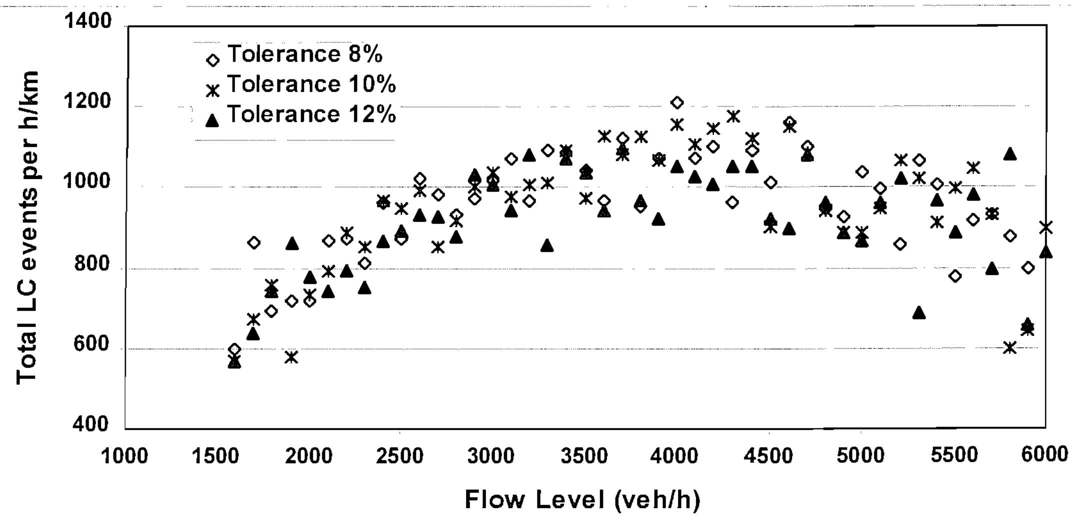


Figure 9.42: The lane changing events to the right regarding to different speed tolerance rates

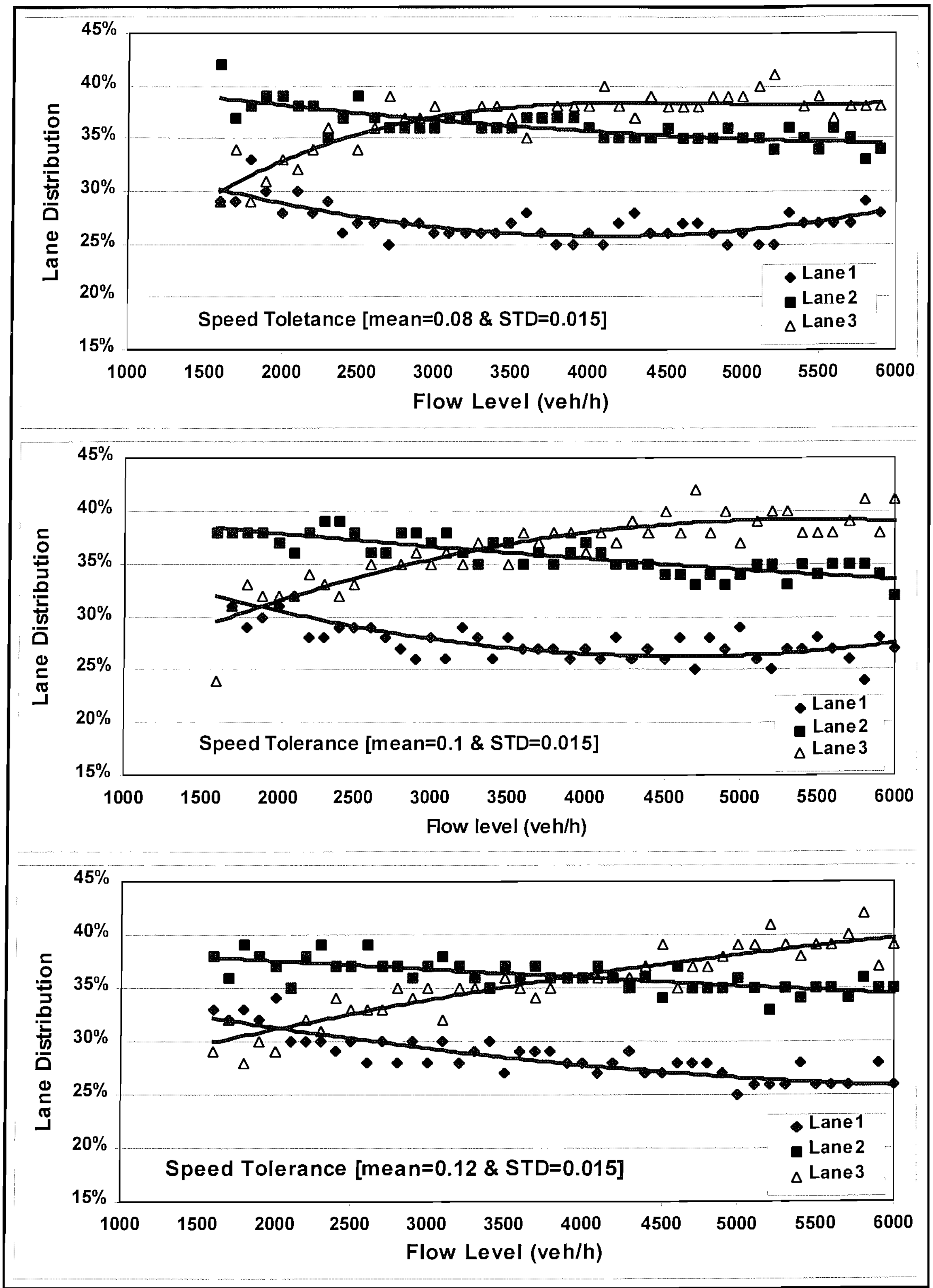


Figure 9.43: The lane utilisation for three lanes motorway regarding different Speed tolerance rates [simulation output]

9.5.7 Evaluating The Influence Of Different Manoeuvre Time:

In order to prevent a simulated vehicle changing lane frequently in a very short time the concept of minimum manoeuvre time was introduced in the model. This also controls the number of lane-changing events without changing the lane-changing algorithm itself. The sensitivity test was conducted by using two values for the manoeuvre time [30 and 40 sec]. The result was as expected, with the average number of lane-changing events decreasing with the increase in lane changing time. Figure 9.44 show the average lane-changing events with different manoeuvre time.

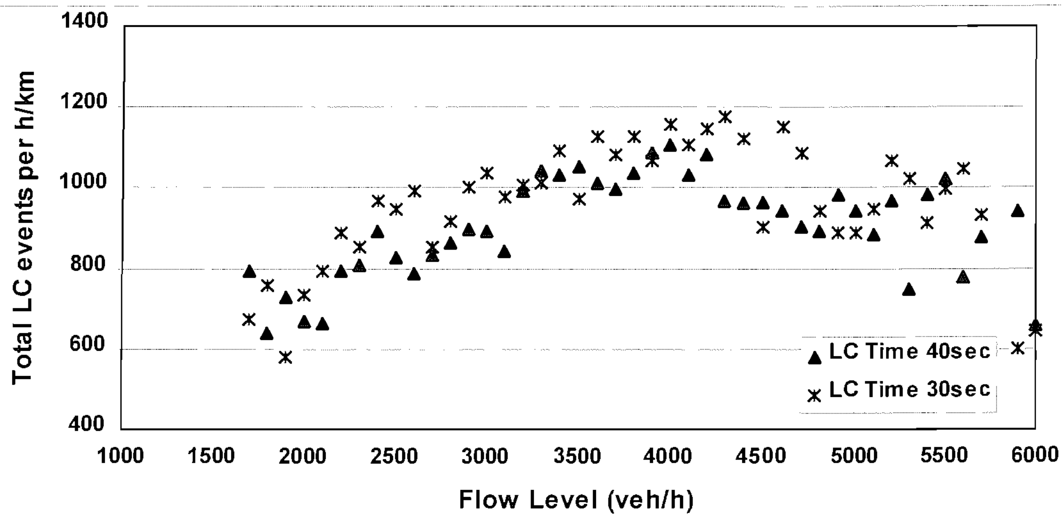


Figure 9.44: The average lane changing events with different manoeuvre time

9.6 COMPARISON OF MODEL OUTPUT WITH REAL DATA:

Obtaining appropriate real data is one of the difficulties to face the lane-changing validation. The data has to present two main aspects:

1- The lane utilisation, which is the proportion of traffic presents on each lane for a various ranges of demand.
2- The number of lane changing events on a certain length of a motorway in the time unit.

Every section of road has its own characteristics, i.e.: uphill, downhill, before or after a junction, ...etc. These characteristics influence the character of the data shape and may bias the results. Thus collecting a database from several sites is important. Because of the difficulty of achieving appropriate experiments in the time available, previous databases were considered.

In 1989, TRG undertook several surveys on the M25 and M6 motorways to investigate breakdown characteristic and the motorway capacity (HOUNSELL N. B. & MCDONALD M. 1994)[9]. The data was collected from several locations over several days using video cameras set on over bridges. Figure 9.45 presents the lane utilisation for three lane motorway extracted from the study data. As the study was concerned with flow breakdown, the data was collected at periods of high demand (Flow over 3500 veh/h). Figure 9.46 presents the lane utilisation from the simulation model. Obviously, the model output is well acceptable in matching the real data in the high demand level, where the lane three percentage around 40% and lane two and one about 35% and 25% respectively.

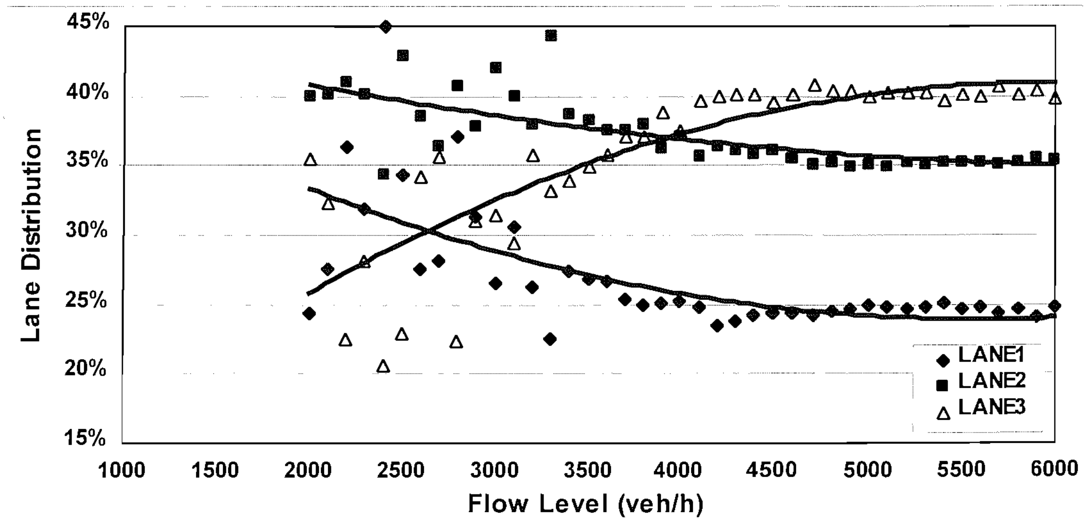


Figure 9.45: The Lane Utilisation over the flow level [real Data] 1989 TRG survey.

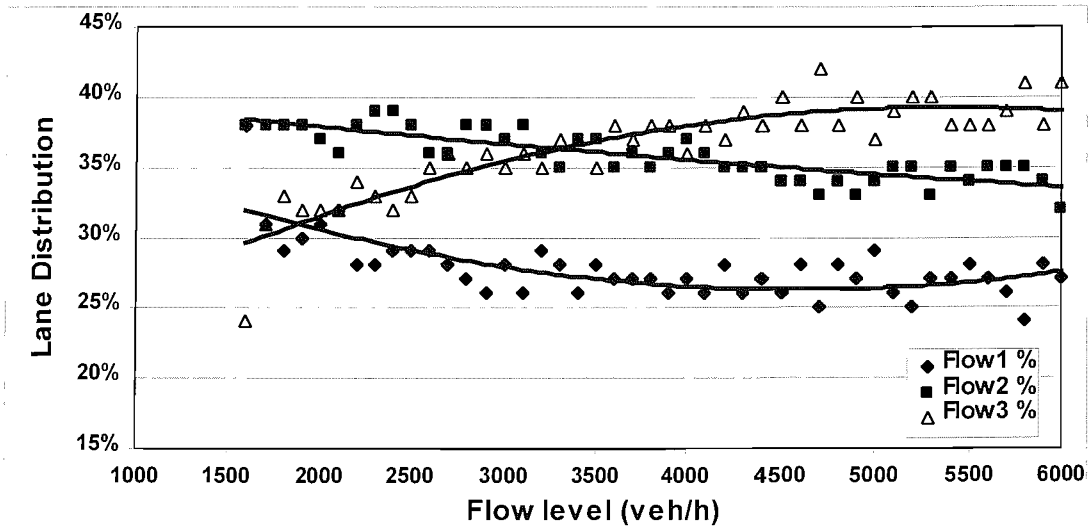


Figure 9.46: The Lane Utilisation over the flow level [Model output].

The number of lane-changing events with respect to the flow level was not available from the survey data. However, another survey undertaken in 1994 by TRG was considered (MCDONALD M., & BRACKSTONE, M. 1994)[26]. It used video cameras positioned on pedestrian bridges over the M1 and M25 motorways. A comparison between the model output and the real data for the number of lane changing events revealed that the model seemed to generat an acceptable lane changing events for demand level up to 5000veh/h. However, for flow levels over 5000 veh/h, the model generated more lane-changing events than measured. Figure 9.47 shows the number of lane changing events per h/km for various flow levels from the real data and the simulation model.

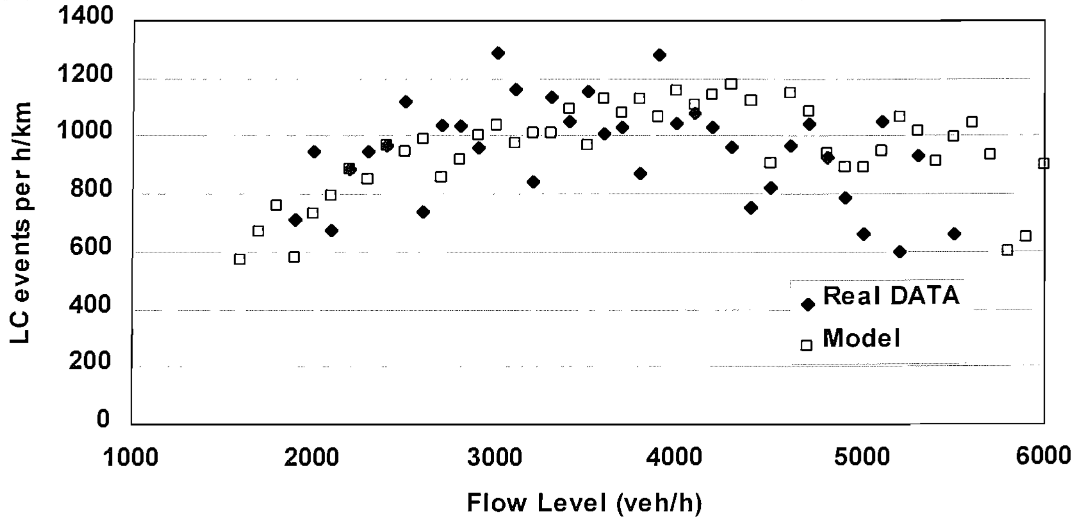


Figure 9.47: Comparison of LC events with flow for the real data and the simulation model output.

An interpretation of the results is that drivers become more reluctant to change lane as the flow level increases, not because the opportunity is not available, but because they do not foresee any advantages from doing so. Although the model has incorporated such behaviour, it is nearly impossible to achieve a perfect result due to the lack of definition of the critical situation at which drivers change their behaviour. Nevertheless, the difference between the model output and the real data is small when the overall model performance is taken in context.

CHAPTER 10

APPLICATION: (ASSESSING THE BENEFITS OF AN ACC SYSTEM)

10.1 INTODUCTION:

Throughout the last decade a number of advanced system concepts for improving safety, efficiency, environmental compatibility and comfort of driving have emerged. One of these new systems is the Automatic Adaptive Cruise Control (ACC). Its main objective is to help the driver to maintain a safe headway with the lead vehicle and provide an alarm to minimise reaction time in an emergency. Because ACC systems are being introduced, several studies have already been undertaken to evaluate their usefulness. Most of these studies have used driving simulators in order to emulate the presence of the ACC system in the vehicle. The results were not consistent and there is uncertainty about the real benefits of the ACC. Lena NILSSON found that there would be more collisions among ACC users than unsupported drivers in approaching a stationary queue, though subjects favoured the ACC system (NILSSON 1991)[80]. Another study by HOEDEMAEKER and BROOKHUIS reported that some drivers would benefit from such a system to reduce their reaction times, although others did not, and a clear safety benefit could not be gained from an ACC system (HOEDEMAEKER M. & BROOKHUIS K. A. 1998)[81]. This Chapter assesses the benefits of developing ACC systems using the simulation model.

10.2 THE ACC SYSTEM:

As ACC systems have a huge market potential, car manufacturers are reluctant to publish detailed algorithms and system characteristics. The system upon which the study will be undertaken was adapted from the TRL report that describes ACC systems in their simulation model SISTM. Basically, the applied ACC is a distance control system, which can be switched off and on by the driver.

10.2.1 The System Parameters:

Several parameters were introduced to define the ACC limits under which it can operate. Some parameters are usually set by the driver, and in order to reflect the individual choice a normal distributions were used instead of fixed values. The ACC parameters are:

1- The desired time headway to be adopted under distance control mode (Th_0): Usually, this value is set by the driver but a minimum value of 1.2 sec is suggested to prevent dangerous driving. The standard deviation was set to 0.2sec and the mean is 1.5 sec.

2- The maximum time headway to enter the distance control mode (Th_E): This must be greater than the desired headway. A value of 2.5sec was used.

3- The maximum time headway to remain in distance control mode (Th_R): This must be greater than the maximum time headway to enter distance control. A value of 2.75sec was used.

4- The maximum acceleration to be adopted under distance control mode (AC_{max}): This is depends on the ACC system, and a value of 0.75 m/sec^2 was accepted.

5- The maximum deceleration to be adopted under distance control mode (DC_{max}): This is depends on the ACC system, and a value of -2 m/sec^2 was accepted.

6- The minimum speed to enter distance control mode (V_{min}): A value of 8 m/sec was accepted.

7- The maximum speed to remain in distance control mode (V_{max}): This is set by the driver, so, the system will adopt the driver's desired speed.

8- The minimum time between leaving and re-entering distance control mode (DT_{min}): This also represents the minimum time to be spent in distance control before an unforced lane changing is made (mean = 30sec & STD = 5sec).

10.2.2 The System Logic:

An ACC system needs an algorithm to transfer the perceived data from its sensors (ie. relative speed and distance) into actions (deceleration/acceleration). In this study, two equations have been applied independently while keeping the other parts of the system logic the same. A comparison between the results from each equation will highlight the importance of such equations to the ACC outcome and its ability to benefit traffic safety and management. These equations are from two car manufacturers.

1. The first equation was introduced by TRL, (Transport Research Laboratory 1997)[82]. It has three main parameters (K_1, K_2 and M) which incorporate the values of the relative speed, relative distance and speed level. Equation 10.1 represents the ACC acceleration/deceleration formula:

$$a_{ACC} = \frac{1}{M} * \left(DX_t * \left(\frac{K_2}{T} - K_1 \right) - DX_{t-1} * \left(\frac{K_2}{T} \right) + K_1 * V * Th_0 \right) \qquad \ldots.10.1$$

Where:

a_{ACC} : The acceleration/deceleration of the ACC system m/sec^2.
M : The mass of the vehicle (200kg).
DX_t &DX_{t-1} : The inter vehicle distances in meters in the previous and current epochs.
T : The Epoch time (0.125 sec).
K_1 & K_2 : damping constants (-31.25 kg/sec^2 and 500kg/sec^2 respectively).
V : The current speed level (m/sec).
Th_0 : The desired time headway sec.

2. The second equation was introduced by TNO Institute For Policy Studies, Traffic and Transportation Unit in Netherlands (1996)[83]. Equation 10.2 has two correcting constants one for the relative distance (Kd) and the other for the relative speed (Kv). Clearly, this equation is very mathematical and has no parameters to represents the variation between vehicles (mass and drivability) as in the TRL equation.

$$a_{ACC} = Kd * (Th - Th_0) + Kv * DV \qquad \ldots..10.2$$

Where: Kd : Correction factor for the time headway.
Kv : Correction factor for the Relative speed.

The other equation needed to build the ACC logic, is that of engine braking which determines the level of deceleration achieved by the engine done. This is useful to determine if the driver is really ready to enter the cruise control operation or not. HIROSHI had reported that the engine braking could be presented by the equation (HIROSHI S. 1994)[84]:

$$DC_{Eng} = -0.0317 * V + 0.132 \qquad \ldots.10.3$$

Where: V is the current speed level ($V \geq 13.5$ m/sec). However, for a speed level less than the mentioned level the model will accept a value of (–0.4 m/sec^2).

The ACC logic is built around the above mentioned limits and equations to ensure that the ACC system acceptably applied. Figure 10.1 presents the flowchart diagram for the ACC logic subroutine. The subroutine was easy to build and the only difficulties are in matching the simulation model epoch [0.25sec] with the ACC model epoch [0.125sec], this was achieved by reducing the epoch to 0.125sec and let the model does its calculation every two epoch. The modification was done for the two simulation programs: the platoon simulator and the motorway simulator.

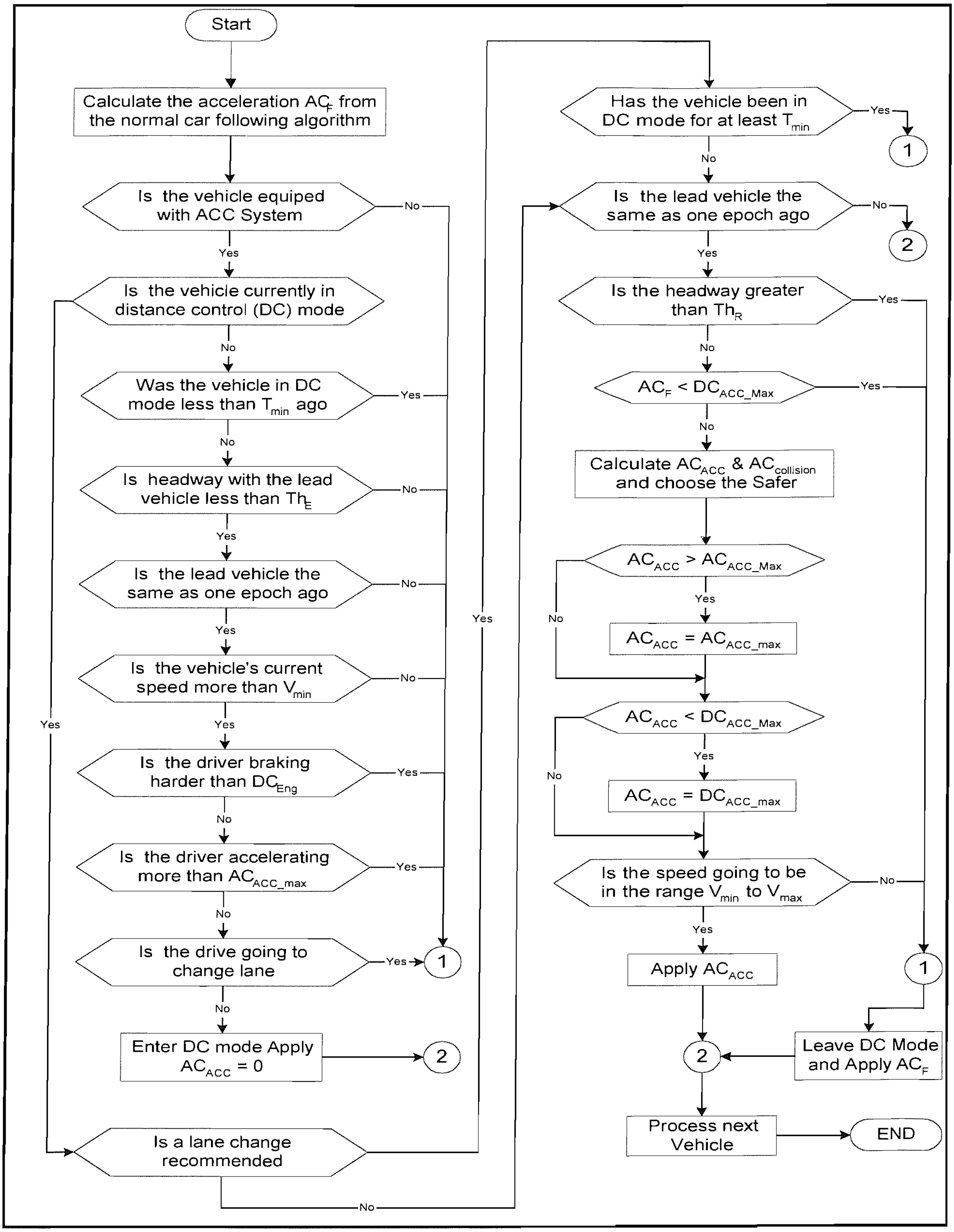


Figure 10.1: The logic of the ACC system.

10.3 THE EFFECT OF ACC ON A PLATOON OF VEHICLES:

A platoon of 15 vehicles was simulated, the lead vehicle had a constant speed for at least 15 seconds, then decelerated to a stop with a constant deceleration equal to $-1.5 m/sec^2$. Three different ratios were used for the number of ACC equipped vehicles (0%, 50% and 100%) and the acceleration, speed, relative distance and relative speed were recorded for every trial. Also, the experiment was repeated several times for each of the TRL (Equation 10.1), and the other for TNO (Equation 10.2). Figures (10.2 and 10.3) show the acceleration sequences with the different ACC ratios (TRL equation 10.1) and Figures (10.4 and 10.5) present the acceleration sequences with the different ACC ratios (TNO equation 10.2).

Discussion

- The ACC system would appear to have improved the traffic situation from a safety point of view, because it generates longer following headways and consequently reduces the levels of deceleration in the shock wave. This was found to be true for both equations (TRL & TNO), although the TNO one worked significantly better. The TNO equation responded better because of its high sensitivity to the change in the relative speed rather than the headway. (This may lead to inconvenient perception by the drivers, although this is not the subject of this research).

- In calm following processes the fluctuation in deceleration does not disappear until all the vehicles in the platoon are equipped with an ACC system. However, the test showed that in the circumstances where the percentage of ACC equipped vehicles less than 100%, that the TRL equation was better at reducing the variation in the deceleration level due to its slow response to change in the relative speed.

The initial conclusion is that: the ACC systems may be beneficial to safety. Also, the TRL equation may work better overall, due to its lower sensitivity to the relative speed. In order to finalise the conclusion another experiment was undertaken using a short constant desired headway instead of normal distribution for the ACC system (the ACC proportion was assumed to be 50%).

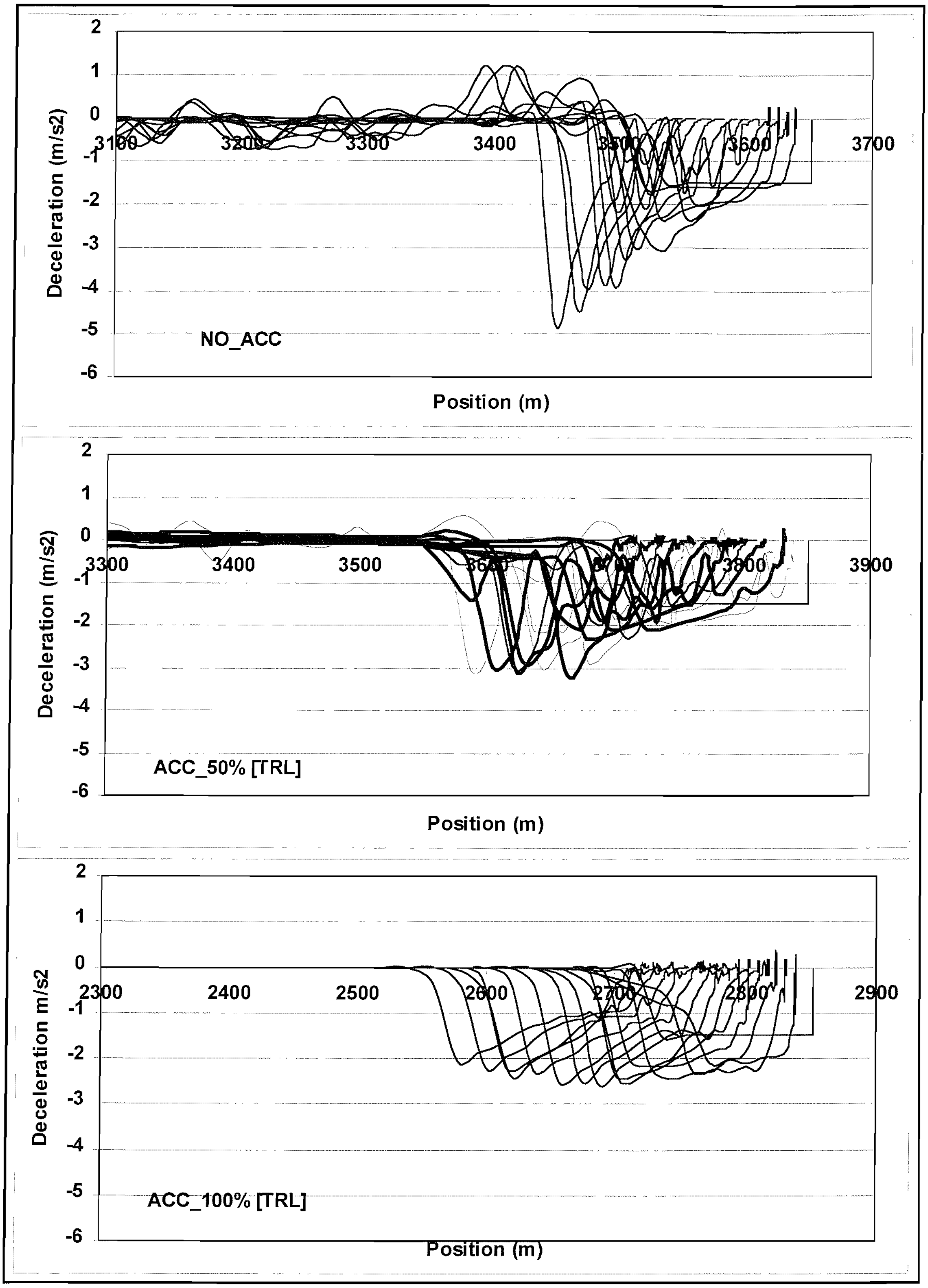


Figure 10.2: The Acceleration variation for a platoon of 15 vehicles and different percentages of ACC equipped vehicles (TRL equation).

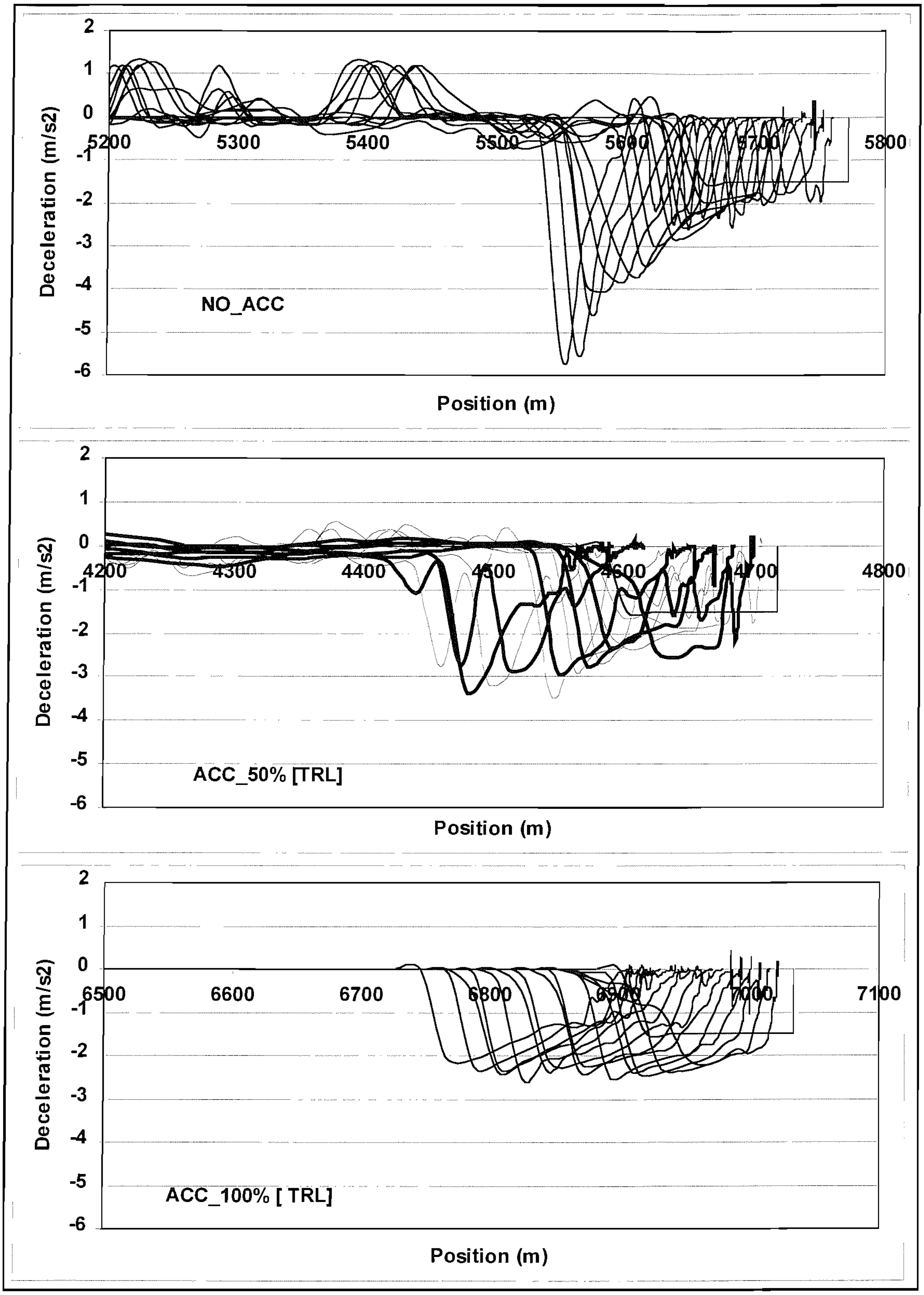


Figure 10.3: The Acceleration variation for a platoon of 15 vehicles and different percentages of ACC equipped vehicles (TRL equation) another trail.

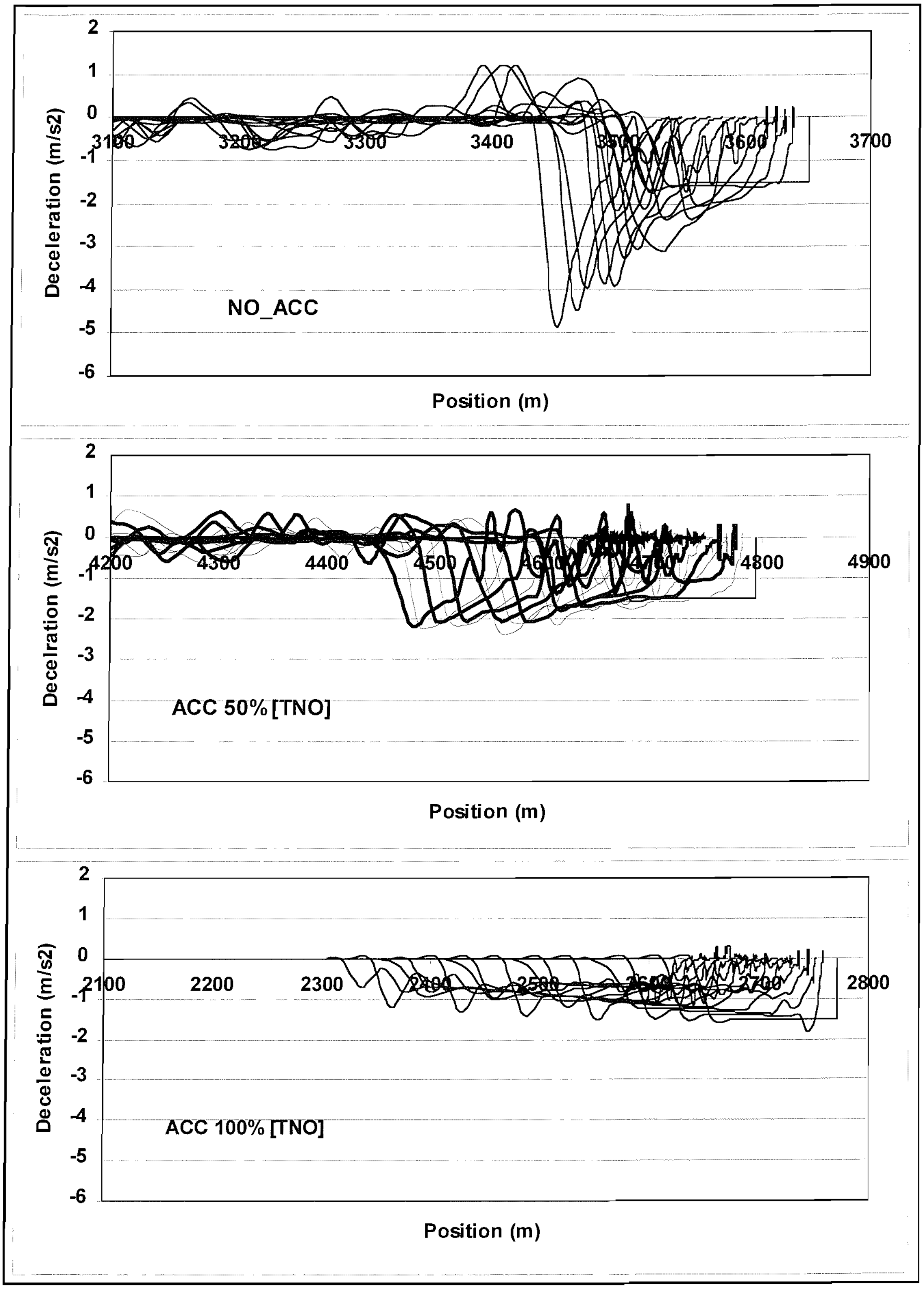


Figure 10.4: The Acceleration variation for a platoon of 15 vehicles and different percentages of ACC equipped vehicles (TNO equation).

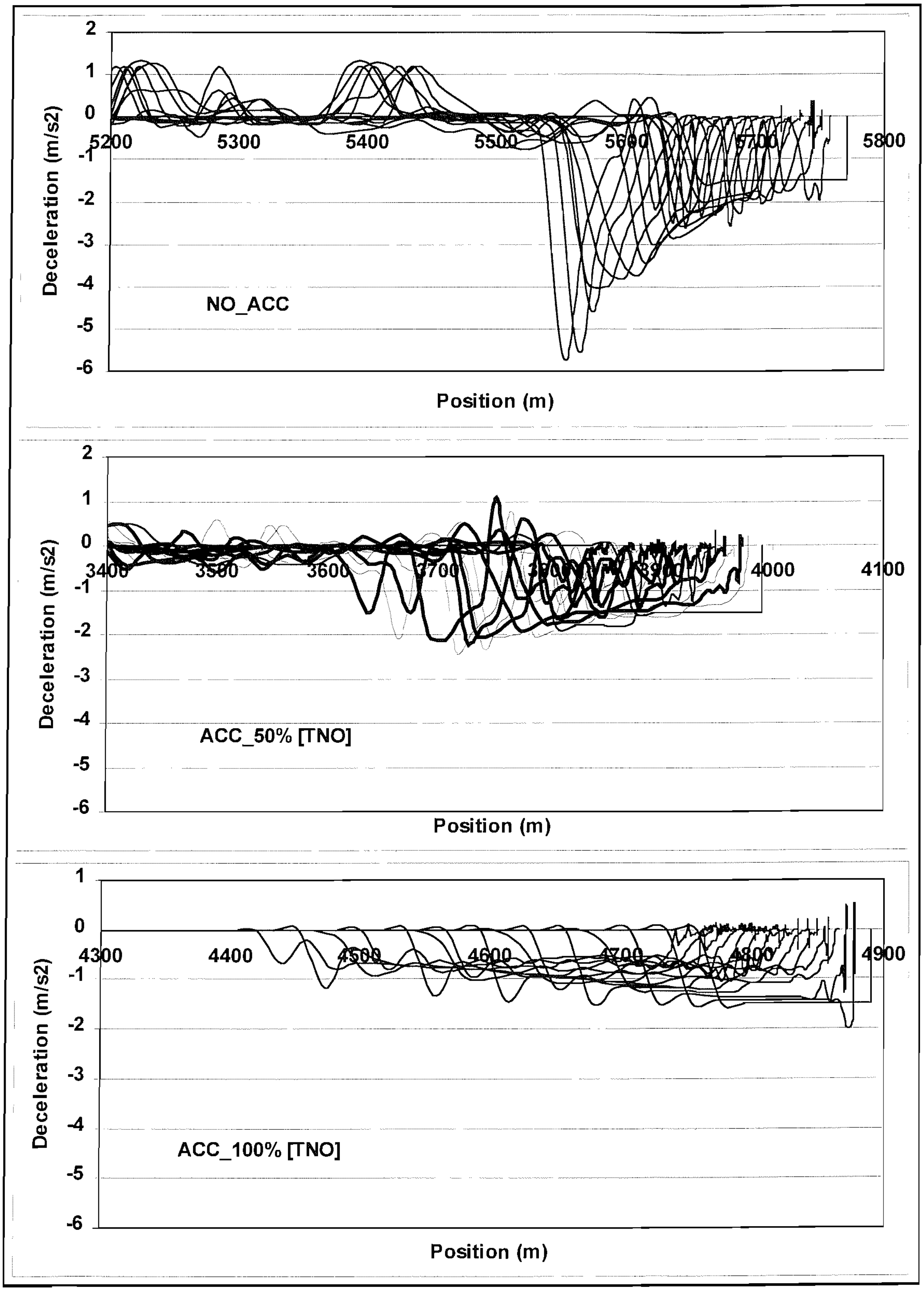


Figure 10.5: The Acceleration variation for a platoon of 15 vehicles and different percentages of ACC equipped vehicles (TNO equation) another trail.

Figure 10.6 presents the results from the TRL equation by considering a short desired headway of 1.2sec. Figure 10.7 presents the same test results for the TNO equation. For the TRL equation, the short headway was not enough to prevent the build up of a shock wave, whereas the TNO equation coped better and prevented the build up of a shock wave. In conclusion, it is clear that the TNO equation is a safer one to apply in an ACC system, although, there is uncertainty about its convenience for drivers. (This uncertainty and the way in which drivers respond may well have its own safety implications).

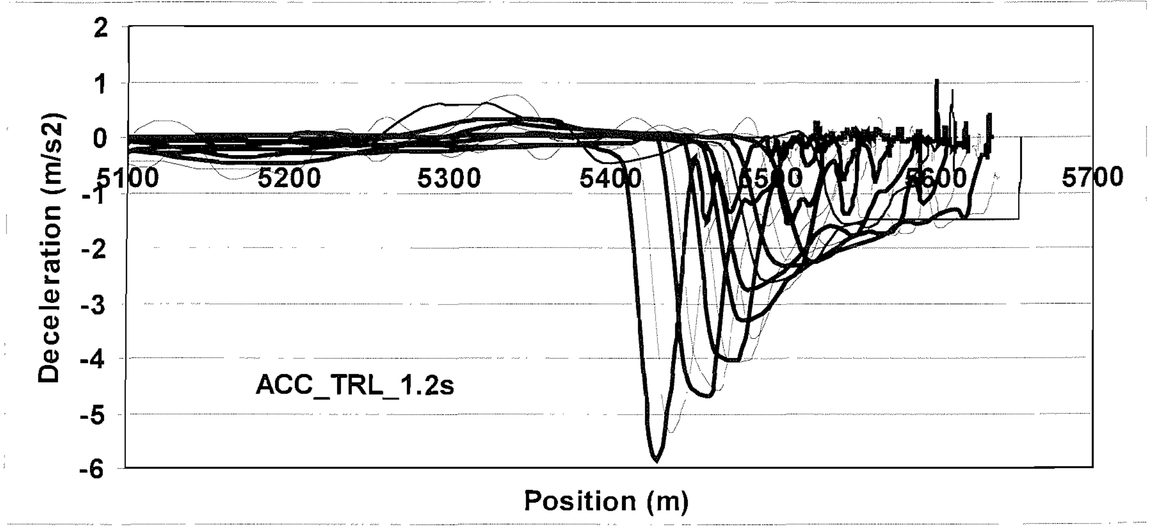


Figure 10.6: The acceleration variation in a platoon of vehicles (50% ACC percentage) with different desired headway for the ACC vehicles (TRL).

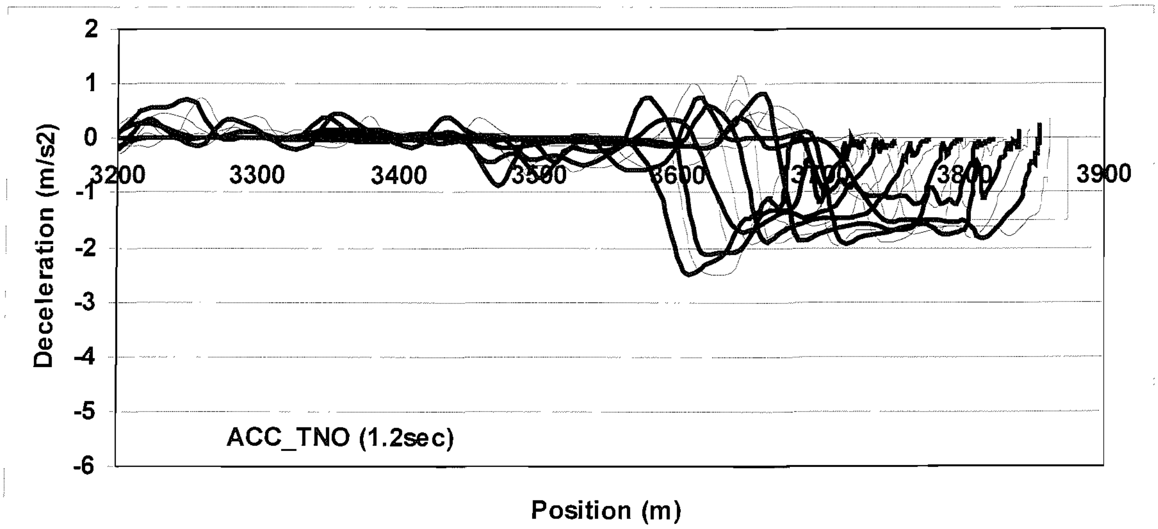


Figure 10.7: The acceleration variation in a platoon of vehicles (50% ACC percentage) with different desired headway for the ACC vehicles (TNO).

Finally, it is very important to note the effect of the other non-equipped vehicles in the platoon, as they also have influence the propagation of a shock wave depending on their following behaviour. The full benefits from an ACC system will not be gained until all vehicles are equipped. Several recent studies have shown that ACC is a highly favourable by customers and manufacturers, but there is no high expectation of safety benefits from such system in the near future (HOEDEMAEKER M. & BROOKHUIS K. A. 1998)[81].

10.4 THE EFFECT OF ACC ON THE MOTORWAY TRAFFIC FLOW:

For each of the TRL and the TNO equation, the simulation was run several times, with different set-ups for three percentages of ACC equipped vehicles (0%, 50% & 100%). However, as demand levels of between 2000 to 6000 veh/h were needed to be covered, the generated demand was set-up to start at 2000 vehicle/hour and rise every 10 minute by 100 veh/h until it reached 6000 veh/h. The measurement section was one kilometre started at 4.5 kilometre from the beginning of the simulated section. The data was saved in a text file and then analysed by Excel spreadsheet. Key results were change in speed and the number of lane-changing events. The ACC system was applied only to the passenger vehicles, and the HGV sustained a fixed percentage of 10%.

10.4.1 The TRL Equation:

The results showed that the ACC system did not affect the speed average over the low and medium demand level (less than 4500 vehicle/hour). However, at high demand levels, the average speed fell with the increase in the ACC proportion, and the traffic was less able to recover after the shock wave. On the other hand, the flow level became more stable, with less fluctuation when the ACC proportion reached 100%. This means that this ACC system (TRL equation) will enable higher capacity on motorway at lower average speed. The interpretation of this is because of the slow response by the equation to change in the relative speed. Figure 10.8 presents the average speed and flow level over the simulation time for this experiment.

A further test was to look at the number of lane changing events per hour per km and how it changed with flow for different proportions of ACC equipped vehicles. The results showed that the number of these events fell substantially as the proportion of ACC equipped vehicles increased. This result was expected, because of the minimum duration time for applying the

ACC system in the model. Figure 10.9 present the number of lane changing events per hour per km over the flow level for this experiment.

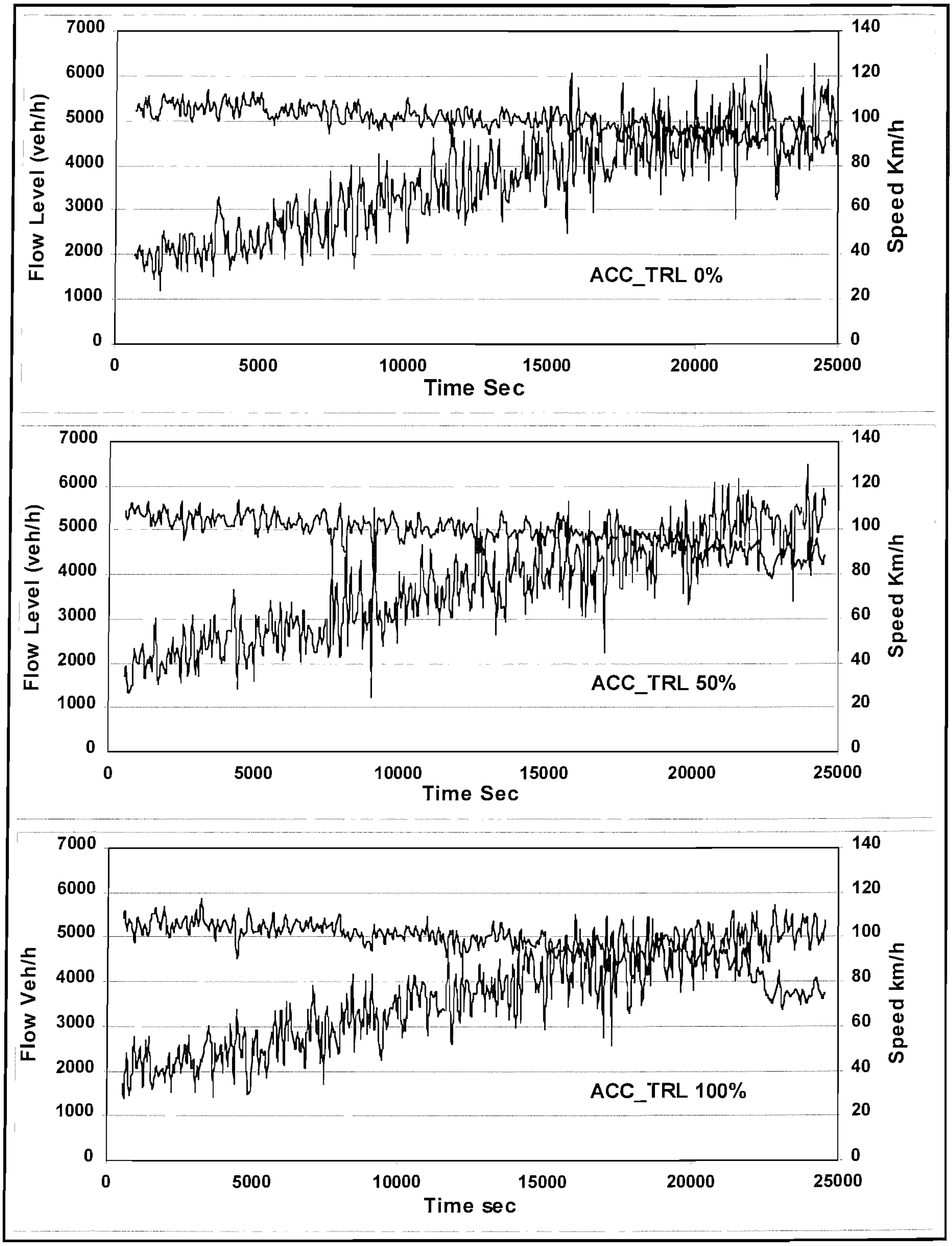


Figure 10.8: The average speed and flow from three simulation runs with different proportions of ACC equipped vehicles (TRL equation).

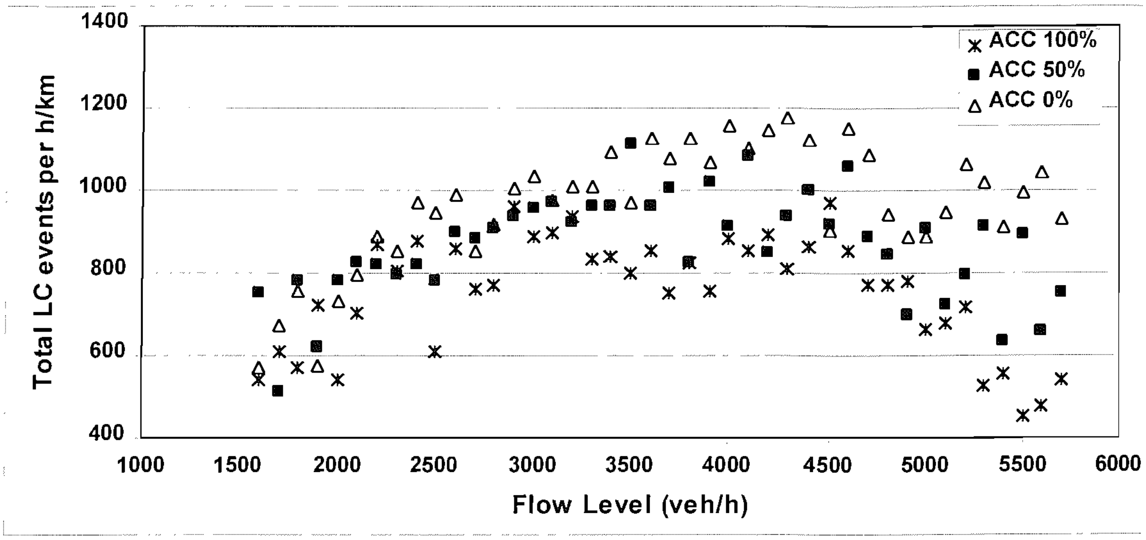


Figure 10.9: The Number of LC events per h/km with the flow (TRL equation).

10.4.2 The TNO Equation:

The TNO equation produced apparently better results for the average speed, and it may found that the ACC algorithm was able to sustain a higher average speed comparing to the non-ACC situation. However, this system failed to dampen the fluctuation in flow level, which means more shock waves and less capacity than the TRL equation. This result is predicted, as the equation is very sensitive to the relative speed of the lead vehicle. Nevertheless, this result questions the safety of this system in real traffic situations. That contradicts the conclusion in paragraph 10.3 above. Figure 10.10 presents the average speed and flow level over the simulation time for this experiment. Finally, the number of lane-changing events per hour per km has also decreased with the increase of the ACC percentage, although, the level of reduction was less than that found with the TRL equation. Figure 10.11 presents the variation in the number of lane changing events per h/km with flow.

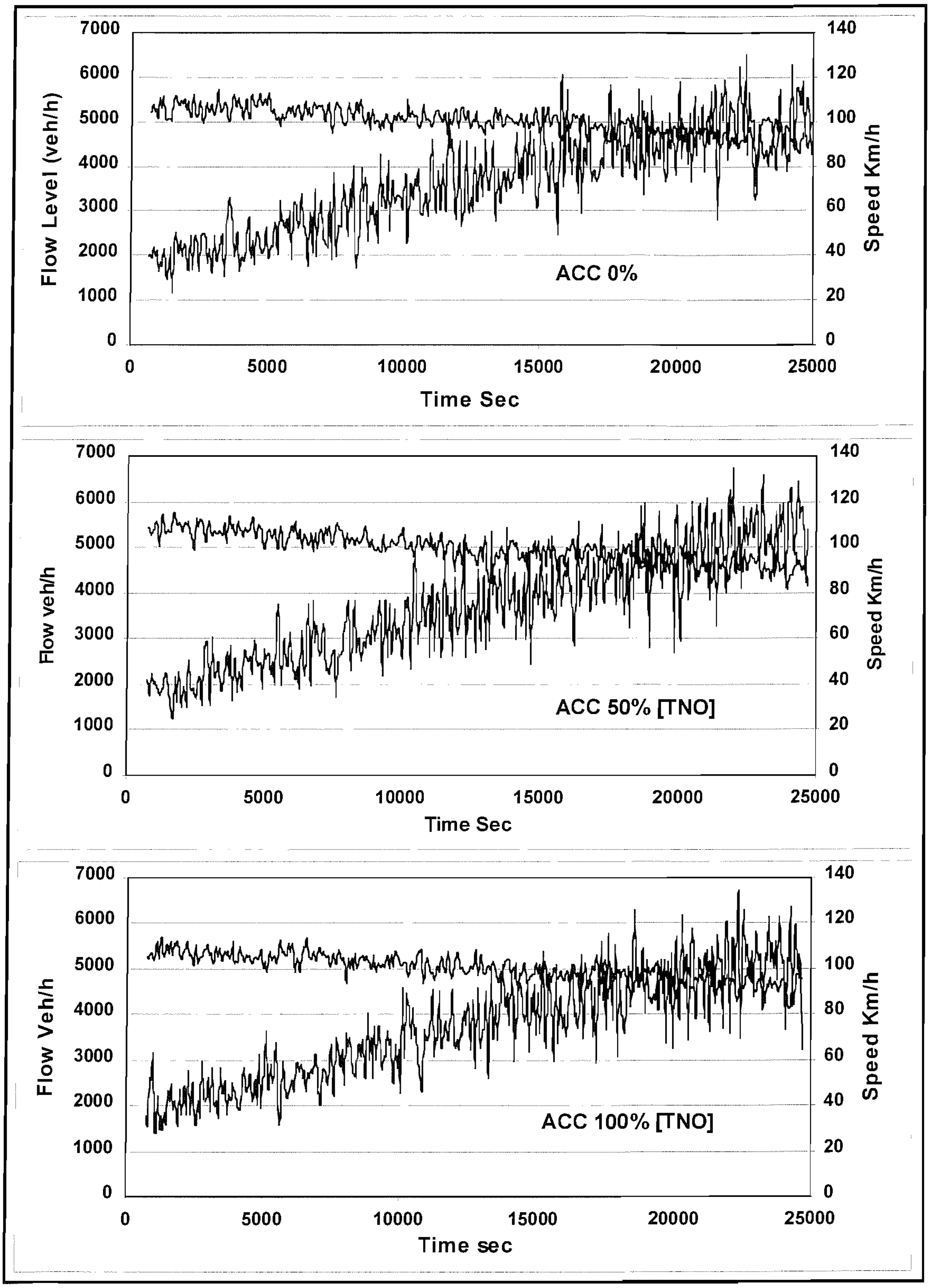


Figure 10.10: The average speed and flow from three simulation runs with different proportions of ACC equipped vehicles (TNO equation)

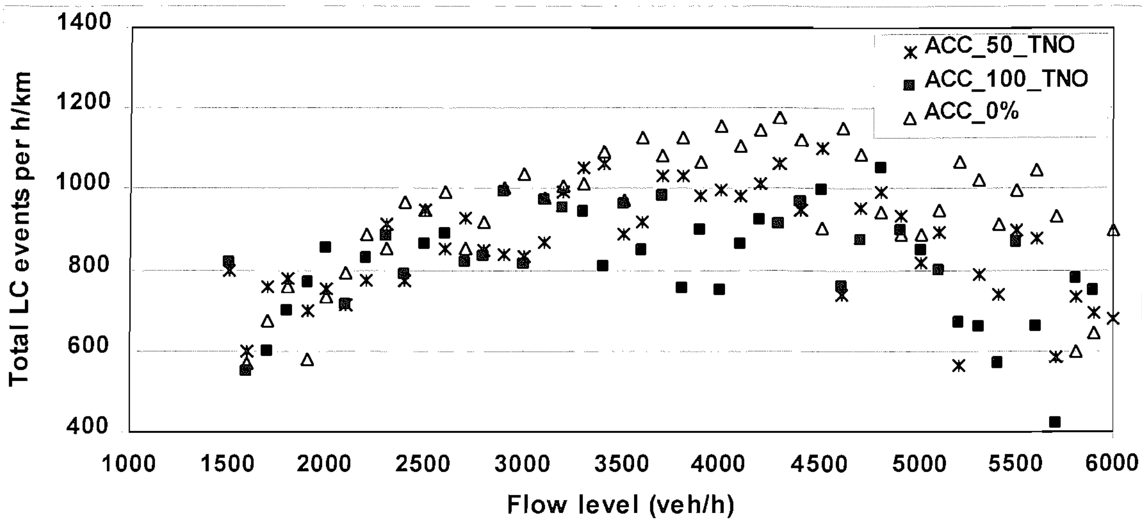


Figure 10.11: The Number of LC events per h/km with flow (TNO equation).

10.4.3 Conclusion:

Car manufacturers see ACC systems as having considerable marketing benefits. At the same time, the implication will effect traffic operations more generally. Nevertheless, the tests and experiments in this chapter have showed that different ACC systems will have different impacts in both safety and capacity. Also, most benefits seem to be achieved when all vehicles are fitted.

CHAPTER 11

CONCLUSIONS

11.1 INTODUCTION:

This research has been concerned with the modelling of driver behaviour mainly in car following and lane changing situations. A simulation model has been developed that describes observed behaviours as close by as possible. The TRG instrumented vehicle was used to collect the car-following's data. The instrumented vehicle is equipped with highly sophisticated tools that enable essential microscopic data for the calibration of simulation model to be conducted. Special programs were developed to filter and smooth the surveyed data. Although the model itself was based on the action point model concept, many changes were made to better reflect observed behaviour. Finally, the calibrated and validated simulation model was used to assess the impacts of an Automatic Cruise Control System on a platoon of vehicles and traffic on motorways. This Chapter presents the main findings and conclusions from this research and identifies recommendations for the future work.

11.2 THE MAIN FINDING AND CONCLUSIONS:

1. The results from the free speed distribution survey on motorway have supported the BURROWS theory (BURROWS 1974)[47]. It was found that the normal free speed distribution of any lane could be derived from the normal free speed distribution of the whole traffic stream for the three lane motorway for the same type of vehicle. Assuming $N(\mu , \sigma)$ is the free speed normal distribution for the whole traffic, while, μ : is the mean and σ : is the variance. The distribution parameters for the three lanes would be:

Lane	Mean	Variance
1	$\mu - \sigma$	$2*\sigma/3$
2	μ	$2*\sigma/3$
3	$\mu + \sigma$	$2*\sigma/3$

Table 11.1: The free speed normal distribution on three lanes motorway

2. In the perception threshold survey, the Optic Flow level was found not to be steady over the change in the relative distance. It increases with a decrease in the relative distance. Therefore it is difficult to use the optic flow to define the drivers perception thresholds (closing, opening or constant).

3. The relative speed / relative distance percentage (DV/DX) was found to be more consistent in representing the perception threshold. There is a small but distinct possibility that a driver may reach a wrong perception decision in which, for example he or she believes that the inter-vehicle distance is increasing, whereas in reality the opposite is happening. Nevertheless, it is very difficult to apply such phenomenon of miss-perception in a deterministic model.

4. The data analysis has shown that the Time-To-Collision TTC is the best parameter to represents the starting action points for the approach process. It was found that the TTC value has a normal distribution (mean = 11.71 sec and Std = 2.29 sec) at the action points.

5. It was found that Drivers are more likely use the TTC to control their approach processes, where they were found to accept a minimum value with lognormal distribution (mean = 8.41 sec and Std = 2.01 sec). They then sustained until they achieved their own following distance.

6. The analysis of the close following measurements showed that the power relationship with the speed level was the best way to model the limits of ABX (the minimum acceptable following distance). Also, the same result was found in modelling SDX (the maximum acceptable following distance).

7. The data from the close following experiments showed that drivers have two phases in distance keeping behaviour:

    - Firstly an increasing following distance with speed, giving way to a constant value at around 65 to 75 kph. Such a trend is perhaps unsurprising as it would seem intuitively obvious that the faster a vehicle travels, the more space a driver will allow to account for stopping distances.

- Secondly, above 105 kph, the time headway decreases with an increase in speed., perhaps reflecting the onset of a more aggressive type of behaviour.

8. The relative speed analysis has different characteristic, the CLDV and OPDV action points, the relative speed level to instigate deceleration and acceleration respectively, were scattered and no obvious equation was determined. Therefore, the cumulative percentage was suggested to present the two thresholds CLDV and OPDV. Also, the analysis gave no relation between the speed level and the relative speed thresholds CLDV and OPDV.

9. The data analysis proved that drivers cannot sustain a zero acceleration, and the noise of deceleration/acceleration in constant speed mode has a normal distribution of [m=0.0 , Std=0.07 m/s^2].

10. Investigation of the drivers' sensitivity to change in the speed level revealed that the drivers' perception of a constant speed is within the limits ±1.2 m/sec. However, the speed deviation had a normal distribution [mean = 0.0 and Std = 0.57 m/sec].

11. In 'calm' close following, the deceleration/acceleration level was found to be in the range of ±0.45 m/sec^2 with the trend to increase with an increase in the relative speed. However, its distribution was uniform between ±0.14 and ±0.45 m/sec^2.

12. Investigating the influence of the more than vehicle ahead proved to be extremely difficult using the conventional data collecting methods, (i.e. video camera, detectors... etc.). A platoon of several instrumented vehicles could open the way to this investigation to be carried out.

13. The lane-changing model successfully used the relative speed and distance with the surrounding vehicle to determine the thresholds and boundaries under which drivers usually perform their lane-changing manoeuvres.

14. The lane utilisation as an output from the model was found to be sensitive for some factors in the lane-changing model such as: the lane hogging percentage, the TTC with the vehicle to the left and the speed tolerance margin.

15. The number of lane-changing events in the model was found to be influenced by the lane hogging percentage, the distribution factor of the threshold to the right and minimum manoeuvre time.

16. In assessing the safety aspect of an ACC system it was found that at short desired headway (1.2sec) the TRL equation was unable to prevent the propagation in the deceleration level in platoon of vehicle. However, the TNO equation proved to be better than TRL in coping with such situation.

17. The slow response from the TRL equation to the relative speed has proved to be useful in reducing the fluctuation in the flow level in the simulation. On the other hand, it suffered from a slow recovery in the speed of flow when the level of demand exceeded the capacity level.

18. In contrast to the TRL equation, the TNO equation worked well in recovering the speed level, though the fluctuation in flow level was very high.

19. In order to achieve the full benefit from an ACC system it has to be applied to all vehicles. Thus in the upgrading process from no ACC to full ACC, benefits may not be clear and there may well be disbenefits. The interaction of many different system in the market may also be significant.

11.3 SOME RECOMMENDATION FOR FUTURE WORK:

The recommendation for future work can be divided into three levels:

11.3.1 The Analysis Level:

1- The data in this research were collected in the UK. The whole research can be repeated in other country to find out whether the drivers' behaviour is going to change or not.

2- It is recommended that more subjects be used in the approach process experiment in order to confirm the results and findings in this research.

3- Using more than one instrumented vehicle will enable the exploration of the influence of more than one vehicle ahead on driver behaviour. Also, it will enable researchers to study the behaviour of a platoon of vehicles in shock wave situations more accurately.

4- It is recommended that fresh data be collected to analyse lane-changing behaviour. (This is supported by the argument that drivers may change behaviour over the long time).

5- During the development of the lane-changing model, many indications suggested that the model output could be affected by the way the model's logic is introduced. Therefore, another research should be carried out to show the effect of any modification to the lane changing logic on the model output.

6- Finally, a study of driver behaviour in the slip roads, and how that will affect drivers in the main stream is important extension of the activity as few motorway sections are unaffected by the adjacent enter changes.

11.3.2 The Simulation Model Level:

1- Expanding the simulation to accept slip roads and junctions in order to make it more realistic.

2- Expanding the model to have a wider variety of vehicle types.

3- Making the model more users' friendly in operating an on line help to guide the new users.

4- Adding some facilities to collect data from specific simulated vehicle.

11.3.3 The Application Level:

1- Using the simulation model to studying the effect of speed control in improving traffic on motorways.

2- Using the simulation model to study ramp metering and how to make it effective and useful for certain situations.

3- Using the simulation model to study the introduction of certain type of drivers' behaviour in car following and lane changing.

REFERENCES

1. GREENSHIELDS-BD (1938): Average Speed And Traffic Density As A Measure Of Road Capacity, Highway Research Board Proceedings. 1938.

2. ADOLF D. MAY (1994): Traffic Management from Theory to Practice, Transportation Research Record 1457, pp 5-14.

3. HIGHWAY CAPACITY MANUAL (1965): Transportation research Board, National research Council. Washington, D. C.

4. HIGHWAY CAPACITY MANUAL (1985): Transportation research Board, National research Council. Washington, D. C.

5. HIGHWAY CAPACITY MANUAL (1994): Transportation research Board, National research Council. Washington, D. C.

6. HALL F. L. & BANKS J. H.(1993): Synthesis of recent work on the nature of speed-flow and flow-occupancy relationships on Freeway, Transportation Research Record 1365, pp 12-17.

7. HALL F. L & SMITH W. S. & MONTGOMERY F. (1994): Comparison of Speed-Flow Relationships based on data from North America, The U.K and Germany. Proceeding of the Second International Symposium on Highway Capacity 1994, Vol. 1 pp. 243-264.

8. JOHN RINGERT AND THOMAS URBANIK (1993): Study of Freeway Bottlenecks in Texas. Transportation Research Record 1398. 31 - 41.

9. HOUNSELL N. B. & MCDONALD M. (1994): An investigation of flow breakdown and merge capacity on motorways, Second International Symposium on Highway Capacity. pp 279-290.

10. MARKOS PAPAGEORGIOU (1995): Concise Encyclopaedia of Traffic & Transportation System

11. MORIN J M (1985): SIMAUT Un programme de simulation du traffic autoroutier. INRETS Report No. 4 INRETS, Arcueil, France.

12. LYONS GLENN (1995): Calibration and Validation of a Neural-Network Driver Decision Model. *Traffic Engineering and Control.* (January 1995), pp 10-15.

13. SHINYA KIKUCHI and PARTHA CHAKROBORTY (1993): Car-Following Model Based on Fuzzy inference System. *Transportation Research Record 1365*, pp 82-91.

14. CHANDLER, R. E., HERMAN, R. AND MONOTROLL, E. W. (1958): Traffic Dynamics: Studies in Car Following. Operation Research, 6, pp 165-184.

15. HERMAN, R., MONOTROLL, E. W. POTTS, R. B. AND ROTHERY, R. W. (1959): Traffic Dynamics: Analysis of Stability in Car Following, Operation Research, 7, pp 86-106.

16. BRACKSTONE M. (1996): Fuzzy Logic Motorway Model , Literature Review, Special Report.

17. GIPPS, P. G. (1981): A behavioural Car-Following Model for computer Simulation. Transportation Research, Vol. 15B, pp 105-111.

18. MICHAELS, R. M. (1963): Perceptual factors in car Following. Proc. of the second International Symp. on the Theory of Road traffic Flow. OECD, 1965 .pp 73-83.

19. TODOSIEV, E.P. (1963): The action-point model of driver vehicle system, the Ohio State University, Report 202 A-3.

20. HOFFMANN, E.R.(1966): Note on detection of vehicle velocity changes. Human Factors 8 . pp 139-141.

21. LEUTZBACH W. AND WIEDEMANN R. (1986): Development and applications of Traffic simulation models at the Karlsruhe Institut fur Verkehrwesen, Traffic Engineering + Control , May, pp 270-278

22. SPARMANN, U (1979): The Importance of lane changing on motorways. Traffic Engineering and Control, June 1979, pp 320-323

23. CHANG, G. L. AND KAO, Y. M. (1991): An Empirical Investigation of Macroscopic Lane changing Characteristic on Uncongested Multilane Freeways. Transportation researche A, 25A, 6, pp 375-389.

24. WORRALL R. D., AND BULLEN A. G. R., (1970) : An Empirical Analysis of Lane changing on Multi lane Highways. Highway research record, No 303, pp 30-43.

25. PAHL J (1972): Gap Acceptance Characteristics in Freeway Traffic Flow. Highway research record, No 409, pp 57-63.

26. MCDONALD M., & BRACKSTONE, M. (1994): Modelling Motorway Operations. Transportation Research Record 1485, pp 56-60.

27. GIPPS, P.G. (1986): A Model For The Structure Of Lane Changing Decision. Transportation Research Board, 20B, 5, pp 403-414.

28. WIEDEMANN, R. & REITER, U. (1992): Microscopic Traffic Simulation: The Simulation System MISSION. Background and Actual State. CEC Project ICARUS (V1052) Final Report, Vol.2 Appendix A. CEC, Brussels.

29. WOOTTON JEFFREYS (1990): SISTM - Microscopic Simulation of Traffic on Motorways, Traffic Engineering Background. Transportaion and Road Research Laboratory.

30. SAAD YOUSIF (1995): Modelling lane Utilisation on British dual-carriageway roads: effects on lane-changing. Traffic Engineering and control.

31. VAN AS S. C. (1979): Traffic signal optimisation–procedures and techniques. Ph.D. dissertation. The University Of Southampton.

32. CHIN, H. C. (1983): A computer Simulation Model Of Traffic Operation at Roundabout. Ph.D. Thesis, Dept. of Civil Eng. University Of Southampton.

33. ADAMS W. F. (1936): Road Traffic Considered as random series, Journal of the Institute of Civil Eng. Vol 4.

34. ASHTON W. D. (1966): A Theory of Road Traffic Flow. Methuenand Co.

35. SKABARDONIS, A (1982): A Model of Traffic Flow at Ramp Entries. Ph.D. Thesis. University Of Southampton.

36. TOLLE J. E. (1976): Vehicular Headway Distributions: Testing and results. Transportation Research Board. Record 567. pp 56 - 64.

37. CLEVELAND DE & CAPELLE DG (1964): Queuing Theory Approaches, Highway Research Board special report 79.

38. BUCKLEY D. J. (1968): A Semi-Poisson Model of Traffic Flow. Transportation Science. Vol. 2 No2 , pp 107-133.

39. SCHULL, A. (1955): The Probability Theory Applied to Distribution of Vehicles on Two Lane Highway. ENO. Connecticut.

40. DAWSON RF AND CHIMINI LA (1968): The Hyperlang Probability Distribution - A generalised traffic headway model, Highway Research Board, HRR 230, pp 45-59.

41. MILLER, A.J. (1961): A Queuing Model For Road Traffic Flow. Journal Of The Royal Statistical Society. Series B, Vol 23, No 1, pp 64 - 75.

42. BRANSTON, D. (1976): Models of Single Lane Time headway Distributions. Transportation Science. Vol. 10, No 2, pp 125 - 148 .

43. WASIELEWSKI J. (1979): Car-Following Headways on freeways interpreted by by the semi-Poisson headway distribution model. Transportation Science Vol.13(1), pp 36-55, February 1979.

44. NAYLOR, T. H. (1966): Computer Simulation Techniques. John Wiley.

45. BOX, C.E.P. and MILLER, ME (1958): Anote on the generation of random normal deviates. Ann. Maths. Stats. , vol.9 , pp 610-611.

46. BRANSTON, D. (1979): A Method of Estimating The Free Speed Distribution For a Road. Transportation Science. Vol. 13, No. 2, May 1079.

47. BURROW, I.J. (1974): Speed of cars on motorways in England, Transport and Road Research Laboratory, LR 663, 1974.

48. GERBER N. J AND HOEL L. A. (1988): Traffic and Highway Engineering. 959p (West Publishing Co. St. Paul, MN, USA)

49. JOHANSSON, G. AND RUMER, K. (1971): Driver's Brake Reaction Times. Human Factors. Vol. 10 , No 1 p 23.

50. BOFF, K. R. AND LINCOLN, J. E. (1988): Engineering Data Compendium : Human Perception and Performance. Wright-Patterson, AFB, OH.

51. WOODWORTH, R. S. (1938): Experimental Psychology. New York : Holt, Rinehart & Winston.

52. FITTS, P. M. AND PETERSON, J. R. (1963): Cognitive aspects of information Processing : II. Adjustments to stimulus redundancy. Journal Of Experimental Psychology, 65 pp 423- 432.

53. POSNER, M. I. AND KLEIN, R. (1973): Components of Attention. Psychological Review. 391 - 408.

54. LEE D. N. (1976): A Theory Of Visual Control Of Braking Based On Information About Time To Collision. Perception Vol. 5 , pp 459-459.

55. MCLEOD R. W., ROSS H. E. (1983): Optic Flow And Cognitive Factors In Time To Collision Estimates. Perception Vol.12 pp. 417-423

56. CAVALLO V., LAURENT M.L (1988): Visual Information And Skill Level In Time To Collision Estimation. Perception vol.17 pp. 623-632.

57. CAVALLO V., GROEGER. J. A. (1991): Judgement Of Time-To- Collision And Time To Coincidence. Vision in Vehicles - III. pp 27-34

58. HOFFMANN E. R. AND MORTIMER. R. G. (1994): Drivers' Estimation Of Time To Collision. Accid. Anal. and Prev. Vol. 26, No 4, pp 511-520.

59. SIDAWAY B. AND FAIRWEATHER M. (1996): Time To Collision Estimation in a Simulated Driving Task. Human Factors 38(1), pp101-113.

60. SPURR, R.T. (1969): Subjective aspects of braking. Automobile engineer, Feb. 1969. pp 58-61.

61. LEONARDO EVANS and RICHARD ROTHERY (1973): Experimental Measurement of Perceptual Thresholds in Car Following , Highway Research Record, 464. Pp 13-23.

62. LEONARDO EVANS and RICHARD ROTHERY (1977): Perceptual Thresholds In Car Following A Comparison Of Recent Measurements With Earlier Results, Transportation Science, Vol. 11 No. 1 February 1977.

63. MONTROLL, E. (1962): Acceleration and Clustering Tendency in Vehicular Traffic. Proceeding of the Symposium on the theory of Traffic Flow. Research Laboratories, General Motors Corporation. 1962. Elsevier. Pp 147- 157 .

64. HOFFMANN E. R. AND MORTIMER. R. G. (1996): Scaling of relative Velocity Between Vehicles. Accident Anal. and Prev. 28(4), pp 415-421.

65. REITER U. (1994): Empirical Studies as Basis for Traffic Flow Models. Proc. of The second International Symp. on Highway Capacity. Vol. 2, pp 493-502.

66. WINSUM, W. and HEINO, A. (1996): Choice of Time Headway In Car Following and The Role Of Time To Collision Information in braking. Ergonomics, 39(4), PP 579-592.

67. GORDON, D. A. (1971): The Driver in Single Lane Traffic, Highway Research Record, NO. 349, pp31-41.

68. BRAUNSTEIN M. L. AND LAUGHERTY K. R. (1964): Detection of Vehicle Velocity Changes During Expressway Driving , Human Factors 6, pp 327-331.

69. LOW, D. AND ADDISON, P. (1995): Chaos in a Car Following Model Including a Desired Inter Vehicle Separation. Proceeding of the 28th ISATA Conference, Advanced Transportation Systems Symposium, Stuttgart, Germany, September 1995. pp 539-46.

70. HMSO (1993): The Highway Code. HMSO, London, U.K.

71. XING, J. (1995): A Parameter Identification of Car Following Model. Proc. Of the 2nd World Congress on ITS, Vol. 4, Yokohama, Japan, Nov. 1995 pp 1739-1745.

72. HOGEMA, J.H. (1996): Modelling of Individual Car Following Behaviour. Proc. of the XV European Annual Conf. on Human Decision Making and Manual Control. TNO, Netherlands, June, 1996.

73. HUDDART, K. W. AND LAFONT, R. (1990): Close Driving - Hazard or Necessity. Proc. of the 18th PTRC Meeting. Seminar G - P334. pp 205-17.

74. PARKER, M.T. (1996): The Effect of Heavy Good Vehicles and Following Behaviour on Capacity at Motorway Roadwork Sites. Traffic Engineering and Control. Sept., pp 524-31.

75. COLBOURN, C.J. , BROWN, I.D. AND COPEMAN, A.K. (1978): Drivers' Judgements of Safe Distances in Vehicle Following. Human Factors, 20(1), pp 1-11.

76. MONOTROLL, E. W (1958): Acceleration Noise and Clustering Tendencey of Vehicular Traffic Operation Research , 4, pp 147-157.

77. MCDONALD M., BRACKSTONE, M. AND JEFFERY D. (1994): Simulation of lane Usage Characteristics on 3 Lane Motorways. Proc. 27th ISATA Conference, ATT/IVHS Symposium, Aachen, Germany, 1994.

78. FERRARI P. (1989): The Effect of driver Behaviour on Motorway Reliability. Transportation Research Board. Vol. 23B, No2 pp 139-150.

79. SALTER R. J. & HOUNSELL N. B. (1989): Highway Traffic Analysis and Design. MACMILAN PRESS Ltd.

80. NILSSON LENA (1991): Safety Effects of Adaptive Cruise Controls In Critical Traffic Situations, Second world congress (1991). Vol(3) pp 1254-1259.

81. HOEDEMARKER M. & BROOKHUIS K. A. (1998): Driving With an Adaptive Cruise Control (ACC), To be published in Transportation Research, Part F.

82. TRL Special Report (1997): Upgrade of SISTM to Version 5 ACC Systems Requirements Specification, Draft 1A., SRN049/N409/SRS5.

83. AREM B. V. & HOGEMA J. H. & VERHUEUL C. H. (1996): An assessment of the impact of Autonomous Intelligent Cruise Control, TNO Institute For Policy Studies.

84. HIROSHI SATOH & IKUHIRO TANIGUCHI (1994): Anew Control Method of Adaptive Cruise Control, Proceeding of the First Wold Congress on Application of Transport Telematics and Intelligent vehicle-Highway Systems (1994), pp 2094-2101.

85. BRANSTON D. (1977): Some Factors Affecting the Capacity of a Motorway. Traffic Engineering and Control, June 1977.

86. Hoque, M.S. (1994). The Modelling of Signalised Intersections in developing countries. Ph.D. dissertation. The University Of Southampton.

APPENDIX (1)

THE FILTER PROOGRAM CODE & ITERFACE

This program was written in Visual Basic programming language. It is consist of two main parts interface Forms and programming code.

1- **Interface Forms:** There are two main forms :

- Program Main Form: this form contain all the control buttons, the massage window, all the counters, and finally the source and destination directories Figure a1.1 shows this from.
- The Set-up From: This from contain all the set-up parameter that can be changed in the filter program. Figure a1.2 shows this from

2- **The programming Code:** This is written in Visual basic which uses a subroutine for every object in the interface forms. However the code is listed at the end of the Appendix (1).

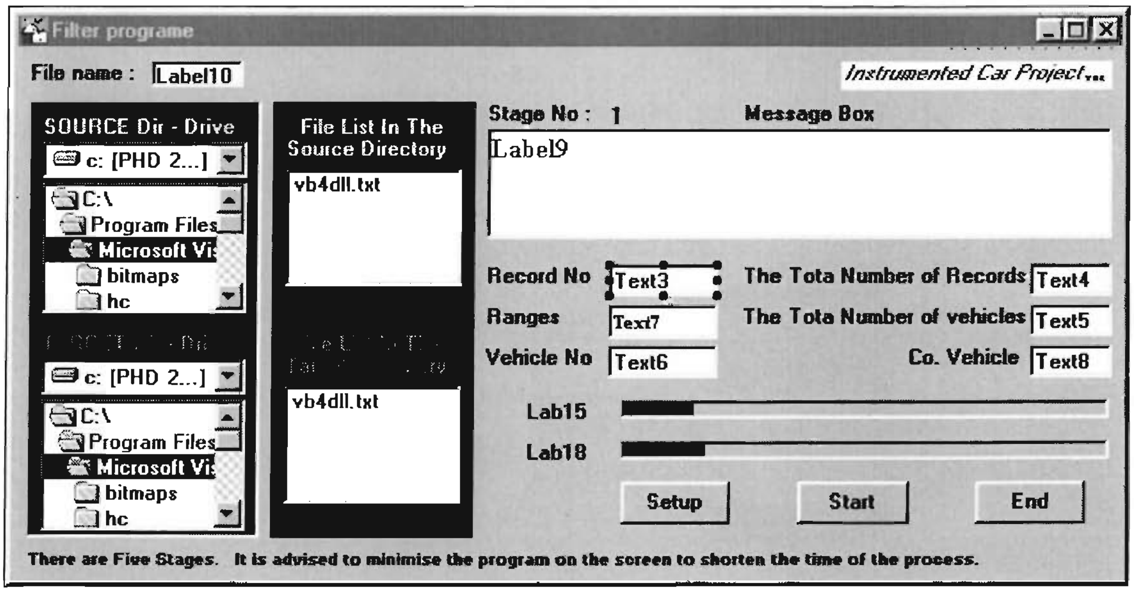


Figure a1.1: The main from in the filter program

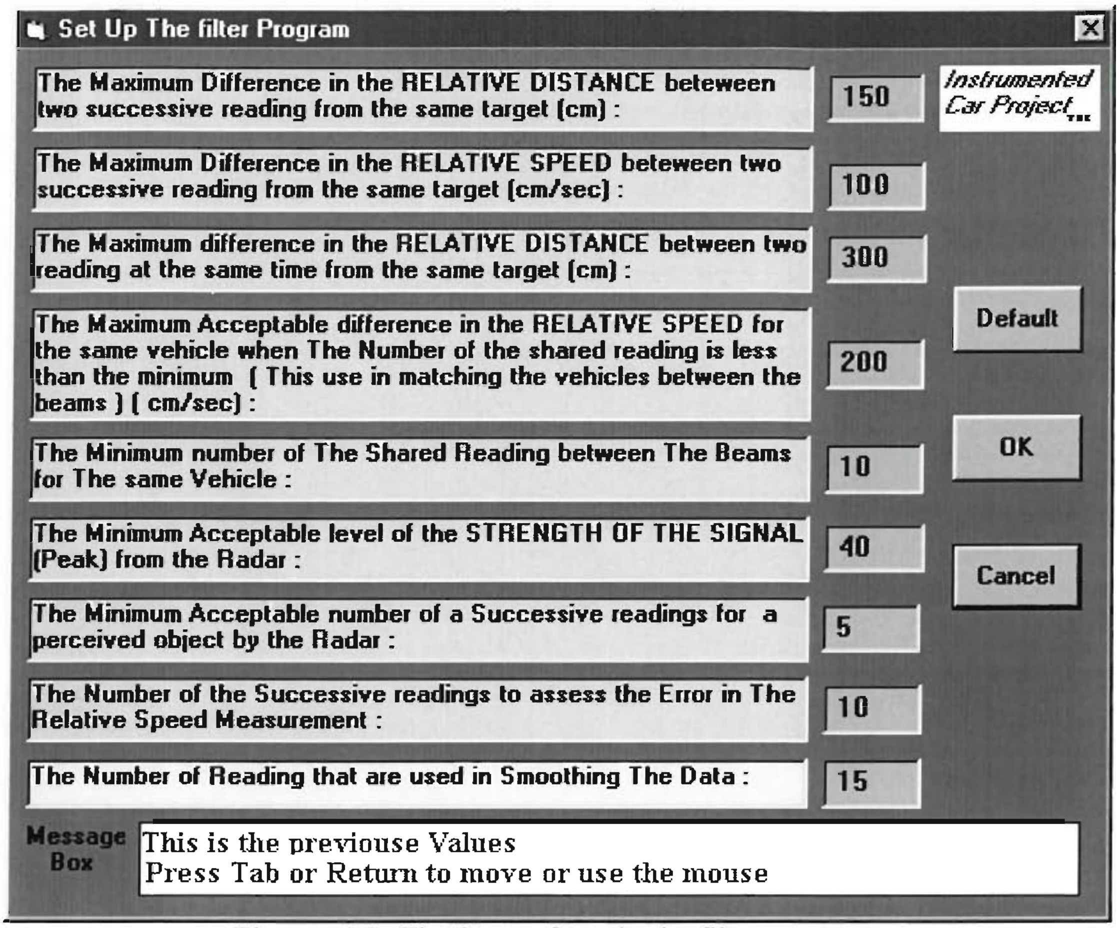


Figure a1.2: The Set up from in the filter program

Code list:

```
Attribute VB_Name = "Module1"
 Public Type datarec
  FrameNo      As Long
  elapse       As Single
  PCTime       As String * 8
  radartime    As Integer
  UnCalSpeed   As Single
  LongAcc      As Single
  LatAcc       As Single
  AccTime      As Single
  Event        As Integer
  range12      As Single
  velocity12   As Single
  SelBeam      As Single
  SelTrack     As Single
  range0       As Integer
  velocity0    As Integer
  peak0        As Integer
  range1       As Integer
  velocity1    As Integer
  peak1        As Integer
  range2       As Integer
  velocity2    As Integer
  peak2        As Integer
  range3       As Integer
  velocity3    As Integer
  peak3        As Integer
  range4       As Integer
```

```
    velocity4   As Integer
    peak4       As Integer
    range5      As Integer
    velocity5   As Integer
    peak5       As Integer
    range6      As Integer
    velocity6   As Integer
    peak6       As Integer
    range7      As Integer
    velocity7   As Integer
    peak7       As Integer
    range8      As Integer
    velocity8   As Integer
    peak8       As Integer
    range9      As Integer
    velocity9   As Integer
    peak9       As Integer
    range10     As Integer
    velocity10  As Integer
    peak10      As Integer
    range11     As Integer
    velocity11  As Integer
    peak11      As Integer
    Buffer      As Integer
    ErrorN      As Integer
 End Type
 Dim openforms
 Public filterrec As datarec
 Public record As datarec
 Public targetpath$, sourcepath$, targetfile$, sourcefile$
 Public total%, rec%, message$, TVehic$, car$, Ccar$
 Public DRDV%, DRDX%, DRDXS%, Mpeak%, Mnobject%, MnErorr%, Mnshared%, DRDVS%,
 smooth%, smooth1%
 Public range%(), range1%(), velocity%(), velocity1%(), peak%(), peak1%()
 Public j%(), vehicle%(), vehicle1%(), VL1%(), VL2%(), VL3%(), VL4%(), DDX!(),
 DDV!(), ACC!(), DDX1!(), DDV1!(), ACC1!()

Sub fromrecord()
  range%(0) = filterrec.range0: velocity%(0) = filterrec.velocity0: peak%(0) =
filterrec.peak0
  range%(1) = filterrec.range1: velocity%(1) = filterrec.velocity1: peak%(1) =
filterrec.peak1
  range%(2) = filterrec.range2: velocity%(2) = filterrec.velocity2: peak%(2) =
filterrec.peak2
  range%(3) = filterrec.range3: velocity%(3) = filterrec.velocity3: peak%(3) =
filterrec.peak3
  range%(4) = filterrec.range4: velocity%(4) = filterrec.velocity4: peak%(4) =
filterrec.peak4
  range%(5) = filterrec.range5: velocity%(5) = filterrec.velocity5: peak%(5) =
filterrec.peak5
  range%(6) = filterrec.range6: velocity%(6) = filterrec.velocity6: peak%(6) =
filterrec.peak6
  range%(7) = filterrec.range7: velocity%(7) = filterrec.velocity7: peak%(7) =
filterrec.peak7
  range%(8) = filterrec.range8: velocity%(8) = filterrec.velocity8: peak%(8) =
filterrec.peak8
  range%(9) = filterrec.range9: velocity%(9) = filterrec.velocity9: peak%(9) =
filterrec.peak9
  range%(10) = filterrec.range10: velocity%(10) = filterrec.velocity10: peak%(10)
= filterrec.peak10
  range%(11) = filterrec.range11: velocity%(11) = filterrec.velocity11: peak%(11)
= filterrec.peak11
End Sub

Sub fromrecord1()
  range1%(0) = record.range0: velocity1%(0) = record.velocity0: peak1%(0) =
record.peak0
  range1%(1) = record.range1: velocity1%(1) = record.velocity1: peak1%(1) =
record.peak1
  range1%(2) = record.range2: velocity1%(2) = record.velocity2: peak1%(2) =
record.peak2
```

```
  range1%(3) = record.range3: velocity1%(3) = record.velocity3: peak1%(3) =
record.peak3
  range1%(4) = record.range4: velocity1%(4) = record.velocity4: peak1%(4) =
record.peak4
  range1%(5) = record.range5: velocity1%(5) = record.velocity5: peak1%(5) =
record.peak5
  range1%(6) = record.range6: velocity1%(6) = record.velocity6: peak1%(6) =
record.peak6
  range1%(7) = record.range7: velocity1%(7) = record.velocity7: peak1%(7) =
record.peak7
  range1%(8) = record.range8: velocity1%(8) = record.velocity8: peak1%(8) =
record.peak8
  range1%(9) = record.range9: velocity1%(9) = record.velocity9: peak1%(9) =
record.peak9
  range1%(10) = record.range10: velocity1%(10) = record.velocity10: peak1%(10) =
record.peak10
  range1%(11) = record.range11: velocity1%(11) = record.velocity11: peak1%(11) =
record.peak11
End Sub

Sub third1()
ReDim VL1%(1500), VL2%(1500), VL3%(1500), VL4%(1500), j%(11)
' VL1% is the start of the vehicle in te file
' VL2% is the end of the vehicle in te file
' VL3% is the actual number of the vehicle in te file
' VL4% is the old number of the vehicle in the new file
j%(0) = 4: j%(1) = 5: j%(2) = 6: j%(3) = 7: j%(4) = 0: j%(5) = 1: j%(6) = 2: j%(7)
= 3: j%(8) = 8: j%(9) = 9: j%(10) = 10: j%(11) = 11
filename1$ = targetpath$ & targetfile$ + ".thi"
Open filename1$ For Random As #1 Len = Len(filterrec)
total% = LOF(1) / Len(filterrec)
Form1.label9 = "The Program Renumbering The Vehicles in Sequence ":
Form1.label9.Refresh
 rec% = 1: i% = 0
 Do While rec% <= total%
  openform = DoEvents: Do While Form3.Visible = True: openform = DoEvents:
Form1.MousePointer = 1: Loop: Form1.MousePointer = 11
  Form1.text3 = Str$(rec%): Form1.text3.Refresh
  Get #1, rec%, filterrec: Call fromrecord
  For EF% = 0 To 11
   k% = j%(EF%)
  ' Form1.text7 = Str$(k%): Form1.text7.Refresh
   If (peak%(k%) <> 0) And (rec% > 1) Then
    Get #1, rec% - 1, record: Call fromrecord1
    F% = 0
    For EF1% = 0 To 11
      k1% = j%(EF1%)
      If peak1%(k1%) = peak%(k%) Then F% = 1: Exit For
    Next EF1%
    If (F% = 0) And (EF% > 0) Then
      Get #1, rec%, record: Call fromrecord1
      For EF1% = 0 To EF% - 1
       k1% = j%(EF1%)
       If peak1%(k1%) = peak%(k%) Then F% = 1: Exit For
      Next EF1%
    End If
    If F% = 0 Then
      i% = i% + 1: VL1%(i%) = rec%: VL3%(i%) = peak%(k%): VL4%(peak%(k%)) = i%
    Else
      VL2%(VL4%(peak%(k%))) = rec%
    End If
   ElseIf (peak%(k%) <> 0) Then
    F% = 0
    If EF% > 0 Then
      Get #1, rec%, record: Call fromrecord1
      For EF1% = 0 To EF% - 1
       k1% = j%(EF1%)
       If peak1%(k1%) = peak%(k%) Then F% = 1: Exit For
      Next EF1%
    End If
    If F% = 0 Then
     i% = i% + 1: VL1%(i%) = rec%: VL3%(i%) = peak%(k%): VL4%(peak%(k%)) = i%
```

```
     End If
    End If
   Next EF%
   rec% = rec% + 1
  Loop
 rec% = 1
 Do While rec% <= total%
  Form1.text3 = Str$(rec%): Form1.text3.Refresh
  openform = DoEvents: Do While Form3.Visible = True: openform = DoEvents:
 Form1.MousePointer = 1: Loop: Form1.MousePointer = 11
  Get #1, rec%, filterrec: Call fromrecord
  For k% = 0 To 11
   If peak%(k%) <> 0 Then peak%(k%) = VL4%(peak%(k%))
  Next k%
  Call torecord: Put #1, rec%, filterrec
  rec% = rec% + 1
 Loop
 '--------------------
 filename2$ = targetpath$ & targetfile$ & ".sev"
 Open filename2$ For Random As #2 Len = Len(filterrec)
 Form1.text5.Visible = True: Form1.label5.Visible = True: Form1.label5.Refresh
 Form1.text6.Visible = True: Form1.label6.Visible = True: Form1.label6.Refresh
 Dim DXV%(), DVV%()
 Form1.text5 = Str$(i%): Form1.text5.Refresh
 If i% > 0 Then
  For k% = 1 To i%
   ReDim DXV%(12), DVV%(12)
   Form1.text6 = Str$(k%): Form1.text6.Refresh
   For rec% = VL1%(k%) To VL2%(k%)
    openform = DoEvents: Do While Form3.Visible = True: openform = DoEvents:
 Form1.MousePointer = 1: Loop: Form1.MousePointer = 11
    Form1.text3 = Str$(rec%): Form1.text3.Refresh
    If rec% > VL1%(k%) Then
     Get #2, rec% - 1, record: Call fromrecord1
     For Count% = 0 To 11
      If peak1%(Count%) = k% Then Exit For
     Next Count%
     If Count% = 12 Then Stop
     R1% = range1%(Count%): V1% = velocity1%(Count%): P1 = record.elapse
     Get #1, rec%, filterrec: Call fromrecord
     P1 = filterrec.elapse - P1
     C1% = 0
     For EF% = 0 To 11
      k1% = j%(EF%)
      If peak%(k1%) = k% Then
       C1% = C1% + 1: DXV%(C1%) = range%(k1%): DVV%(C1%) = velocity%(k1%)
      End If
     Next EF%
     If C1% = 1 Then
      C2% = 1: C3% = 1
     ElseIf C1% > 1 Then
      C2% = C1%: C3% = C1%: DeltaX% = DXV%(1) - Int((R1% + V1% * P1)): DeltaV% =
DVV%(1) - V1%
      For Count% = 2 To C1%
       If (Abs(DeltaX%) > Abs(DXV%(Count%) - Int(R1% + V1% * P1))) And (Sgn(V1%) =
Sgn(DXV%(Count%) - Int(R1% + V1% * P1)) Or Abs(V1%) < 100) Then
        DeltaX% = DXV%(Count%) - Int(R1% + V1% * P1): C2% = Count%
       End If
       If (Abs(DeltaV%) > Abs(DVV%(Count%))) And (Sgn(V1%) = Sgn(DVV%(Count%)) Or
Abs(V1%) < 100) Then
        DeltaV% = DVV%(Count%): C3% = Count%
       End If
      Next Count%
     End If
    Else
     R1% = 0: V1% = 0: P1 = 0: C1% = 0
     Get #1, rec%, filterrec: Call fromrecord
     For EF% = 0 To 11
      k1% = j%(EF%)
      If peak%(k1%) = k% Then
       DXV%(1) = range%(k1%): DVV%(1) = velocity%(k1%): C2% = 1: C1% = 1: C3% = 1:
Exit For
```

```
      End If
     Next EF%
    End If
    If C1% <> 0 Then
      Get #2, rec%, record: Call fromrecord1
      If rec% = VL1%(k%) Then
       For EF1% = 0 To 11
        If peak1%(EF1%) = 0 Then
         VL3%(k%) = EF1%
         range1%(EF1%) = DXV%(C2%): velocity1%(EF1%) = DVV%(C3%): peak1%(EF1%) = k%
         Exit For
        End If
       Next EF1%
       If EF1% > 11 Then VL3%(k%) = 11
      Else
       range1%(VL3%(k%)) = DXV%(C2%): velocity1%(VL3%(k%)) = DVV%(C3%):
peak1%(VL3%(k%)) = k%
      End If
      Call torecord1: Put #2, rec%, record
    Else
     VL3%(k%) = -1
     Exit For
    End If
   Next rec%
  Next k%
End If
Close #1, 2
Kill filename1$
'----- Smoothing The Data ---------
If i% > 0 Then
filename1$ = targetpath$ & targetfile$ & ".sev"
'filename2$ = targetpath$ & targetfile$ & ".six"
'FileCopy filename1$, filename2$
Open filename1$ For Random As #2 Len = Len(filterrec)
Form1.label9 = "The Program Is Smoothing The Data ": Form1.label9.Refresh
For k% = 1 To i%
 If VL3%(k%) <> -1 Then
  openform = DoEvents: Do While Form3.Visible = True: openform = DoEvents:
Form1.MousePointer = 1: Loop: Form1.MousePointer = 11
  Form1.text6 = Str$(k%): Form1.text6.Refresh
  VhN% = VL2%(k%) - VL1%(k%) + 1
  If VhN% > 30 Then
    ReDim DDX!(VhN%), DDX1!(VhN% + smooth1% * 2), DDV!(VhN%), DDV1!(VhN% +
smooth1% * 2)
    rec1% = 0
    For rec% = VL1%(k%) To VL2%(k%)
     openform = DoEvents: Do While Form3.Visible = True: openform = DoEvents:
Form1.MousePointer = 1: Loop: Form1.MousePointer = 11
     rec1% = rec1% + 1
     Get 2#, rec%, filterrec: Call fromrecord
     DDX1!(rec1% + smooth1%) = range%(VL3%(k%))
     DDX!(rec1%) = range%(VL3%(k%))
     DDV1!(rec1% + smooth1%) = velocity%(VL3%(k%))
     DDV!(rec1%) = velocity%(VL3%(k%))
     Form1.text3 = Str$(rec%): Form1.text3.Refresh
    Next rec%
    DC! = 0: DCC! = 0
    For Count% = 1 To smooth1%: DC! = DC! + DDX!(Count%): DCC! = DCC! +
DDV!(Count%): Next Count%
    DC! = DC! / smooth1%: DCC! = DCC! / smooth1%
    For Count% = 1 To smooth1%: DDX1!(Count%) = DC!: DDV1!(Count%) = DCC!: Next
Count%
    DC! = 0: DCC! = 0
    For Count% = VhN% - smooth1% + 1 To VhN%: DC! = DC! + DDX!(Count%): DCC! =
DCC! + DDV!(Count%): Next Count%
    DC! = DC! / smooth1%: DCC! = DCC! / smooth1%
    For Count% = VhN% + smooth1% + 1 To VhN% + (2 * smooth1%): DDX1!(Count%) =
DC!: DDV1!(Count%) = DCC!: Next Count%
    For Count% = smooth1% + 1 To VhN% + smooth1%
     openform = DoEvents: Do While Form3.Visible = True: openform = DoEvents:
Form1.MousePointer = 1: Loop: Form1.MousePointer = 11
     DC! = 0: DCC! = 0
```

```
      For kk% = Count% - smooth1% To Count%
       DC! = DC! + DDX1!(kk%) * (kk% - Count% + smooth1% + 1)
       DCC! = DCC! + DDV1!(kk%) * (kk% - Count% + smooth1% + 1)
      Next kk%
      For kk% = 1 To smooth1%
       DC! = DC! + DDX1!(Count% + kk%) * (smooth1% - kk% + 1)
       DCC! = DCC! + DDV1!(Count% + kk%) * (smooth1% - kk% + 1)
      Next kk%
      DDX!(Count% - smooth1%) = DC! / (smooth1% + 1) / (smooth1% + 1)
      DDV!(Count% - smooth1%) = DCC! / (smooth1% + 1) / (smooth1% + 1)
      rec% = Count% - smooth1%
      Form1.text3 = Str$(rec%): Form1.text3.Refresh
     Next Count%
     rec1% = 0
     For rec% = VL1%(k%) To VL2%(k%)
      openform = DoEvents: Do While Form3.Visible = True: openform = DoEvents:
Form1.MousePointer = 1: Loop: Form1.MousePointer = 11
      rec1% = rec1% + 1
      Get 2#, rec%, filterrec: Call fromrecord
      range%(VL3%(k%)) = DDX!(rec1%)
      velocity%(VL3%(k%)) = DDV!(rec1%)
      Call torecord: Put 2#, rec%, filterrec
      Form1.text3 = Str$(rec%): Form1.text3.Refresh
     Next rec%
   End If
  End If
 Next k%
 End If
 Close #2
 Form1.text5.Visible = False: Form1.label5.Visible = False: Form1.label5.Refresh:
 Form1.text5.Refresh
 Form1.text6.Visible = False: Form1.label6.Visible = False: Form1.label6.Refresh:
 Form1.text6.Refresh
 Form1.text7.Visible = False: Form1.Label7.Visible = False: Form1.Label7.Refresh:
 Form1.text7.Refresh
 End Sub

 Sub torecord1()
  record.range0 = range1%(0): record.velocity0 = velocity1%(0): record.peak0 =
 peak1%(0)
  record.range1 = range1%(1): record.velocity1 = velocity1%(1): record.peak1 =
 peak1%(1)
  record.range2 = range1%(2): record.velocity2 = velocity1%(2): record.peak2 =
 peak1%(2)
  record.range3 = range1%(3): record.velocity3 = velocity1%(3): record.peak3 =
 peak1%(3)
  record.range4 = range1%(4): record.velocity4 = velocity1%(4): record.peak4 =
 peak1%(4)
  record.range5 = range1%(5): record.velocity5 = velocity1%(5): record.peak5 =
 peak1%(5)
  record.range6 = range1%(6): record.velocity6 = velocity1%(6): record.peak6 =
 peak1%(6)
  record.range7 = range1%(7): record.velocity7 = velocity1%(7): record.peak7 =
 peak1%(7)
  record.range8 = range1%(8): record.velocity8 = velocity1%(8): record.peak8 =
 peak1%(8)
  record.range9 = range1%(9): record.velocity9 = velocity1%(9): record.peak9 =
 peak1%(9)
  record.range10 = range1%(10): record.velocity10 = velocity1%(10): record.peak10 =
 peak1%(10)
  record.range11 = range1%(11): record.velocity11 = velocity1%(11): record.peak11 =
 peak1%(11)
 End Sub

 Sub torecord()
  filterrec.range0 = range%(0): filterrec.velocity0 = velocity%(0): filterrec.peak0
 = peak%(0)
  filterrec.range1 = range%(1): filterrec.velocity1 = velocity%(1): filterrec.peak1
 = peak%(1)
  filterrec.range2 = range%(2): filterrec.velocity2 = velocity%(2): filterrec.peak2
 = peak%(2)
```

```
 filterrec.range3 = range%(3): filterrec.velocity3 = velocity%(3): filterrec.peak3
= peak%(3)
 filterrec.range4 = range%(4): filterrec.velocity4 = velocity%(4): filterrec.peak4
= peak%(4)
 filterrec.range5 = range%(5): filterrec.velocity5 = velocity%(5): filterrec.peak5
= peak%(5)
 filterrec.range6 = range%(6): filterrec.velocity6 = velocity%(6): filterrec.peak6
= peak%(6)
 filterrec.range7 = range%(7): filterrec.velocity7 = velocity%(7): filterrec.peak7
= peak%(7)
 filterrec.range8 = range%(8): filterrec.velocity8 = velocity%(8): filterrec.peak8
= peak%(8)
 filterrec.range9 = range%(9): filterrec.velocity9 = velocity%(9): filterrec.peak9
= peak%(9)
 filterrec.range10 = range%(10): filterrec.velocity10 = velocity%(10):
filterrec.peak10 = peak%(10)
 filterrec.range11 = range%(11): filterrec.velocity11 = velocity%(11):
filterrec.peak11 = peak%(11)
End Sub

Sub convert():
 Dim T As Long:
 Form1.text3.Visible = True: Form1.text3.Refresh
 Form1.Label3.Visible = True: Form1.Label3.Refresh
 Form1.label9 = " Now The program is building the Random file "
 Form1.label9.Refresh
 filename2$ = sourcepath$ & sourcefile$
 filename3$ = targetpath$ & targetfile$ & ".fil"
 Open filename2$ For Input As #1
 Open filename3$ For Random As #2 Len = Len(filterrec)
 If LOF(2) <> 0 Then
  Close #2: Kill filename3$
  Open filename3$ For Random As #2 Len = Len(filterrec)
 End If
 Input #1, FrameNo$, ElapseTime$, PCTime$, RadarTimerPeriod$, UnCalSpeed$,
LongAccel$, LatAccel$, AccelTime$
 Input #1, Event$, R13$, V13$, SelBeam$, SelTrack$, R1$, V1$, P1$, R2$, V2$
 Input #1, p2$, R3$, V3$, P3$, R4$, V4$, p4$, r5$, v5$, p5$
 Input #1, R6$, V6$, P6$, R7$, V7$, p7$, R8$, V8$, P8$, R9$, V9$, p9$
 Input #1, R10$, V10$, p10$, R11$, V11$, P11$, R12$, V12$, P12$, Buffer$, ErrorN$
 rec% = 1
 Do While Not EOF(1)
  openform = DoEvents:  Do While Form3.Visible = True:  openform = DoEvents:
Form1.MousePointer = 1:  Loop:  Form1.MousePointer = 11
  Input #1, filterrec.FrameNo, filterrec.elapse, filterrec.PCTime,
filterrec.radartime, filterrec.UnCalSpeed, filterrec.LongAcc
  Input #1, filterrec.LatAcc, filterrec.AccTime, filterrec.Event,
filterrec.range12, filterrec.velocity12, filterrec.SelBeam, filterrec.SelTrack
  Input #1, filterrec.range0, filterrec.velocity0, filterrec.peak0,
filterrec.range1, filterrec.velocity1, filterrec.peak1
  Input #1, filterrec.range2, filterrec.velocity2, filterrec.peak2,
filterrec.range3, filterrec.velocity3, filterrec.peak3
  Input #1, filterrec.range4, filterrec.velocity4, filterrec.peak4,
filterrec.range5, filterrec.velocity5, filterrec.peak5
  Input #1, filterrec.range6, filterrec.velocity6, filterrec.peak6,
filterrec.range7, filterrec.velocity7, filterrec.peak7
  Input #1, filterrec.range8, filterrec.velocity8, filterrec.peak8,
filterrec.range9, filterrec.velocity9, filterrec.peak9
  Input #1, filterrec.range10, filterrec.velocity10, filterrec.peak10,
filterrec.range11, filterrec.velocity11, filterrec.peak11
  Input #1, filterrec.Buffer, filterrec.ErrorN
  Put #2, rec%, filterrec
  Form1.text3.Text = Str$(rec%)
  Form1.text3.Refresh
  rec% = rec% + 1
 Loop
 Close #1, 2
End Sub

Sub first()
 filename1$ = targetpath$ & targetfile$ & ".fil"
 filename2$ = targetpath$ & targetfile$ & ".fir"
```

```
 filename3$ = targetpath$ & targetfile$ & ".sev"
 Open filename1$ For Random As #1 Len = Len(filterrec)
 Open filename2$ For Random As #2 Len = Len(filterrec)
 Open filename3$ For Random As #3 Len = Len(filterrec)
 total% = LOF(1) / Len(filterrec)
 Form1.label9 = "The program is Filtering The false Reading from the Random File":
Form1.label9.Refresh
 Form1.text4.Text = Str$(total%): Form1.text4.Refresh
 rec% = 1
 Do While (rec% <= total%)
  openform = DoEvents:  Do While Form3.Visible = True:  openform = DoEvents:
Form1.MousePointer = 1:  Loop:  Form1.MousePointer = 11
  Get #1, rec%, filterrec
  Call fromrecord
'-----------
  For i% = 0 To 11
   If velocity%(i%) = 0 Then
    If (range%(i%) = -1) Or peak%(i%) < Mpeak% Then
     range%(i%) = 0: peak%(i%) = 0
    Else
     F% = 0
     Select Case i%
      Case Is < 4
       For j1% = 0 To 3
        If (Abs(range%(i%) - range%(j1%)) < 100) And (i% <> j1%) Then F% = 1: Exit
For
       Next j1%
      Case Is > 7
       For j1% = 8 To 11
        If (Abs(range%(i%) - range%(j1%)) < 100) And (i% <> j1%) Then F% = 1: Exit
For
       Next j1%
      Case Else
       For j1% = 4 To 7
        If (Abs(range%(i%) - range%(j1%)) < 100) And (i% <> j1%) Then F% = 1: Exit
For
       Next j1%
     End Select
     If F% = 0 Then
      If rec% > 1 Then
       Get #1, rec% - 1, record: Call fromrecord1
       If velocity1%(i%) = 0 Then
        range%(i%) = 0: peak%(i%) = 0
       Else
        If Abs(range1%(i%) - range%(i%)) > DRDX% Then
          range%(i%) = 0: peak%(i%) = 0
        ElseIf rec% < total% Then
          Get #1, rec% + 1, record: Call fromrecord1
          If velocity1%(i%) = 0 Then range%(i%) = 0: peak%(i%) = 0
        End If
       End If
      Else
       range%(i%) = 0: peak%(i%) = 0
      End If
     Else
      range%(i%) = 0: peak%(i%) = 0
     End If
    End If
  ElseIf (range%(i%) = -1) Or (peak%(i%) < Mpeak%) Then
    velocity%(i%) = 0: range%(i%) = 0: peak%(i%) = 0
  End If
 Next i%
 Get #1, rec%, filterrec
 Call torecord
'------------
 Put #2, rec%, filterrec
 Form1.text3.Text = Str$(rec%): Form1.text3.Refresh
 rec% = rec% + 1
Loop
Form1.label9 = "The Program Is Ereasing the Peaks of the Ranges":
Form1.label9.Refresh
rec% = 1
```

```
 Do While (rec% <= total)
  openform = DoEvents: Do While Form3.Visible = True: openform = DoEvents:
Form1.MousePointer = 1: Loop: Form1.MousePointer = 11
  Get #2, rec%, filterrec: Call fromrecord
'-----------
  For i% = 0 To 11
   If peak%(i%) <> 0 Then peak%(i%) = 0
  Next i%
  Call torecord
  Put #2, rec%, filterrec
  For i% = 0 To 11: velocity%(i%) = 0: range%(i%) = 0: Next i%
  Call torecord
  Put #3, rec%, filterrec
  rec% = rec% + 1
  Form1.text3.Text = Str$(rec%): Form1.text3.Refresh
 Loop
 Close #1, 2, 3
' Form1.Label13.Visible = False: Form1.label14.Visible = False:
Form1.label15.Visible = False
 Kill filename1$
End Sub
Sub filterACC()
filename1$ = targetpath$ & targetfile$ & ".fil"
Open filename1$ For Random As #1 Len = Len(filterrec)
total% = LOF(1) / Len(filterrec)
ReDim DDX!(total%), DDX1!(total% + smooth1% * 2)
Form1.label9 = "The Program Is Smoothing the Acceleration reading":
Form1.label9.Refresh
For rec% = 1 To total%
 openform = DoEvents:  Do While Form3.Visible = True:  openform = DoEvents:
Form1.MousePointer = 1:  Loop:  Form1.MousePointer = 11
 Get 1#, rec%, filterrec
 DDX1!(rec% + smooth1%) = filterrec.LongAcc
 DDX!(rec%) = filterrec.LongAcc
 Form1.text3 = Str$(rec%): Form1.text3.Refresh
Next rec%
DC! = 0: For i% = 1 To smooth1%: DC! = DC! + DDX!(i%): Next i%
DC! = DC! / smooth1%: For i% = 1 To smooth1%: DDX1!(i%) = DC!: Next i%
DC! = 0: For i% = total% - smooth1% + 1 To total%: DC! = DC! + DDX!(i%): Next i%
DC! = DC! / smooth1%: For i% = total% + smooth1% + 1 To total% + (2 * smooth1%):
DDX1!(i%) = DC!: Next i%
For i% = smooth1% + 1 To total% + smooth1%
 openform = DoEvents:  Do While Form3.Visible = True:  openform = DoEvents:
Form1.MousePointer = 1:  Loop:  Form1.MousePointer = 11
 DC! = 0
 For k% = i% - smooth1% To i%
   DC! = DC! + DDX1!(k%) * (k% - i% + smooth1% + 1)
 Next k%
 For k% = 1 To smooth1%
   DC! = DC! + DDX1!(i% + k%) * (smooth1% - k% + 1)
 Next k%
 DDX!(i% - smooth1%) = DC! / (smooth1% + 1) / (smooth1% + 1)
 rec% = i% - smooth1%
 Form1.text3 = Str$(rec%): Form1.text3.Refresh
Next i%
If smooth% >= total% Then smoth% = total% - 1 Else smoth% = smooth%
ReDim DDX1!(total% + smoth% * 2), ACC!(total%)
For rec% = 1 To total%
 openform = DoEvents:  Do While Form3.Visible = True:  openform = DoEvents:
Form1.MousePointer = 1:  Loop:  Form1.MousePointer = 11
 DDX1!(rec% + smoth%) = DDX!(rec%)
 ACC!(rec%) = DDX!(rec%)
 Form1.text3 = Str$(rec%): Form1.text3.Refresh
Next rec%
DC! = 0: For i% = 1 To smoth%: DC! = DC! + DDX!(i%): Next i%
DC! = DC! / smoth%: For i% = 1 To smoth%: DDX1!(i%) = DC!: Next i%
DC! = 0: For i% = total% - smoth% + 1 To total%: DC! = DC! + DDX!(i%): Next i%
DC! = DC! / smoth%: For i% = total% + smoth% + 1 To total% + (2 * smoth%):
DDX1!(i%) = DC!: Next i%
For i% = smoth% + 1 To total% + smoth%
 openform = DoEvents:  Do While Form3.Visible = True:  openform = DoEvents:
Form1.MousePointer = 1:  Loop:  Form1.MousePointer = 11
```

```
  DC! = 0
  For k% = i% - smoth% To i%
    DC! = DC! + DDX1!(k%) * (k% - i% + smoth% + 1)
  Next k%
  For k% = 1 To smoth%
    DC! = DC! + DDX1!(i% + k%) * (smoth% - k% + 1)
  Next k%
  DDX!(i% - smoth%) = DC! / (smoth% + 1) / (smoth% + 1)
  rec% = i% - smoth%
  Form1.text3 = Str$(rec%): Form1.text3.Refresh
 Next i%
 For rec% = 1 To total%
  openform = DoEvents:  Do While Form3.Visible = True:  openform = DoEvents:
 Form1.MousePointer = 1:  Loop:  Form1.MousePointer = 11
  Get 1#, rec%, filterrec
  filterrec.velocity12 = ACC!(rec%)
  filterrec.SelTrack = ACC!(rec%) - DDX!(rec%)
  Put 1#, rec%, filterrec
  Form1.text3 = Str$(rec%): Form1.text3.Refresh
 Next rec%
 ReDim ACC!(1), DDX!(1), DDX1!(1)
 Close 1#
 End Sub

 Sub Filterspeed()
 filename1$ = targetpath$ & targetfile$ & ".fil"
 Open filename1$ For Random As #1 Len = Len(filterrec)
 total% = LOF(1) / Len(filterrec)
 Form1.text4.Visible = True: Form1.Label4.Visible = True: Form1.Label4.Refresh
 Form1.text4.Text = Str$(total%): Form1.text4.Refresh
 ReDim DDV!(total%), DDV1!(total% + smooth1% * 2), ACC!(total%), ACC1!(total% +
 smooth1% * 2)
 Form1.label9 = "The Program Is Smoothing the Speed's reading":
 Form1.label9.Refresh
 For rec% = 1 To total%
  openform = DoEvents:  Do While Form3.Visible = True:  openform = DoEvents:
 Form1.MousePointer = 1:  Loop:  Form1.MousePointer = 11
  Get 1#, rec%, filterrec
  DDV1!(rec% + smooth1%) = filterrec.UnCalSpeed
  DDV!(rec%) = filterrec.UnCalSpeed
  Form1.text3 = Str$(rec%): Form1.text3.Refresh
 Next rec%
 DC! = 0: For i% = 1 To smooth1%: DC! = DC! + DDV!(i%): Next i%
 DC! = DC! / smooth1%: For i% = 1 To smooth1%: DDV1!(i%) = DC!: Next i%
 DC! = 0: For i% = total% - smooth1% + 1 To total%: DC! = DC! + DDV!(i%): Next i%
 DC! = DC! / smooth1%: For i% = total% + smooth1% + 1 To total% + (2 * smooth1%):
 DDV1!(i%) = DC!: Next i%
 For i% = smooth1% + 1 To total% + smooth1%
  openform = DoEvents:  Do While Form3.Visible = True:  openform = DoEvents:
 Form1.MousePointer = 1:  Loop:  Form1.MousePointer = 11
  DC! = 0
  For k% = i% - smooth1% To i%
    DC! = DC! + DDV1!(k%) * (k% - i% + smooth1% + 1)
  Next k%
  For k% = 1 To smooth1%
    DC! = DC! + DDV1!(i% + k%) * (smooth1% - k% + 1)
  Next k%
  DDV!(i% - smooth1%) = DC! / (smooth1% + 1) / (smooth1% + 1)
  rec% = i% - smooth1%
  Form1.text3 = Str$(rec%): Form1.text3.Refresh
 Next i%
 For rec% = 1 To total%
  openform = DoEvents:  Do While Form3.Visible = True:  openform = DoEvents:
 Form1.MousePointer = 1:  Loop:  Form1.MousePointer = 11
  Get 1#, rec%, filterrec
  filterrec.range12 = DDV!(rec%)
  If rec% < total% - 4 Then
   Get 1#, rec% + 5, record
   filterrec.SelBeam = (DDV!(rec% + 5) - DDV!(rec%)) / (record.elapse -
 filterrec.elapse) / 36
  Else
   If rec% <> total% Then
```

```
    Get 1#, total%, record
    filterrec.SelBeam = (DDV!(total%) - DDV!(rec%)) / (record.elapse -
filterrec.elapse) / 36
   Else
    Get 1#, total% - 1, record
    filterrec.SelBeam = record.SelBeam
   End If
  End If
  ACC!(rec%) = filterrec.SelBeam
  ACC1!(rec% + smooth1%) = ACC!(rec%)
  Put 1#, rec%, filterrec
  Form1.text3 = Str$(rec%): Form1.text3.Refresh
 Next rec%
 DC! = 0: For i% = 1 To smooth1%: DC! = DC! + ACC!(i%): Next i%
 DC! = DC! / smooth1%: For i% = 1 To smooth1%: ACC1!(i%) = DC!: Next i%
 DC! = 0: For i% = total% - smooth1% + 1 To total%: DC! = DC! + ACC!(i%): Next i%
 DC! = DC! / smooth1%: For i% = total% + smooth1% + 1 To total% + (2 * smooth1%):
 DDV1!(i%) = DC!: Next i%
 For i% = smooth1% + 1 To total% + smooth1%
  openform = DoEvents:  Do While Form3.Visible = True:  openform = DoEvents:
 Form1.MousePointer = 1:  Loop:  Form1.MousePointer = 11
  DC! = 0
  For k% = i% - smooth1% To i%
    DC! = DC! + ACC1!(k%) * (k% - i% + smooth1% + 1)
  Next k%
  For k% = 1 To smooth1%
    DC! = DC! + ACC1!(i% + k%) * (smooth1% - k% + 1)
  Next k%
  ACC!(i% - smooth1%) = DC! / (smooth1% + 1) / (smooth1% + 1)
  rec% = i% - smooth1%
  Form1.text3 = Str$(rec%): Form1.text3.Refresh
 Next i%
 For rec% = 1 To total%
  openform = DoEvents:  Do While Form3.Visible = True:  openform = DoEvents:
 Form1.MousePointer = 1:  Loop:  Form1.MousePointer = 11
  Get 1#, rec%, filterrec
  If rec% <> total% Then
    filterrec.SelBeam = ACC!(rec%)
  Else
    filterrec.SelBeam = ACC!(rec% - 1)
  End If
  Put 1#, rec%, filterrec
  Form1.text3 = Str$(rec%): Form1.text3.Refresh
 Next rec%
 ReDim DDV!(1), DDV1!(1), ACC1!(1), ACC!(1)
 Close 1#
 End Sub

 Sub second()
 ReDim VL1%(1500), VL2%(1500), VL3%(1500)
 Form1.text7.Visible = True: Form1.Label7.Visible = True: Form1.Label7.Refresh
 filename2$ = targetpath$ & targetfile$ & ".fir"
 'filename2$ = targetpath$ & targetfile$ & ".sec"
 'FileCopy filename1$, filename2$
 'Kill filename1$
 Open filename2$ For Random As #1 Len = Len(filterrec)
 total% = LOF(1) / Len(filterrec)
 filename3$ = targetpath$ & targetfile$ & ".thi"
 filename4$ = targetpath$ & targetfile$ & ".sev"
 FileCopy filename4$, filename3$
 Open filename3$ For Random As #2 Len = Len(filterrec)
 Form1.text4.Text = total%: Form1.text4.Refresh
 For i% = 0 To 11: range%(i%) = 0: velocity%(i%) = 0: peak%(i%) = 0: Next i%
 For i% = 0 To 11: range1%(i%) = 0: velocity1%(i%) = 0: peak1%(i%) = 0: Next i%
 j%(0) = 4: j%(1) = 0: j%(2) = 8: F% = 0
 For p% = 0 To 2
  Select Case p%
   Case 0
    Form1.text7.Text = "5,6,7,8"
   Case 1
    Form1.text7.Text = "1,2,3,4"
   Case 2
```

```
    Form1.text7.Text = "9,10,11,12"
  End Select
  Form1.text7.Refresh
   rec% = 1: vehicle%(0) = F% + 1
'---------------------------
' Identify the sequence between the radar reading
'---------------------------
   Dim DXV%(), DVV%()
   Form1.label9 = "The program is Identifing the sequence between the radar reading
 in The Random File in the ranges :  " & Form1.text7.Text
   Form1.label9.Refresh
   Do While rec% < total%
    openform = DoEvents: Do While Form3.Visible = True: openform = DoEvents:
Form1.MousePointer = 1: Loop: Form1.MousePointer = 11
    Get #1, rec%, record
    Get #1, rec% + 1, filterrec
    Call fromrecord1: Call fromrecord
    For k% = j%(p%) To (j%(p%) + 3)
     If range1%(k%) <> 0 Then
      If peak1%(k%) = 0 Then
        F% = F% + 1: VL1%(F%) = rec%: peak1%(k%) = F%
        Call torecord1: Put #1, rec%, record
      End If
      If (Abs(range%(k%) - (range1%(k%) + velocity1%(k%) * (filterrec.elapse -
record.elapse))) < DRDX%) And (Abs(velocity%(k%) - velocity1%(k%)) < DRDV%) Then
       'If (Abs(range%(k%) - range1%(k%)) < DRDX%) And (Abs(velocity%(k%) -
velocity1%(k%)) < DRDV%) Then
        peak%(k%) = peak1%(k%)
      Else
       For i% = j%(p%) To (j%(p%) + 3)
        If (Abs(range%(i%) - (range1%(k%) + velocity1%(k%) * (filterrec.elapse -
record.elapse))) < DRDX%) And (Abs(velocity%(i%) - velocity1%(k%)) < DRDV%) Then
        'If (Abs(range%(i%) - range1%(k%)) < DRDX%) And (Abs(velocity%(i%) -
velocity1%(k%)) < DRDV%) Then
         If peak%(i%) = 0 Then peak%(i%) = peak1%(k%): Exit For
        End If
       Next i%
      End If
     End If
    Next k%
    Call torecord:   Put #1, rec% + 1, filterrec
    rec% = rec% + 1
    Form1.text3.Text = Str$(rec%): Form1.text3.Refresh
    Loop
'---------------------------
' Arranging and Saving the data for vehicles in successiv way
'---------------------------
   Form1.label9 = "The program is Arranging and Saving the vehicles' data in
successiv way in The ranges : " & Form1.text7.Text
   Form1.label9.Refresh
   Form1.label5.Visible = True: Form1.text5.Visible = True: Form1.label5.Refresh
   vehicle%(1) = F%:
   vehicle1%(p%) = vehicle%(1) - vehicle%(0) + 1
   Form1.text5.Text = Str$(vehicle1%(p%)): Form1.text5.Refresh
   For i% = vehicle%(0) To vehicle%(1)
    Form1.text6.Text = Str$(i% - vehicle%(0) + 1)
    rec% = VL1%(i%): F1% = 0
    Do While rec% < total%
     openform = DoEvents: Do While Form3.Visible = True: openform = DoEvents:
Form1.MousePointer = 1: Loop: Form1.MousePointer = 11
     Form1.text3.Text = Str$(rec%): Form1.text3.Refresh
     Get #1, rec%, record: Call fromrecord1
     For k% = j%(p%) To (j%(p%) + 3)
      F2% = 0
      If peak1%(k%) = i% Then F1% = F1% + 1: F2% = 1: Exit For
     Next k%
     If F2% = 0 And F1% > 0 Then VL2%(i%) = rec% - 1: Exit Do
     If F2% = 1 And F1% > 0 Then
      Get #2, rec%, filterrec: Call fromrecord
      If F1% = 1 Then
       st% = j%(p%): EN% = (j%(p%) + 3)
310:
```

```
       For L% = st% To EN%
        If range%(L%) = 0 Then Exit For
       Next L%
       If st% = j%(p%) Then L1% = L%
       If L% > (j%(p%) + 3) Then L% = (j%(p%) + 3)
       If rec% > 1 Then
        Get #2, rec% - 1, filterrec: Call fromrecord
        If (range%(L%) <> 0) And (Abs(range1%(k%) - range%(L%)) > DRDXS%) Then
         If L% < (j%(p%) + 3) Then
          Get #2, rec%, filterrec: Call fromrecord
          st% = L% + 1: GoTo 310
         Else
          L% = L1%
         End If
        End If
        Get #2, rec%, filterrec: Call fromrecord
       End If
      End If
      range%(L%) = range1%(k%): VL3%(i%) = L%
      velocity%(L%) = velocity1%(k%): peak%(L%) = peak1%(k%)
      Call torecord: Put #2, rec%, filterrec
     End If
     rec% = rec% + 1
    Loop
   Next i%
 Form1.text5.Text = "": Form1.text5.Refresh
 Next p%
 '--------------------------
 ' Matching Vehicle with each other
 '--------------------------
 Form1.label9 = "The program is Matching The Vehicles with each other"
 Form1.label9.Refresh
 For p% = 0 To 1
  Select Case p%
    Case 0
     Form1.text7.Text = "5,6,7,8"
     start1% = 1: end1% = vehicle1%(0)
     start2% = vehicle1%(0) + 1: end2% = vehicle%(1)
    Case 1
     Form1.text7.Text = "1,2,3,4"
     start1% = vehicle1%(0) + 1: end1% = vehicle1%(0) + vehicle1%(1)
     start2% = vehicle1%(0) + vehicle1%(1) + 1: end2% = vehicle%(1)
  End Select
  Form1.text7.Refresh
  Form1.text5.Text = Str$(vehicle1%(p%)): Form1.text5.Refresh
  For k% = start1% To end1%
   If start1% > vehicle1%(0) Then
    Form1.text6.Text = Str$(k% - vehicle1%(0))
   Else
    Form1.text6.Text = Str$(k%)
   End If
   Form1.text6.Refresh
   For i% = start2% To end2%
    openform = DoEvents: Do While Form3.Visible = True: openform = DoEvents:
Form1.MousePointer = 1: Loop: Form1.MousePointer = 11
    Form1.text8.Text = Str$(i%): Form1.text8.Refresh
    If Not ((VL1%(i%) > VL2%(k%)) Or (VL2%(i%) < VL1%(k%))) Then
      If VL1%(i%) < VL1%(k%) Then t1% = VL1%(k%) Else t1% = VL1%(i%)
      If VL2%(i%) < VL2%(k%) Then t2% = VL2%(i%) Else t2% = VL2%(k%)
      rec% = t1%: DeltaX = 0: DeltaV = 0: Nomiss% = 0
      Do While rec% <= t2%
       openform = DoEvents: Do While Form3.Visible = True: openform = DoEvents:
Form1.MousePointer = 1: Loop: Form1.MousePointer = 11
       Get #2, rec%, filterrec
       Call fromrecord
       If Abs(range%(VL3%(i%)) - range%(VL3%(k%))) > DRDXS% Then Nomiss% = Nomiss%
+ 1
       If Nomiss% > Int((t2% - t1%) * 0.1) Then Exit For
       If DeltaX < Abs(range%(VL3%(i%)) - range%(VL3%(k%))) Then DeltaX =
Abs(range%(VL3%(i%)) - range%(VL3%(k%)))
       If DeltaV < Abs(velocity%(VL3%(i%)) - velocity%(VL3%(k%))) Then DeltaV =
Abs(velocity%(VL3%(i%)) - velocity%(VL3%(k%)))
```

```
        rec% = rec% + 1
      Loop
      If (((t2% - t1%) >= Mnshared%) And (DeltaX <= DRDXS%)) Or (((t2% - t1%) <
Mnshared%) And (DeltaX <= DRDXS%) And (DeltaV <= DRDVS%)) Or ((Nomiss% <= Int((t2%
- t1%) * 0.1)) And (DeltaX <= DRDXS% * 3)) Then
       F% = 0
       If (peak%(VL3%(i%)) > peak%(VL3%(k%))) Then
         peakn% = peak%(VL3%(k%))
         If VL3%(i%) < 4 Then
          For k1% = 0 To 3
           For rec% = VL1%(i%) To VL2%(i%)
            Get #2, rec%, filterrec: Call fromrecord
            If (peak%(k1%) = peakn%) And (Abs(range%(k1%) - range%(VL3%(i%))) >
DRDXS%) Then
              F% = 1: Exit For
            End If
           Next rec%
           If F% = 1 Then Exit For
          Next k1%
         Else
          For k1% = 8 To 11
           For rec% = VL1%(i%) To VL2%(i%)
            Get #2, rec%, filterrec: Call fromrecord
            If (peak1%(k1%) = peakn%) And (Abs(range1%(k1%) - range1%(VL3%(i%))) >
DRDXS%) Then
              F% = 1: Exit For
            End If
           Next rec%
           If F% = 1 Then Exit For
          Next k1%
         End If
         If F% = 0 Then
          CO% = VL3%(i%): NV% = peakn%: t1% = VL1%(i%): t2% = VL2%(i%)
         End If
       ElseIf (peak%(VL3%(i%)) < peak%(VL3%(k%))) Then
        peakn% = peak%(VL3%(i%))
        For k1% = j%(p%) To (j%(p%) + 3)
           For rec% = VL1%(k%) To VL2%(k%)
            Get #2, rec%, filterrec: Call fromrecord
            If (peak%(k1%) = peakn%) And (Abs(range%(k1%) - range%(VL3%(k%))) >
DRDXS%) Then
              F% = 1: Exit For
            End If
           Next rec%
           If F% = 1 Then Exit For
        Next k1%
        If F% = 0 Then
           CO% = VL3%(k%): NV% = peakn%: t1% = VL1%(k%): t2% = VL2%(k%)
        End If
       Else
        F% = 1
       End If
       If F% = 0 Then
        rec% = t1%
        Do While rec% <= t2%
         openform = DoEvents: Do While Form3.Visible = True: openform = DoEvents:
Form1.MousePointer = 1: Loop: Form1.MousePointer = 11
         Get #2, rec%, filterrec: Call fromrecord
         peak%(CO%) = NV%: Call torecord: Put #2, rec%, filterrec
         Form1.text3 = Val(rec%): Form1.text3.Refresh
         rec% = rec% + 1
        Loop
       End If
     End If
   End If
  Next i%
 Next k%
Next p%
Form1.label8.Visible = False: Form1.label8.Refresh
Form1.text8.Visible = False: Form1.text8.Refresh
Form1.label5.Visible = False: Form1.label5.Refresh
Form1.text5.Visible = False: Form1.text5.Refresh
```

```
 Form1.label6.Visible = False: Form1.label6.Refresh
 Form1.text6.Visible = False: Form1.text6.Refresh
 '-------------------------------------
 ' Ereasing the short reading in the file
 '-------------------------------------
 Form1.label9 = "The program Now Ereasing the Short reading ": Form1.label9.Refresh
 For k% = 0 To 11
   Form1.text7 = Str$(k%): Form1.text7.Refresh
   rec% = 1: F1% = 0
   Get #2, rec%, filterrec: Call fromrecord
   F% = peak%(k%)
   If F% <> 0 Then F1% = F1% + 1
   rec% = 2
  Do While rec% <= total%
   openform = DoEvents: Do While Form3.Visible = True: openform = DoEvents:
 Form1.MousePointer = 1: Loop: Form1.MousePointer = 11
   Get #2, rec%, filterrec: Call fromrecord
   If ((peak%(k%) = 0) Or (peak%(k%) <> F%)) And (F1% <= Mnobject%) And (F1% <> 0)
 Then
    For i% = (rec% - F1%) To rec% - 1
     Get #2, i%, record: Call fromrecord1
     peak1%(k%) = 0: range1%(k%) = 0: velocity1%(k%) = 0
     Call torecord1: Put #2, i%, record:
    Next i%
    F1% = 0: F% = 0
   End If
   If (peak%(k%) = F%) And (F% <> 0) Then F1% = F1% + 1
   If (peak%(k%) = 0) And (F1% > Mnobject%) Then F1% = 0: F% = 0
   If peak%(k%) <> F% Then F% = peak%(k%): F1% = 1
   Form1.text3 = Str$(rec%): Form1.text3.Refresh
   rec% = rec% + 1
  Loop
Next k%
Close #1, 2
Kill filename2$
'Form1.Label13.Visible = False: Form1.Label14.Visible = False:
Form1.Label15.Visible = False:
'Form1.Label16.Visible = False: Form1.Label17.Visible = False:
Form1.Label18.Visible = False:
End Sub

Sub finall()
 If targetpath$ = sourcepath$ Then
  filename1$ = sourcepath$ & sourcefile$
  filename2$ = targetpath$ & targetfile$ & ".old"
  Name filename1$ As filename2$
 End If
 filename1$ = targetpath$ & targetfile$ & ".txt"
 filename2$ = targetpath$ & targetfile$ & ".sev"
 Form1.label9 = "The program is building the filtered file for the video data
program, its extention would be (.txt) and has the same name of the source file ":
Form1.label9.Refresh
 Open filename1$ For Output As #1
 If LOF(1) <> 0 Then
  Close #1: Kill filename1$
  Open filename1$ For Output As #1
 End If
 Open filename2$ For Random As #2 Len = Len(filterrec)
  Print #1, "FrameNo"; ","; "ElapseTime"; ", "; "PCTime"; ", ";
"RadarTimerPeriod"; ","; "Speed"; ","; "LongACC"; ","; "LatAccel"; ",";
"AccelTime"; ",";
  Print #1, "Event"; ","; "Smo_Speed"; ","; "Smo_ACC"; ","; "CalcACC"; ",";
"Smo_ACC_Hill"; ",";
  Print #1, " range1"; ","; " velocity1"; ","; " peak1"; ","; " range2"; ","; "
velocity2"; ",";
  Print #1, "peak2"; ","; " range3"; ","; " velocity3"; ","; " peak3"; ","; "
range4"; ","; " velocity4"; ","; " peak4"; ","; " range5"; ","; " velocity5"; ",";
" peak5"; ",";
  Print #1, "range6"; ","; " velocity6"; ","; " peak6"; ","; " range7"; ","; "
velocity7"; ","; " peak7"; ","; " range8"; ","; " velocity8"; ","; " peak8"; ",";
" range9"; ","; " velocity9"; ","; " peak9"; ",";
```

```
  Print #1, "range10"; ","; " velocity10"; ","; " peak10"; ","; " range11"; ","; "
velocity11"; ","; " peak11"; ","; " range12"; ","; " velocity12"; ","; " peak12";
","; " Buffer"; ","; " ErrorN"
 rec% = 1
 Do While Not EOF(2)
  openform = DoEvents
  Do While Form3.Visible = True
   openform = DoEvents: Form1.MousePointer = 1
  Loop
  Form1.MousePointer = 11
  Get #2, rec%, filterrec
  Call fromrecord
  Print #1, filterrec.FrameNo; ","; filterrec.elapse; ","; filterrec.PCTime; ",";
filterrec.radartime; ","; filterrec.UnCalSpeed; ","; filterrec.LongAcc; ",";
  Print #1, filterrec.LatAcc; ","; filterrec.AccTime; ","; filterrec.Event; ",";
filterrec.range12; ","; filterrec.velocity12; ","; filterrec.SelBeam; ",";
filterrec.SelTrack; ",";
  For i% = 0 To 11
     If peak%(i%) = 0 Then
      Print #1, constr$(range%(i%)); ","; constr$(velocity%(i%)); ","; peak%(i%);
",";
     Else
      Print #1, range%(i%); ","; velocity%(i%); ","; peak%(i%); ",";
     End If
  Next i%
  Print #1, filterrec.Buffer; ","; filterrec.ErrorN
  Form1.text3.Text = Str$(rec%): Form1.text3.Refresh
  rec% = rec% + 1
 Loop
 Close #1, 2
' Form1.Label13.Visible = False: Form1.label14.Visible = False:
Form1.label15.Visible = False:
 'Form1.Label3.Visible = False: Form1.Label4.Visible = False
 'Form1.text3.Visible = False: Form1.text4.Visible = False
 Kill filename2$
End Sub
'-----------------------------------------------------------------
Function constr$(m%)
If m% = 0 Then constr$ = "" Else constr$ = Str$(m%)
End Function
```

APPENDIX (2)

THE APPROACH PROCESS TEST PROGRAM CODE & ITERFACE

This program was developed using Turbo Pascal programming language. It simulates the approach process for a vehicle closing to slower one ahead which has a constant speed. Figure a2.1 shows the program graphical interface that helps to enhance the understanding of how the model works. The program code is listed below:

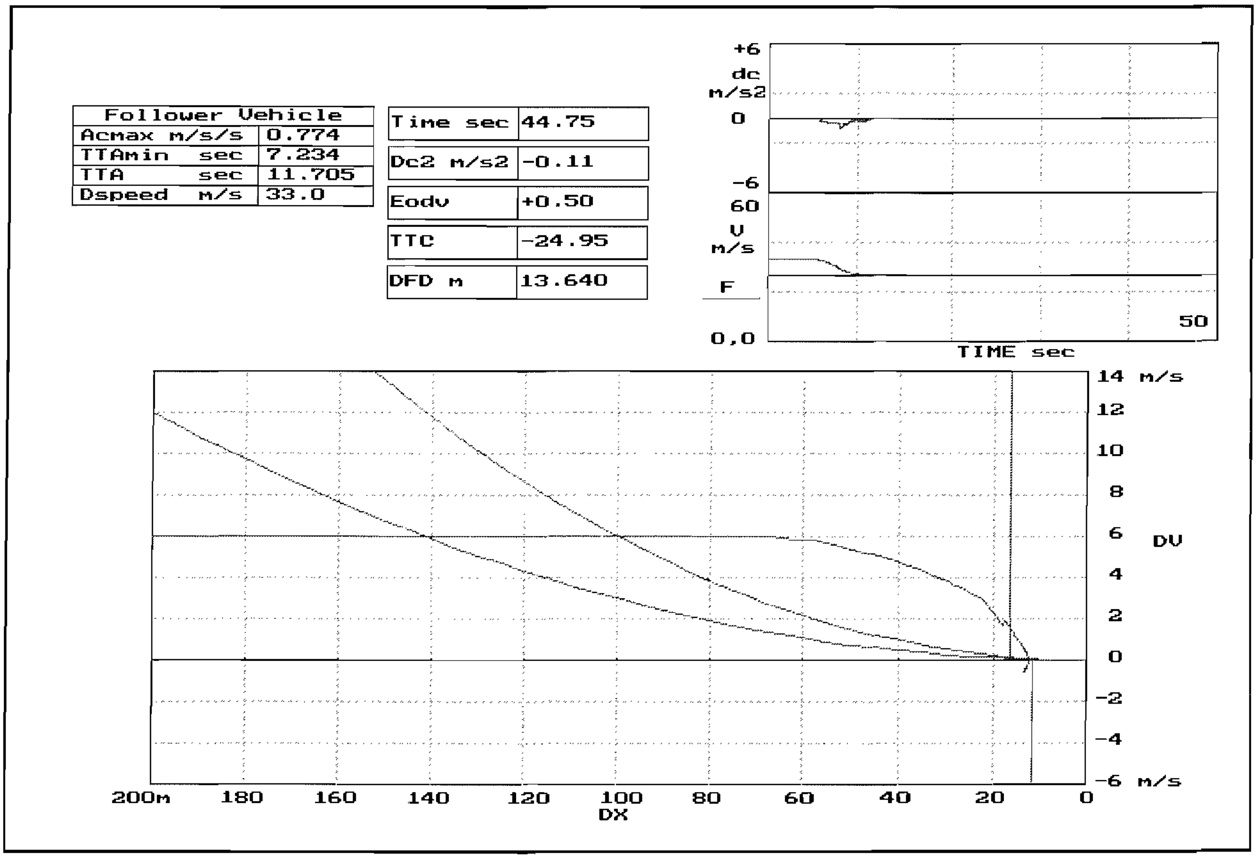


Figure a2.1: The Graphical interface of the Approach process programe.

The CODE LIST

```
program ClosPro5a;
uses CLd5,Crt,Graph;
label 100,40,30,20;
var  key    : boolean;
begin
 randomize;
 norm1[1]:=1000; norm1[2]:=1000; norm1[3]:=1000;
 lognorm1[1]:=1000; lognorm1[2]:=1000; lognorm1[3]:=1000;

 Gd := Detect; InitGraph (Gd,Gm,'');
 if GraphResult <> grOk then Halt(1);
```

```
{ -------------- Entering Data -----------}
 Filename := 'Close.txt';
 Assign (F,filename);
 rewrite(F);
 Writeln(F,'NO    TTA     TTAmin    TTAminR   Dcmax     DX      DV        F3');
{--------------------------------}
counter :=0;
repeat
begin
 counter :=counter+1;
 repeat
  v2 := random(35); { The speed of the following vehicle}
 until v2>=20.0;
 repeat
  x:=trunc(v2-3);
  v1:=random(x); { The speed of the Leading vehicle}
 until (v2-v1)<14;
 dx:=200;       { The relative distance}
{------------------------}
 ov2 :=v2;  ox2 :=0.0;     x2 :=0.25*ov2;
 odc2:=0.0; odc21:=0.0;   odc22:=0.0;
 ov1 :=v1;   x1 := X2+dx;  ox1 :=x1-0.25*ov1;
 odc1:=0.0; odc12:=odc1;     dv := v1-v2;
 odv :=dv;   odx :=ox1-ox2; dx  := x1-x2;
 odv1:= odv;DVper1:='s';
 theta1 := (odv1*2)/(sqr((odx-odv*0.25))+ 1);
 OEST :=0.0;Eodv1:=odv;
 fdc2:=-0.5;
 {------------------------}
 threshold; drawtimer; speed_time; drawDC2; DrawEST; DrawDFD; DrawTTC; DrawF;
 { -------------- Following vehicle -----------}
 fvehicle; S:='';
 Acmax2 := 0.5+0.5*random;{Acmax2:=0.531;}
 x:=105;y:=54;W:=3;outtextxy (x,y,roundstr(Acmax2,W));
 {-----------------}
turn:=1;
if (norm1[turn]=1000)or (norm1[turn]<8) then begin
 repeat
   mean :=11.7026;std:=2.292;  TTA:= Normal (mean,std);
 until ((TTA>=8));
end
else begin
   TTA:=norm1[turn];norm1[turn]:=1000;
end;
 {-----------------}
turn1:=1;
if (lognorm1[turn1]=1000)or (lognorm1[turn1]<5) then begin
 repeat
  mean :=8.41; std:=2.1;
  TTAmin:= lognormal (mean,std);
 until ((TTAmin>=5));
end
else begin
   TTAmin:=lognorm1[turn1];lognorm1[turn1]:=1000;
end;
if TTAmin > (TTA-0.5) then TTAmin:=TTA-2;
 {-----------------}
turn1:=2;
```

```
if (lognorm1[turn1]=1000)or (lognorm1[turn1]<0.4)or(lognorm1[turn1]>1.6) then begin
 repeat
  mean :=0.86771; std:=0.342;
  FDCmax:= lognormal (mean,std);
 until ((fdcmax>=0.4)and (fdcmax<=1.6));
end
else begin
  Fdcmax:=lognorm1[turn1];
end;
Fdcmax:=-1*Fdcmax;
 {-----------------}
 x:=105;y:=78;W:=3; outtextxy (x,y,roundstr(TTA,W));
 x:=105;y:=66;W:=3; outtextxy (x,y,roundstr(TTAmin,W));

 x:=105;y:=90;W:=3;vd:=v2;outtextxy (x,y,roundstr(Vd,W));
 dc1 :=0.0;dc2 :=0.0;
 setcolor(15);
 T1:=200; W:=1; S:='200'; outtextxy (600,165,'50');S:='';
 {---------------------------------}
 VV1 :=round(160-v1*3/2);  VV2 :=round(160-v2*3/2);
 CC1 :=round(45-odc1*7.5); CC2 :=round(45-odc2*7.5);
 {---------------------------------}
 TT1 := round(380); TT2:=TT1; Ax :=3;
  x := round(550 - dx*5/2);y:=round(375 + dv*12.5);
 {---------------------------------}
 factor := 0; factor1 :=0; factor2:=0; factorth :=0; factor5 :=0;
 {---------------------------------}
 setcolor (10);line(1,200,1,375);
 getmem(P1,imagesize(1,200,1,375));
 getmem(P3,imagesize(1,200,1,375));
 getmem(P2,imagesize(1,375,1,450));
 getmem(P4,imagesize(1,375,1,450));
 getimage(1,200,1,375,P1^);
 setcolor (0);line(1,200,1,375);
 setcolor(10);line(1,375,1,450);
 getimage(1,375,1,450,P2^);
 setcolor (0);line(1,375,1,450);
{---------------------------------}
data;
{---------------------------------}
T:=0.25;F1:=0;F2:=0;TTAminF:=TTA;DCmaxF:=0;TTAF:=TTA;F3:=0;F4:=0;
rand:=random;rand1:=random;
CLDV:=-0.6;Number:=1;
OPDV:=+0.6;

repeat
 begin
 if Number<>1 then begin
  Rand1:=random; Rand:=random;
  Eodv:=0.0;F2:=0;F3:=0;
  turn:=1;
  if (norm1[turn]=1000)or (norm1[turn]<8) then begin
   repeat
    mean :=11.7026;std:=2.292;  TTA:= Normal (mean,std);
   until ((TTA>=8));
  end
  else begin
   TTA:=norm1[turn];norm1[turn]:=1000;
```

```
 end;
{-----------------}
 repeat
 turn1:=1;
 if (lognorm1[turn1]=1000)then begin
  mean :=8.41; std:=2.1;
  TTAmin:= lognormal (mean,std);
 end
 else begin
  TTAmin:=lognorm1[turn1];lognorm1[turn1]:=1000;
 end;
until ((TTAmin>=5)and(TTAmin<=(TTA-2)));
{-----------------}
 Number:=1;
 CLDV:=CLDVthresh(odx); OPDV:=OPDVthresh(odx);
end;
{-----------------}
 speed:=ov1;
 ABX:=ABXthresh(speed,rand);
 space:=ABX;
 SDX:=SDXthresh(space,speed,rand1);
 AX:=2+1.5*random;
 if ABX < (AX+1) then ABX:=AX+1;
 if (SDX-ABX)<5 then begin
  DFD := ABX+2.5;SDX:=5+ABX;
 end
 else DFD:=(SDX+ABX)/2;
 mean:=DFD;STD:=(DFD-ABX)/2;
 if DFD <= ABX then DFD :=ABX+1;
 if DFD >= SDX then DFD :=SDX-1;
 turn:=2;DFD:=normal(mean,std);
 {------------------}
 theta := (odv*2)/(sqr(odx)+ 1);
 space:= odx;Speed:= odv;
 DVper:= DVperception(space,speed,theta,theta1,DVper1);
 theta1:=theta;DVper1:=DVper;
{---------------------------------}
 if (DVper='c')and (odv>=0.0) then DVper:='s';
 if (DVper='o')and (odv<=0.0) then DVper:='s';
{---------------------------------}
 xx1 :=round(550-sdx*5/2);
 xx2 :=round(550-ABX*5/2);
 getImage(xx1,200,xx1,375,P3^);
 getImage(xx2,375,xx2,450,P4^);
 putImage(xx1,200,P1^,normalput);
 putImage(xx2,375,P2^,normalput);
{---------------------------------}
 NULL := 0.1+0.15*random;NULL1 :=-1.0*NULL;
{------------------------------------}
if DVper <>'s' then begin
 if ov1 > 0.0 then begin
  if (Eodx > ABX) then begin
   PDC :=0.0;
   if (odc12<=-0.95)and(factor2=2)  then begin factor2:=22;PDC:=odc12;end;
   if (odc12<=-0.95)and(factor2<>22) then factor2 :=2 ;
   if (odc12>= 0.5) and(factor2 =1)  then begin factor2 :=11;PDC:=odc12;end;
   if (odc12>= 0.5) and(factor2<>11) then factor2 :=1 ;
   if (odc12 < 0.5) and(odc12>-0.95) then begin factor2 :=0 ;end;
```

```
    end;
    if (Eodx <= ABX) then begin
     PDC :=0.0;
     if (odc12<=-0.5)and(factor2=2)  then begin factor2:=22;PDC:=odc12;end;
     if (odc12<=-0.5)and(factor2<>22) then factor2 :=2 ;
     if (odc12>= 0.25) and(factor2 =1)  then begin factor2 :=11;PDC:=odc12;end;
     if (odc12>= 0.25) and(factor2<>11) then factor2 :=1 ;
     if (odc12 < 0.25) and(odc12>-0.5) then begin factor2 :=0 ;end;
    end;
   end
   else begin
    PDC:=0.0;factor2:=0;
   end;
   Eodx:=odx+odv*0.5+(-odc22)*0.125;
   if (odv<=-0.5)and (odx>ABX)and(DVper='c') then begin
    TTC:= -odx/odv;
    est:=(-0.18+0.767*TTC);
    if (Est < 1)or(EST > TTC) then Est:=TTC;
    if (EST < 0.4*TTc) then Est :=0.4*TTC;
    Eodv :=-odx/EST;
    if (odx>SDX)and(OEST<20)and((factor2=11)or(factor2=22))and(ov1>5)then
     PDC := odc22 -((EST-OEST+0.25)*odv1/((EST+0.125)*0.25));
    OEST :=EST;odv1:=odv;
  end
  else begin
   EST:=100;Eodv:=odv;OEST :=EST;odv1:=odv;
  end;
 end
 else begin
  if (ov1<=0.0) then begin PDC :=0.0;factor2:=0;end;
  Eodx:=odx+odv*0.5-odc2*0.125;
  EST:=100;Eodv:=odv;odv1:=odv;
 end;
{-----------------------------------}
 if (DVper='c')and(Eodx>SDX) then begin
   factorth:=0;
   if dv < 0.0 then TTC:= -dx/dv else TTC:=100;
   if TTAminF > TTC then TTAminF:=TTC;
   if ov1<=0.0 then begin
    if (TTC <= TTA)or(factor<>0) then begin
      factor:=1; dc2 := 0.65*sqr(Eodv) / (ABX-Eodx);
      if dc2 > NULL1 then dc2:= null;
    end
    else begin dc2:=NULL;end;
   end
   else begin
    if (factor2=22)and(Eodv<0.0)and (ov2>0.0) then begin
     DTC:=(Eodx-AX)/ov2;if DTC <0.0 then DTC:=0.0;
     DCmax := odc12*SQR(ov2)/(SQR(Eodv1+ov2)-(2*odc12*ov2*DTC));
     if DTC=0.0 then DCmax:=dcmax+NULL1;
     If DCmax >(-0.5*sqr(Eodv)/(Eodx-(ABX/2))) then DCmax :=-0.5*sqr(Eodv)/(Eodx-(ABX/2));
     if DCmax > NULL1 then DCmax :=0.0;
    end
    else DCmax:=0.0;
    case factor of
    3,6:begin
       factor := 6;
       if (TTC < TTA)or(F2<>0.0) then begin
```

```
  F2:=1;dc2:=0.45*sqr(Eodv)/(ABX-odx);
  if dc2 > NULL1 then dc2:=NULL;
  end else dc2:=0.5*Acmax2;
case factor2 of
  22: begin
     if dc2 < 0.0 then begin
      dc2 := PDC +dc2;
      if dc2 < Dcmax then dc2 := Dcmax;
     end;
    end;
  11: begin
     if (dc2 <NULL) and (dc2 > NULL1) then dc2:=NULL1;
    end;
  end;
  end;
{------------------------}
  0:begin
   if (TTC>TTA)and (F2=0) then begin
    if (Eodx<(2*ABX))or(Eodx <=50) then begin
     case factor2 of
      22:begin
        dc2:=NULL;
       end;
      11:begin dc2:=0.5*Acmax2;end;
     1,2,0:begin dc2:=NULL;  end;
     end;
    end
    else begin dc2:=0.5*Acmax2;end;
   end
   else begin
    factor:=1;
    if F2 =0 then begin F2:=1;DXF:=dx;DVF:=dv;end;
    F1:=0.45*exp(((TTC-TTA)/(TTamin-TTA))-1);
    if (EST <>0.0)and (Eodv<0.0)then begin
     dc2:=F1*Eodv/(EST-0.5);
     if dc2 > NULL1 then dc2:=NULL;
    end
    else begin
     dc2:=NULL;
    end;
    if TTC <=TTAmin then F3:=1;
    if ((Eodx<(2*ABX))or(f3<>0))and(Eodv<0.0) then begin
     if dc2>(0.45*sqr(Eodv)/(ABX-Eodx)) then
      dc2:=(0.45*sqr(Eodv)/(ABX-Eodx));
    end;
    case factor2 of
     2: begin dc2:= dc2 +NULL1; end;
     22: begin
        if ((Eodx<(2*ABX))or(f3<>0))and(Eodv<0.0) then begin
         dc2 :=PDC +dc2;if dc2 < Dcmax then dc2 := Dcmax;
        end;
       end;
     11: begin
        if (dc2 <NULL) and (dc2 > NULL1) then dc2 :=null1;
       end;
    end;
   end;
  end;
```

```
{-----------------------}
      1:begin
       if (TTC>TTA)and (F2=0) then begin
        F3:=0;
        if (Eodx<(2*ABX))or(f3<>0) then begin
         case factor2 of
           22:begin
              dc2:=NULL;
             end;
           11:begin dc2:=0.5*Acmax2;end;
         1,2,0:begin dc2:=NULL;  end;
         end;
       end
       else begin dc2:=0.5*Acmax2;end;
      end
      else begin
       if F2 =0 then F2:=1;
       F1:=0.45*exp(((TTC-TTA)/(TTamin-TTA))-1);
       if (EST <>0.0)and (Eodv<0.0)then begin
        dc2:=F1*Eodv/(EST-0.5);
        if dc2 > NULL1 then dc2:=NULL;
       end
       else begin
        dc2:=NULL;
       end;
       if TTC <=TTAmin then F3:=1;
       if ((Eodx<(2*ABX))or(f3<>0))and(Eodv<0.0) then begin
        if dc2 > (0.45*sqr(Eodv)/(ABX-Eodx)) then
         dc2:=(0.45*sqr(Eodv)/(ABX-Eodx));
       end;
       if dc2> NULL1 then begin F2:=0;F3:=0;end;
       case factor2 of
        22: begin
             if ((Eodx<(2*ABX))or(f3<>0))and(Eodv<0.0) then begin
              dc2 :=PDC+dc2;if dc2<Dcmax then dc2:=Dcmax;
             end;
            end;
        11: begin
             {dc2 :=dc2+PDC;}
             if (dc2<NULL)and(dc2>NULL1) then dc2:=null1;
            end;
        2: begin dc2:= dc2 +NULL1; end;
       end;
       if dc2>NULL1 then dc2:=NULL;
      end;
     end;
{-----------------------}
   end;
  end;
 if DCmaxF > DC2 then DCmaxF := dc2;
 end;
{-----------------------------------}
if (DVper='s')and (Eodx > SDX)  then begin
 factorth:=0; F2:=0;F3:=0;
 if ov1<=0.0 then begin
  if ((odx-AX) > 75)  then begin
   factor :=0;dc2:= Acmax2;
  end
```

```
   else dc2:=NULL;
  end
  else begin
   if ((odx-AX) > 75)  then begin
    factor :=0;dc2:= Acmax2;
   end
   else begin
    if factor=1 then factor:=0;
    if (factor<>0)and (factor<>6) then begin
     rand1:=random;rand:=random;
     factor :=6;
    end;
    if factor = 0 then dc2 := acmax2;
    if factor = 6 then dc2 := 0.5*acmax2;
   end;
  end;
 end;
{---------------------------------}
 if (DVper='o')and(Eodx>SDX)  then begin
  factorth:=0;
  if ((Eodx-AX)>(1.2*SDX))  then begin
    factor :=0;dc2:= Acmax2;
  end
  else begin
   case factor of
   5,3,6:begin
          if factor<>6 then begin
           factor :=6;
           rand1:=random;rand:=random;
          end ;
          dc2:=0.5*Acmax2;
          If (odx >(2*ABX))and(odx>75) then factor:=0;
         end;
    1,0:begin
          dc2:=ACmax2;
         end;
  end;
 end;
end;
{---------------------------------}
 if(Eodx<=SDX)and(Eodx>ABX)and(DVper='c') then begin
  if ov1<=0.0 then begin
   dc2 := 0.65*sqr(Eodv)/(AX-Eodx);
   if dc2 > NULL1 then dc2:= NULL;
  end
  else begin
   if (factor2=22)and(ov2 >0.0)then begin
    DTC:=(Eodx-AX)/ov2;if DTC<0.0 then DTC:=0.0;
    if (Eodv+ov2)<=0.0 then
     DCmax:=0.5*sqr(Eodv)/(AX-Eodx)
    else
     DCmax := odc12*SQR(ov2)/(SQR(Eodv1+ov2)-(2*odc12*ov2*DTC));
     if (ABX/2)> AX then begin
      if Dcmax > (0.5*sqr(Eodv)/((ABX/2)-odx)) then Dcmax:=0.5*sqr(Eodv)/((ABX/2)-odx);
     end
     else begin
      if Dcmax > (0.5*sqr(Eodv)/(AX-odx)) then Dcmax:=0.5*sqr(Eodv)/(AX-odx);
     end;
```

```
    if DCmax>NULL1 then DCmax:=NULL1;
  end
  else Dcmax :=NULL1;
  case factor of
 1,0,6: begin
       factorth:=1;factor:=6;
       dc2:= -0.5*sqr(Eodv)/(odx-ABX);
       if dc2 < 0.5*Eodv then dc2 := 0.5*Eodv;
       if dc2 > NULL1 then begin
        if Eodx > DFD then dc2:=NULL*random else dc2:=NULL1;
       end;
       case factor2 of
        11:begin
           dc2:=dc2+PDC;
           if (dc2 <NULL) and (dc2 > NULL1) then dc2:=NULL1;
          end;
        22:begin
           dc2:=dc2+PDC;
           if (dc2<Dcmax) and((odx/ov2)>1.3)then dc2 := Dcmax;
           if dc2 > NUl1 then dc2:=NULL1;
          end;
       end;
      end;
 5,3 : begin
       if dv <= CLDV then begin
        Space:=Eodx-AX;OPDV:=OPDVthresh(Space);
        factorth:=1;
        dc2:= -0.5*sqr(Eodv)/(odx-ABX);
        if dc2 < Eodv/2 then dc2 := Eodv/2;
        if dc2 > NULL1 then dc2:=NULL1;
       end
       else begin
        case factorth of
         1:begin
           dc2:= -0.5*sqr(Eodv)/(odx-ABX);
           if dc2 < Eodv/2 then dc2 := Eodv/2;
           if dc2 > NULL1 then dc2:=NULL1;
          end;
       2,0:begin
           factorth:=0;dc2:=NULL;
          end;
        end;
       end;
       case factor2 of
        22:begin
           if factorth=1 then begin
             dc2:=PDC+dc2;
             if (dc2<Dcmax) then dc2:=Dcmax+2*NULL1;
           end;
          end;
       end;
      end;
 end;
end;
end;
{------------------------------}
if(Eodx>=ABX)and(Eodx<=SDX)and(DVper='o')then begin
 if (factor2=22) then begin
```

```
    if ov2 >0.0 then begin
     DTC :=(Eodx-Ax)/ov2;if DTC<0.0 then DTC:=0.0;
     DCmax:=PDC*sqr(ov2)/(sqr(ov1)-(2*PDC*ov2*DTC));
     if DCmax>NULL1 then DCmax:=NULL1;
    end else  DCmax :=0.0
  end;
  case factor of
  1,5,6,3:begin
        factor := 3;
        if (dv >= OPDV) then begin
         space := Eodx-AX; CLDV:=CLDVthresh(Space);
         factorth:=2;dc2:=0.5*Eodv;
         if factor2 =11 then begin
          if odx >DFD then
           dc2:=0.5*Acmax2
          else begin
           dc2:=PDC+dc2+2*NULL;
           if dc2 < odc2 then dc2:=odc2;
          end;
         end;
         if dc2 < NULL then dc2:=NULL;
        end
        else begin
         case factorth of
         2:begin
           dc2:=0.5*Eodv;
           if dc2<odc2 then dc2:=odc2;
           if dc2 < NULL then dc2:=NULL;
           case factor2 of
           11:begin
             dc2:=PDC+dc2+2*NULL;
             if dc2 < odc2 then dc2:=odc2;
            end;
           end;
          end;
      1,0:begin
            factorth:=0;dc2:=NULL1;
           end;
         end;
        end;
       end;
     0:begin
       dc2:=0.5*ACmax2;
       if factor2=22 then begin dc2:=NULL1;factor:=3;end;
      end;
 end;
end;
      {-----------------}
if(Eodx<=SDX)and(Eodx>=ABX)and(DVper='s')then begin
 factor:=3;
 if ov1=0.0 then begin
  dc2:= NULL;
 end
 else begin
  case factorth of
   1:begin
     if PDC <>0.0 then dc2:=odc2 else dc2:=NULL1;
    end;
```

```
   2:begin
     if PDC <>0.0 then dc2:=odc2 else dc2:=NULL;
    end;
   0:begin
     dc2:=odc2;
    end;
  end;
 end;
end;
{--------------------------------}
 if (Eodx < ABX) then begin
  factorth:=0;
  if ov1<=0.0 then begin
   if Eodv <0.0 then begin
     if (v2/0.25)>(sqr(v2)/(dx-1))then
       dc2:=-v2/0.25
     else dc2:=sqr(v2)/(1-dx);
   end
   else dc2:=NULL*Random;
  end
  else begin
   factor :=5;
   if Eodv<0.0 then begin
    if (factor2 =22) then begin
     if ov2<>0.0 then begin
      DTC :=(Eodx-Ax)/ov2;
      if DTC < 0.25 then
       Dcmax := -1*ABS(dv/0.25)+PDC
      else begin
       Dcmax:=PDC*sqr(ov2)/(SQR(Eodv1+ov2)-(2*PDC*ov2*DTC));
       if Eodx > ((ABX+AX)/2)then
        Dcmax := Dcmax +NULL1+(2*(ABX-Eodx)*NULL1/(ABX-AX))
       else  Dcmax := Dcmax +2*NULL1;
       if (factor2=2)then DCmax:=DCmax+NULL1;
        if dcmax>(sqr(Eodv)/(Ax-Eodx))then dcmax:=sqr(Eodv)/(Ax-Eodx);
      end;
      if DCmax > -0.6 then DCmax :=-0.6;
    end
    else Dcmax:=0.0;
   end;
   dc2:=-0.5;
   if AX < Eodx then begin
    if (sqr(Eodv)/(Ax-Eodx))< dc2 then begin
     dc2:=sqr(Eodv)/(Ax-Eodx);
     if dc2<Eodv then dc2 :=Eodv;
    end;
    if odx<((AX+ABX)/2) then dc2:=dc2+NULL1;
   end
   else begin
    dc2:=dc2+NULL1;
    if (Eodv/0.25)<dc2 then dc2:=Eodv/0.25;
   end;
   case factor2 of
     2: begin
         dc2 := dc2+Null1;
        end;
    22: begin
         dc2:=dc2+PDC;
```

```
          if dc2<DCmax then dc2:=DCmax;
          end;
     end;
    end;
  {-------------------------}
    if Eodv >=0.0 then begin
     dc2:=NULL1;
     if factor2=22 then begin
      dc2 := PDC+NULL1;
      if Eodx<((AX+ABX)/2)then begin
       if dc2 > -0.6 then dc2 :=-0.6 else dc2:=dc2+NUll1;
      end;
     end;
     if factor2=11 then begin
      if Eodv >1 then dc2 := PDC else dc2:=PDC-0.3;
     end;
    end;
  {-------------------------}
   end;
  end;
  {-------------------------}
  TTC1:=-dx/dv;
  S:= '+'; if TTC1<0 then s:='-';
  str(round(int(ABS(TTC1))),S1); S := S+S1+'.';
  str(round(frac(ABS(TTC1))*100),S1);
  val(S1,hel,code); if hel<10 then S:=S+'0';
  S:=S+S1;
  setcolor(0); outtextxy(247,118,'ÛÛÛÛÛÛÛÛ'); setcolor (15);
  outtextxy(247,118,S); setcolor (9);
 {------- UPDATING ------------}
if dc2 > Acmax2 then dc2 := Acmax2;
 odc12 :=odc1; odc1:=dc1; ov1 :=v1; ox1 :=x1;
 x1:=x1+0.25*v1+0.5*dc1*(0.0625);
 v1:=v1+0.25*dc1;
 ox2:=x2; ov2:=v2; x2:=x2+0.25*v2+0.5*odc2*(0.0625); v2:=v2+0.25*odc2;
 VL := v2 + dc2*0.25;
 if (VL> vd) then begin
  dc2 := 4*(Vd-v2);
  if odx > SDX then factor:=0;
 end;
 odc22 :=odc21;odc21 :=odc2;
 odc2:=dc2; {OEst := Est;}
 {-------------------}
 odv:=dv;odx:=dx;dv :=v1-v2; dx:=x1-x2; moveto(x,y);
 x:=round(550-dx*5/2); y:=round(375+dv*12.5);
 setcolor (9); if odc2 <0 then setcolor(10);lineto (x,y);
 {-------------------}
 str(round(int(T)),S1); S := S1+'.'; str(round(frac(T)*100),S1); S:=S+S1;
 setcolor(0); outtextxy(247,46,'ÛÛÛÛÛÛÛÛ'); setcolor (15);
 outtextxy(247,46,S); setcolor (9);
{-------------------}
S:= '+'; if EODV<0 then S:='-';
str(round(int(abs(EODV))),S1); S := S+S1+'.';
str(round(frac(abs(EODV))*100),S1);
S:=S+S1;
setcolor(0); outtextxy(247,94,'ÛÛÛÛÛÛÛÛ'); setcolor (15);
outtextxy(247,94,S); setcolor (9);
{-------------------}
```

```
str((Factor),S);setcolor(0); outtextxy(142,118,'Û'); setcolor (15);
outtextxy(142,118,S); setcolor (9);
str((Factor1),S);setcolor(0); outtextxy(142,142,'Û'); setcolor (15);
outtextxy(142,142,S); setcolor (9);
str((Factor2),S);setcolor(0); outtextxy(142,166,'ÛÛ'); setcolor (15);
 outtextxy(142,166,S); setcolor (9);
{-------------------}
str(round(int(DFD)),S1); S := S1+'.';
str(round(frac(DFD)*100),S1); S:=S+S1;
setcolor(0); outtextxy(247,142,'ÛÛÛÛÛ'); setcolor (15);
outtextxy(247,142,S); setcolor (9);
{-------------------}
S:= '+'; if dc2<0 then S:='-';
str(round(int(abs(dc2))),S1);
S := S+S1+'.';
str(round(frac(abs(
dc2))*100),S1);
S:=S+S1;
setcolor(0); outtextxy(247,70,'ÛÛÛÛÛ'); setcolor (15);
outtextxy(247,70,S); setcolor (9);
{-------------------}
moveto(TT1,VV1);VV1 :=round(180-v1*3/2);
TT1 := round(380+T*240/(t1)); setcolor(15); lineto(TT1,VV1);
moveto(TT2,VV2);
TT2:=TT1;VV2 :=round(180-v2*3/2); setcolor(13);lineto(TT2,VV2);
Moveto(TT1,CC1);CC1 :=round(45-dc1*7.5);setcolor(15);lineto(TT1,CC1);
Moveto(TT1,CC2);CC2 :=round(45-dc2*7.5);setcolor(13);lineto(TT1,CC2);
{-------------------}
T:=T+0.25;
{-------------------
S:=readkey;
if S=chr(27) then goto 100;
{-------------------}
key := keypressed;S:='';
if key = true then begin
 S:=readkey; key := false;
 if S=chr(27) then goto 100;
 S:=readkey;
end;
{-------------------}
putimage(xx1,200,P3^,normalput);
putimage(xx2,375,P4^,normalput);
{-------------------}
{ delay (10);}
end;
until T > 100;
savedata;
putImage(xx1,200,P1^,normalput);
putImage(xx2,375,P2^,normalput);
{----------------------------------------------------}
ClearViewPort;
end;
until counter >100;
100: close(f);
closegraph;
clrscr;
end.
```

```
unit Cld5;
{ The CLD5 is the same as the CLD4
  the only difference is in the save data procedure }
interface
uses crt,graph;
var
  Gd,Gm,x,y,W,xx1,xx2,vv1,vv2,tt1,tt2,Dvy1,Dvy2,Dvy3,CC1,CC2,counter,F3 :integer;
  DFD,ABX,SDX,ttc,est,sdv,i,T,t1,Tx,NULL1,Ax,TTC2,F1,F2,DCmaxF,DXF  :single;
  a1,theta1,NULL,CLDV,OPDV,SDV1,Vdfd,TTAF,TTAminF,DVF,dcmax,Eodx     :single;
  x1,v1,ox1,ov1,dc1,odc1,odc12,x2,v2,ox2,ov2,dc2,odc2,Eodv,DXtta    :single;
  acmax2,TTA,TTAmin,FDC2,dcmax2,vd,VL,dv,dx,odx,odv,theta,EDVtta    :single;
  odc21,odv1,odc22,PDC,ttc1,EST1,OEST,DTC,dax,mean,std,fdcmax,Eodva :single;
  rand,speed,space,rand1,Eodv1,v3                          :single;
  S ,S1                                        :string;
  P1,P2,P3,P4                                   :pointer;
  hel,code,F4,number                             :integer;
  C,DVper,DVper1                                 :char;
  factor, factor1, factor2, factor3,factorth,factor5,op,turn,turn1 :byte;
  F,DF                                         : text;
  Filename                                      : string;
  DVclose       : array[1..5] of single;
  DVopen        : array[1..3] of single;
  lognorm1      : array[1..3]of single;
  norm1        : array[1..4]of single;
procedure data;
function  power(var N:single;M:single):single;
procedure beep;
function  Lognormal (var mean,std:single):single;
function  Normal (var mean,std : single):single;
function  roundstr (N : single;  M:integer ): string;
procedure speed_Time;
procedure threshold;
procedure fvehicle ;
procedure Lvehicle ;
procedure drawtimer;
procedure drawDC2;
procedure drawEST;
procedure drawTTC;
procedure drawDFD;
procedure drawF;
procedure savedata;
function DVperception(var space,Speed,theta,theta1:single; DVper1:char):Char;
function CLDVThresh(var space:single ):single;
function OPDVThresh(var space:single ):single;
function ABXThresh(var speed,rand:single ):single;
function SDXThresh(var space,speed,rand:single ):single;

{procedure Fdcmaxtest;}
implementation

function power(var N:single;M:single):single;
begin
 power := EXP(M*LN(N));
end;
{------------------------------------------}
procedure beep;
 begin
  sound(220);
```

```
  delay(200);
  nosound;
 end;
 {-------------------------------------------}
 function CLDVThresh(var space:single ):single;
 var
 per,DVmin,Y,x,Z1,Z2 :single;
 begin
  per:=100*random; DVmin:=-0.01*Space;
  if PER >= 99 then begin CLDVThresh:=-2.5;Exit;end;
  if space <=20 then begin
   y:=-16.1*(sqr(Dvmin)*DVmin)-96.01*Sqr(DVmin)-191.2*Dvmin-28.733;
   if per <= Y then begin
    CLDVThresh:=DVmin;Exit;
   end
   else begin
    Z2:=-2.5;Z1:=DVmin;
    X:=(Z1+Z2)/2;
    y:=-16.1*(sqr(X)*X)-96.01*Sqr(X)-191.2*X-28.733;
    while ABS(y-per)>0.05 do begin
     if per > Y then z1:=X else z2:=x;
     X:=(Z1+Z2)/2;
     y:=-16.1*(sqr(X)*X)-96.01*Sqr(X)-191.2*X-28.733;
    end;
    CLDVThresh:=X;Exit;
   end;
  end
  else begin
   y:=-6.6715*(sqr(Dvmin)*DVmin)-54.11*Sqr(DVmin)-151.73*Dvmin-45.325;
   if per <= Y then begin
    CLDVThresh:=DVmin;Exit;
   end
   else begin
    Z2:=-2.5;Z1:=DVmin;
    X:=(Z1+Z2)/2;
    y:=-6.6715*(sqr(X)*X)-54.11*Sqr(X)-151.73*X-45.325;
    while ABS(y-per)>0.05 do begin
     if per > Y then z1:=X else z2:=x;
     X:=(Z1+Z2)/2;
     y:=-6.6715*(sqr(X)*X)-54.11*Sqr(X)-151.73*X-45.325;
    end;
    CLDVThresh:=X;Exit;
   end;
  end;
end;{ end of function CLDVThresh
---------------------------------}
function OPDVThresh(var space:single ):single;
var
per,DVmin,Y,x,Z1,Z2 :single;
begin
 per:=100*random; DVmin:=0.01*Space;
 if PER >= 99 then begin OPDVThresh:=2.5;Exit;end;
 if space <=20 then begin
  y:=10.863*(sqr(Dvmin)*DVmin)-73.993*Sqr(DVmin)+167.79*Dvmin-27.066;
  if per <= Y then begin
   OPDVThresh:=DVmin;
  end
  else begin
```

```
  Z2:=2.5;Z1:=DVmin;
  X:=(Z1+Z2)/2;
  y:=10.863*(sqr(X)*X)-73.993*Sqr(X)+167.79*X-27.066;
  while ABS(y-per)>0.05 do begin
   if per > Y then z1:=X else z2:=x;
   X:=(Z1+Z2)/2;
   y:=10.863*(sqr(X)*X)-73.993*Sqr(X)+167.79*X-27.066;
  end;
  OPDVThresh:=X;
 end;
end
else begin
 y:=3.744*(sqr(Dvmin)*DVmin)-39.95*Sqr(DVmin)+131.83*Dvmin-38.607;
 if per <= Y then begin
  OPDVThresh:=DVmin;
 end
 else begin
  Z2:=2.5;Z1:=DVmin;
  X:=(Z1+Z2)/2;
  y:=3.744*(sqr(X)*X)-39.95*Sqr(X)+131.83*X-38.607;
  while ABS(y-per)>0.05 do begin
   if per > Y then z1:=X else z2:=x;
   X:=(Z1+Z2)/2;
   y:=3.744*(sqr(X)*X)-39.95*Sqr(X)+131.83*X-38.607;
  end;
  OPDVThresh:=X;
 end;
end;
end;{ end of function CLDVThresh
----------------------------------}
function DVperception(var space,Speed,theta,theta1:single; DVper1:char):char;
var
 per,TTC   :single;
 N1        :integer;
begin
 TTC:=Speed/Space;
 if      (TTC < -0.11)       then begin DVperception:='c';Exit;end;
 if (TTC >-0.01) and (TTC <0.01) then begin DVperception:='s';Exit;end;
 if      (TTC >0.09)         then begin DVperception:='o';Exit;end;
 if (theta1 > theta) and (DVper1='c')and (space<100)and(TTC<-0.01) then
 begin DVperception:='c';Exit;end;
 if (theta1 < theta) and (DVper1='o')and (space<100)and(TTC>0.01) then
 begin DVperception:='o';Exit;end;
 {----------------------------------}
if TTC < -0.01 then begin
 N1:=0;per:=-0.01;repeat n1:=n1+1; until (TTC>(per-(N1*0.02)))or(n1=5);
 per :=100*random;
 if per <=DVClose[n1] then begin
  DVperception:='c';Exit;
 end
 else begin DVperception:='s';Exit;end;
end
else begin
 if TTC >0.01 then begin
  N1:=0;per:=0.01;repeat n1:=n1+1; until (TTC<(per+(N1*0.02)))or(n1=3);
  per :=100*random;
  if per <=DVopen[n1] then begin
   DVperception:='o';Exit;
```

```
  end
  else begin DVperception:='s';Exit;end;
 end;
 end;
end;{ End of function DVper
----------------------------------}
function ABXThresh(var speed,rand:single ):single;
var
ABXmin,ABXmax,V,BX  :single;
ABXper  :array[1..5] of single;
begin
 V:=Speed*3.6;
 if rand >0.65 then rand:=(rand-0.05);
 if V <= 55 then begin
  ABXmin:=-0.001159*SQR(V)+0.174643*V+2.276; ABXmax:=(-7.1772E-05)*sqr(V)*V+0.01135*sqr(V)-
0.129646*V+9.7725;
 end
 else begin
  ABXmin:=0.0198*V+6.3771; ABXmax:=0.2518*V+9.634;
 end;
 ABXper[1]:=(((7.035E-06)*sqr(V)*Sqr(V))-((2.3742E-03)*Sqr(V)*V)+0.28992*sqr(V)-
15.198*V+297.746)/100;
 if V >125 then begin
  ABXper[1]:=0.07;
  ABXper[2]:=(-((555.36E-06)*sqr(V)*V)+0.1571*sqr(V)-14.3114*V+450.14)/100;
  if ABXper[2]< 0 then ABXper[2] :=0.0;
  if ABXper[2]< ABXper[1] then ABXper[1]:=ABXper[2];
 end
 else begin
  if ABXper[1] >1 then ABXper[1]:=1;
 end;
 if rand < ABXper[1] then begin
  BX := ABXmin+((10-ABXmin)* RAND / ABXper[1]);
  {BX := ABXmin+random(10-round(ABXmin));}
 end
 else begin
  ABXper[2]:=(-((555.36E-06)*sqr(V)*V)+0.1571*sqr(V)-14.3114*V+450.14)/100;
  if ABXper[2] >1 then ABXper[2]:=1;
  if ABXper[2]< 0 then ABXper[2] :=0.0;
  if rand < ABXper[2] then begin
   if ABXper[2]<1 then
    BX := 10+(5*(RAND-ABXper[1])/(ABXper[2]-ABXper[1]))
   else begin
    if ABXmax <=15 then
     BX := 10+((ABXmax-10)*(RAND-ABXper[1])/(1-ABXper[1]))
    else BX := 10+(5*(RAND-ABXper[1])/(ABXper[2]-ABXper[1]));
   end;
  end
  else begin
   ABXper[3]:=((-(368.84E-06)*sqr(V)*V)+0.11646*sqr(V)-11.7954*V+442.64)/100;
   if ABXper[3] >1 then ABXper[3]:=1;
   if rand < ABXper[3] then begin
    if ABXper[3]<1 then
     BX := 15+(5*(RAND-ABXper[2])/(ABXper[3]-ABXper[2]))
    else begin
     if ABXmax <=20 then
      BX := 15+((ABXmax-15)*(RAND-ABXper[2])/(1-ABXper[2]))
     else BX := 15+(5*(RAND-ABXper[2])/(ABXper[3]-ABXper[2]));
```

```
     end;
    end
    else begin
     ABXper[4]:=((20.822E-03)*sqr(V)-4.2138*V+290.08)/100;
     if ABXper[4] >1 then ABXper[4]:=1;
     if rand < ABXper[4] then begin
      if ABXper[4]<1 then
       BX := 20+(5*(RAND-ABXper[3])/(ABXper[4]-ABXper[3]))
      else begin
       if ABXmax <=25 then
        BX := 20+((ABXmax-20)*(RAND-ABXper[3])/(1-ABXper[3]))
       else BX := 20+(5*(RAND-ABXper[3])/(ABXper[4]-ABXper[3]));
      end;
     end
     else begin
      ABXper[5]:=((14.521E-03)*sqr(V)-3.0708*V+250.92)/100;
      if ABXper[5] >1 then ABXper[5]:=1;
      if rand < ABXper[5] then begin
       if ABXper[5]<1 then
        BX := 25+(5*(RAND-ABXper[4])/(ABXper[5]-ABXper[4]))
       else begin
        if ABXmax <=30 then
         BX := 25+((ABXmax-25)*(RAND-ABXper[4])/(1-ABXper[4]))
        else BX := 25+(5*(RAND-ABXper[4])/(ABXper[5]-ABXper[4]));
       end;
      end
      else begin
       BX := 30+((ABXmax-30)*(RAND-ABXper[5])/(1-ABXper[5]));
      end;
     end;
    end;
   end;
  end;
  if BX< ABXmin then BX := ABXmin;
  if BX> ABXmax then BX := ABXmax;
  ABXThresh:=BX;
 end;{ End of function ABXThresh
 -----------------------------------}
 function SDXThresh(var space,speed,rand:single ):single;
 var
  SDXmin,SDXmax,SDXpermin,SDXpermax,V,SDX    :single;
  SDXper                    : array[1..7] of single;
 begin
  V:=Speed*3.6;
 if V <=55 then begin
  SDXmin:=(-1.2582E-03)*sqr(V)+0.221909*V+4.0235;
  SDXmax:=(-2.3317E-03)*sqr(v)+0.719573*V+2.5486;
  if SDXmax < SDXmin then SDXmax := SDXmin;
 end
 else begin
  SDXmin:=0.0458*V+8.6894; SDXmax:=0.3931*V+11.195;
 end;
  SDXpermin:=1.139;  SDXpermax:=(104.96E-06)*sqr(V)-(8.0921E-03)*V+2.0991;
  SDXper[1]:=(-0.06*V+19.68)/100;
  if SDXper[1] <0.0 then SDXper[1]:=0.0;
  if rand < SDXper[1] then
   SDX := SDXpermin+((1.25-SDXpermin)* RAND / SDXper[1])
  else begin
```

```
 SDXper[2]:=(-0.17*V+63.2)/100;
 if SDXper[2] <0.0 then SDXper[2]:=0.0;
 if rand < SDXper[2] then begin
  if SDXper[2] <1 then
   SDX := 1.25+(0.25*(RAND-SDXper[1])/(SDXper[2]-SDXper[1]))
  else
   SDX := 1.25+((SDXpermax-1.25)*(RAND-SDXper[1])/(1-SDXper[1]));
 end
 else begin
  SDXper[3]:=(-0.11*V+84.55)/100;
  if SDXper[3] <0.0 then SDXper[3]:=0.0;
  if rand < SDXper[3] then begin
   if SDXper[3] <1 then
    SDX := 1.5+(0.25*(RAND-SDXper[2])/(SDXper[3]-SDXper[2]))
   else
    SDX := 1.5+((SDXpermax-1.5)*(RAND-SDXper[2])/(1-SDXper[2]));
  end
  else begin
   SDXper[4]:=(-0.16*V+102.62)/100;
   if SDXper[4] <0.0 then SDXper[4]:=0.0;
   if rand < SDXper[4] then begin
    if SDXper[4] <1 then
     SDX := 1.75+(0.25*(RAND-SDXper[3])/(SDXper[4]-SDXper[3]))
    else
     SDX := 1.75+((SDXpermax-1.75)*(RAND-SDXper[3])/(1-SDXper[3]));
   end
   else begin
    SDXper[5]:=(-0.37*V+133.4)/100;
    if SDXper[5] <0.0 then SDXper[5]:=0.0;
    if rand < SDXper[5] then begin
     if SDXper[5] <1 then
      SDX := 2+(0.25*(RAND-SDXper[4])/(SDXper[5]-SDXper[4]))
     else begin
      if SDXpermax < 2.25 then
       SDX := 2+((SDXpermax-2)*(RAND-SDXper[4])/(1-SDXper[4]))
      else SDX := 2+(0.25*(RAND-SDXper[4])/(SDXper[5]-SDXper[4]));
     end;
    end
    else begin
     SDXper[6]:=(-0.21*V+121.78)/100;
     if SDXper[6] <0.0 then SDXper[6]:=0.0;
     if rand < SDXper[6] then begin
      if SDXper[6] <1 then
       SDX := 2.25+(0.25*(RAND-SDXper[5])/(SDXper[6]-SDXper[5]))
      else begin
       if SDXpermax < 2.5 then
        SDX := 2.25+((SDXpermax-2.25)*(RAND-SDXper[5])/(1-SDXper[5]))
       else SDX := 2.25+(0.25*(RAND-SDXper[5])/(SDXper[6]-SDXper[5]));
      end;
     end
     else begin
       SDX := 2.5+((SDXpermax-2.5)*(RAND-SDXper[6])/(1-SDXper[6]));
     end;
    end;
   end;
  end;
 end;
end;
```

```
SDX := Space*SDX;
if SDX < SDXmin then SDX :=SDXmin;
if SDX > SDXmax then begin
 if SDXmax > (1.139*Space) then SDX := SDXmax else SDX:=1.139*Space;
end;
SDXThresh:=SDX;
end;{ End of function SDXThresh
----------------------------------}
procedure data;
begin
{--------------------------------}
 DVClose[1]:=39.4;DVClose[2]:=63.4;DVClose[3]:=80.7;
 DVClose[4]:=92;  DVClose[5]:=97.9;
 DVOPen[1]:=50.1;  DVOPen[2]:=71.6;  DVOPen[3]:=86.2;
end;
{------------------------------------------}
function Lognormal(var mean,std:single ):single;
var Y1,X1,r1,r2,N1,N2 : single;
begin
 if mean=0.0 then readln(op);
 Y1:=ln(sqr(std/mean)+1);
 X1:=ln(mean)-Y1/2;
 repeat
  repeat r1:=random;r2:=random; until (r1<>0.0)and(r2<>0.0);
  N1:=sqrt(-2*Y1*ln(r1))*cos(2*pi*r2);
  N2:=sqrt(-2*Y1*ln(r1))*sin(2*pi*r2);
 until ((ABS(N1)<(sqrt(Y1)*2))and(ABS(N2)<(Sqrt(Y1)*2)));
 Lognormal:=exp(X1+N1);
 case turn1 of
  1: lognorm1[1]:=exp(X1+N2);
  2: lognorm1[2]:=exp(X1+N2);
  3: lognorm1[3]:=exp(X1+N2);
 end;
end;
{-------------------------------------------}
function Normal(var mean,std : single):single;
var r1,r2,N1,N2 : single;
begin
 repeat
  repeat r1:=random;r2:=random; until (r1<>0.0)and(r2<>0.0);
  N1:=sqrt(-2*sqr(std)*ln(r1))*cos(2*pi*r2);
  N2:=sqrt(-2*sqr(std)*ln(r1))*sin(2*pi*r2);
 until ((ABS(N1)<(2*std))and(ABS(N2)<(2*std)));
 Normal:=mean+N1;
 case turn of
  1: norm1[1]:=mean + N2;
  2: norm1[2]:=mean + N2;
  3: norm1[3]:=mean + N2;
  7: norm1[4]:=mean + N2;
 end;
end;
{------------------------------------------}
function  roundstr (N : single;  M:integer ): string;
var
S1,S :string;
i   : word;
n1   : single;
begin
```

```
 n1:= 10.0;
 str(round(int(N)),S1); S := S1+'.';
 for i:= 1 to M-1 do begin N1:=n1*10;end;
 str(round(frac(N)*n1) ,S1); S:=S+S1;
 roundstr :=S;
end;
{------------------------------------------}
procedure speed_Time;
var
x,y:integer;
begin
 setcolor(15); Rectangle(380,0,620,180);
 setlinestyle(1,0,1);setcolor(7);
 line (380,30,620,30);line (380,60,620,60);
 line (380,120,620,120); line (380,150,620,150);

 line (428,0,428,180); line (476,0,476,180);
 line (524,0,524,180); line (572,0,572,180);
 setcolor(15);
 line (380,45,620,45);
 setlinestyle(0,0,1);
 line (380,90,620,90);
 outtextxy(350,175,'0,0');outtextxy(480,185,'TIME sec');
 outtextxy(360,110,'V');outtextxy(350,120,'m/s');
 outtextxy(360,15,'dc');outtextxy(350,25,'m/s2');
 outtextxy(360,81,'-6'); outtextxy(360,0,'+6'); outtextxy(360,42,'0');
 outtextxy(360,95,'60');
 setcolor(13);line(345,140,375,140);outtextxy(355,130,'F');
 setcolor(15);line(345,155,375,155);outtextxy(355,145,'L');
end;
{------------------------------------------}
procedure threshold;
 var
  x,y:integer;
  theta,dv,dx1,dx,cldv,opdv :single;
 begin
  setcolor (15);
  Rectangle(50,200,550,450);  {Rectangle(100,200,500,400);}
  setlinestyle(1,0,1);
  setcolor(7);
  LINE (100, 450, 100, 200); LINE (150, 450,150, 200);
  LINE (200, 450, 200, 200); LINE (250, 450,250, 200);
  LINE (300, 450, 300, 200); LINE (350, 450,350, 200);
  LINE (400, 450, 400, 200); LINE (450, 450,450, 200);
  LINE (500, 450, 500, 200);
  LINE (50, 225,550, 225); LINE (50, 250,550, 250);
  LINE (50, 275,550, 275); LINE (50, 300,550, 300);
  LINE (50, 325,550, 325); LINE (50, 350,550, 350);
  LINE (50, 400,550, 400);LINE (50, 425,550, 425);
  setlinestyle(0,0,1);LINE (50, 375,550, 375);
  setcolor(15);

  outtextxy(30 ,455,'200m'); outtextxy(292, 465,'DX' );
  outtextxy(88 , 455,'180'); outtextxy(138, 455,'160');
  outtextxy(188, 455,'140'); outtextxy(242, 455,'120' );
  outtextxy(292, 455,'100') ; outtextxy(342, 455,'80' );
  outtextxy(392, 455,'60') ; outtextxy(442, 455,'40' );
  outtextxy(492, 455,'20') ; outtextxy(548, 455,'0'  );
```

```
  outtextxy(555, 220,'12') ; outtextxy(555, 245,'10');
  outtextxy(555, 270,' 8') ; outtextxy(555, 295,' 6');
  outtextxy(555, 320,' 4') ; outtextxy(555, 345,' 2');
  outtextxy(555, 370,' 0') ; outtextxy(555, 395,'-2');
  outtextxy(555, 420,'-4') ; outtextxy(555, 445,'-6 m/s');
  outtextxy(587, 300,'DV') ; outtextxy(555, 200,'14 m/s');

 { drawing the threshold }
  setlinestyle(0,0,1);setcolor(2);
  theta:=0.0012;dv:=14;dx1:=sqrt(dv*2/theta);
  if dx1 >200 then
   begin
    dx1:=200; dv :=sqr(dx1)*theta/2;
  end;
  x:=round(550-dx1*5/2); y:=round(375-dv*12.5); moveto (x,y); dx:=dx1;
  repeat
   begin
    dv:=sqr(dx)*theta/2; x:=round(550-dx*5/2); y:=round(375-dv*12.5);
    lineto(x,y); dx:=dx-5;
   end;
  until dx <= 5;
  theta:=0.0006;dv:=14;dx1:=sqrt(dv*2/theta);
  if dx1 >200 then
  begin
    dx1:=200; dv :=sqr(dx1)*theta/2;
  end;
  x:=round(550-dx1*5/2); y:=round(375-dv*12.5); moveto (x,y); dx:=dx1;
  repeat
   begin
    dv:=sqr(dx)*theta/2; x:=round(550-dx*5/2);
    y:=round(375-dv*12.5); lineto(x,y); dx:=dx - 5;
   end;
  until dx <= 5;
  setcolor (15); outtextxy(90,215,'é=12E-4'); outtextxy(235,360,'é=6E-4');
  setcolor(9);line(0,340,45,340);outtextxy(20,327,'Ac');
  setcolor(10);line(0,370,45,370);outtextxy(20,357,'Dc');
END;
{------------------------------------------}
procedure fvehicle ;
 var S:string;
 begin
  setcolor (15);
  Rectangle(0,40,162,100); line (0,52,162,52); line (0,64,162,64);
  line (0,76,162,76); line (0,88,162,88); line (100,52,100,100);
  S:=' Follower Vehicle'; setcolor (10); outtextxy(10,42,S);
  S:='Acmax m/s/s'; setcolor (10); outtextxy(5,54,S);
  S:='TTAmin  sec'; setcolor (10); outtextxy(5,66,S);
  S:='TTA     sec'; setcolor (10); outtextxy(5,78,S);
  S:='Dspeed  m/s'; setcolor (10); outtextxy(5,90,S);
 end;
{------------------------------------------}
procedure Lvehicle ;
 var S:string;
 begin
  setcolor (15);
  Rectangle(0,112,115,184); line (0,124,115,124); line (0,136,115,136);
  line (0,148,115,148); line (0,160,115,160);line (0,172,115,172);
```

```
   line (10,136,10,184);line (75,124,75,184);
   S:='Lead Vehicle'; setcolor (10); outtextxy(10,114,S);
   S:=' Ac/Dc'; setcolor (10); outtextxy(5,126,S);
   S:='Time'; setcolor (10); outtextxy(77,126,S);
   S:='1'; setcolor (10); outtextxy(2,138,S);
   S:='2'; setcolor (10); outtextxy(2,150,S);
   S:='3'; setcolor (10); outtextxy(2,162,S);
   S:='4'; setcolor (10); outtextxy(2,174,S);
  end;
 { ------------------------------------------}
 procedure drawtimer;
  var S:string;
  begin
   setcolor (15); Rectangle(175,40,315,60); line (245,40,245,60);
   S:='Time sec'; setcolor (10); outtextxy(177,46,S);
  end;
 {------------------------------------------}
 procedure drawDC2;
  var S:string;
  begin
   setcolor (15); Rectangle(175,64,315,84); line (245,64,245,84);
   S:='Dc2 m/s2'; setcolor (10); outtextxy(177,70,S);
  end;
 {------------------------------------------}
 procedure drawEST;
  var S:string;
  begin
   setcolor (15); Rectangle(175,88,315,108); line (245,88,245,108);
   S:='Eodv'; setcolor (10); outtextxy(177,94,S);
  end;
 {------------------------------------------}
 procedure drawTTC;
  var S:string;
  begin
   setcolor (15); Rectangle(175,112,315,132); line (245,112,245,132);
   S:='TTC '; setcolor (10); outtextxy(177,118,S);
  end;
 {------------------------------------------}
 procedure drawDFD;
  var S:string;
  begin
   setcolor (15); Rectangle(175,136,315,156); line (245,136,245,156);
   S:='DFD m'; setcolor (10); outtextxy(177,142,S);
  end;
 {------------------------------------------}
 procedure drawF;
  var S:string;
  begin
   setcolor (15); Rectangle(120,112,155,132); line (140,112,140,132);
   S:='F '; setcolor (10); outtextxy(122,118,S);
   setcolor (15); Rectangle(120,136,155,156); line (140,136,140,156);
   S:='F1'; setcolor (10); outtextxy(122,142,S);
   setcolor (15); Rectangle(120,160,170,180); line (140,160,140,180);
   S:='F2'; setcolor (10); outtextxy(122,168,S);
  end;
 {------------------------------------------}
 procedure savedata;
 var
```

```
 con    : byte;
 con1   : byte;
 NO     : string[6];
 DCmax  : string[10];
 TTCmin : string[10];
 TTRmin : string[10];
 TTA    : string[10];
 DX     : string[10];
 DV     : string[10];
 Ind    : string[6];
begin

 S:=' ';
 str(ABS(Counter),S1); S := S+S1;
 NO := S;
 con := length(DCmax);
 for con1 :=con to 6 do begin NO := NO+' ';end;
{------------------}
 S:=' ';
 str(ABS(F3),S1); S := S+S1;
 IND := S;
 con := length(DCmax);
 for con1 :=con to 6 do begin IND := IND+' ';end;
{------------------}
 S:=' ';
 if DcmaxF < 0.0 then S :='-';
 str(round(int(ABS(DcmaxF))),S1); S := S+S1+'.';
 str(round(INT(frac(ABS(DcmaxF))*100)),S1);
 val(S1,hel,code); if hel<10 then S:=S+'0';if hel>99 then S1:='99';
 S:=S+S1;
 DCmax := S;
 con := length(DCmax);
 for con1 :=con to 10 do begin DCmax := DCmax+' ';end;
{------------------}
 S:=' ';
 if TTAF < 0.0 then S :='-';
 str(round(int(ABS(TTAF))),S1); S := S+S1+'.';
 str(round(Int(frac(ABS(TTAF))*100)),S1);
 val(S1,hel,code); if hel<10 then S:=S+'0';if hel>99 then S1:='99';
 S:=S+S1;
 TTA := S;
 con := length(TTA);
 for con1 :=con to 10 do begin TTA := TTA+' ';end;
{------------------}
 S:=' ';
 if TTAminF < 0.0 then S :='-';
 str(round(int(ABS(TTAminF))),S1); S := S+S1+'.';
 str(round(Int(frac(ABS(TTAminF))*100)),S1);
 val(S1,hel,code); if hel<10 then S:=S+'0';if hel>99 then S1:='99';
 S:=S+S1;
 TTCmin := S;
 con := length(TTCmin);
 for con1 :=con to 10 do begin TTCmin := TTCmin+' ';end;
{------------------}
 S:=' ';
 if TTAmin < 0.0 then S :='-';
 str(round(int(ABS(TTAmin))),S1); S := S+S1+'.';
 str(round(Int(frac(ABS(TTAmin))*100)),S1);
```

```
  val(S1,hel,code); if hel<10 then S:=S+'0';if hel>99 then S1:='99';
  S:=S+S1;
  TTRmin := S;
  con := length(TTRmin);
  for con1 :=con to 10 do begin TTRmin := TTRmin+' ';end;
{-----------------}
  S:=' ';
  str(round(int(ABS(DXF))),S1); S := S+S1+'.';
  str(round(INT(frac(ABS(DXF))*100)),S1);
  val(S1,hel,code); if hel<10 then S:=S+'0';if hel>99 then S1:='99';
  S:=S+S1;
  DX := S;
  con := length(DX);
  for con1 :=con to 10 do begin DX := DX+'  ';end;
{-----------------}
  S:=' ';
  if DVF < 0.0 then S :='-';
  str(round(int(ABS(DVF))),S1); S := S+S1+'.';
  str(round(Int(frac(ABS(DVF))*100)),S1);
  val(S1,hel,code); if hel<10 then S:=S+'0';if hel>99 then S1:='99';
  S:=S+S1;
  DV := S;
  con := length(DV);
  for con1 :=con to 10 do begin DV := DV+' ';end;
{-----------------}
  writeln(F,No,TTA,TTCmin,TTRmin,DCmax,DX,DV,IND);
end; {end of procedure-------------}
begin
end.
```

APPENDIX (3)

THE ABX & SDX VARIATION

When Discarding A Proportion Change In The Relative Distance DX For Semi-Spirals Less Than (0.12)

1. INTRODUCTION :

In Chapter (8) the ABX & SDX thresholds were defined as :

- **ABX:** where a trace with decreasing DX and negative DV changes to one with increasing DX and positive DV.
- **SDX:** where a trace with DX increasing and positive DV changes to one with DX decreasing and negative DV.

However, in isolating and extracting these action points it was mentioned that: [*because of the reported minimum measured value for the just noticeable distance (JND) is [JND ≥ 0.06] LEONARDO EVANS and RICHARD ROTHERY (1977) Ref. [62], any semi-spiral with a proportion change in the relative distance DX less than 0.06 was discarded.*]. Several researchers however, suggested, that JND might be in the limits of (0.12), BRAUNSTEIN M. L. AND LAUGHERTY K. R. (1964) Ref. [68]. Therefore, another criteria was considered by discarding any semi-spiral with [(SDX / ABX) < 1.12]. This Appendix will present the results when considering [JND ≥ 0.12] and compare it with the initial finding in Chapter (8).

2. THE ABX & SDX LIMITS :

Figure (a3.1) presents the ABX and SDX limits over the speed range for each of the JND critical value (0.06 and 0.12). Obviously, the change is minor and very unnoticeable and difference in range of about less than five meters. Nevertheless, the fitted power relationship equations were found to be :

$ABX_{min} = 3.2361 * V^{0.213}$ $ABX_{max} = 1.3856 * V^{0.704}$

$SDX_{min} = 2.8308 * V^{0.3515}$ $SDX_{max} = 2.1472 * V^{0.6854}$

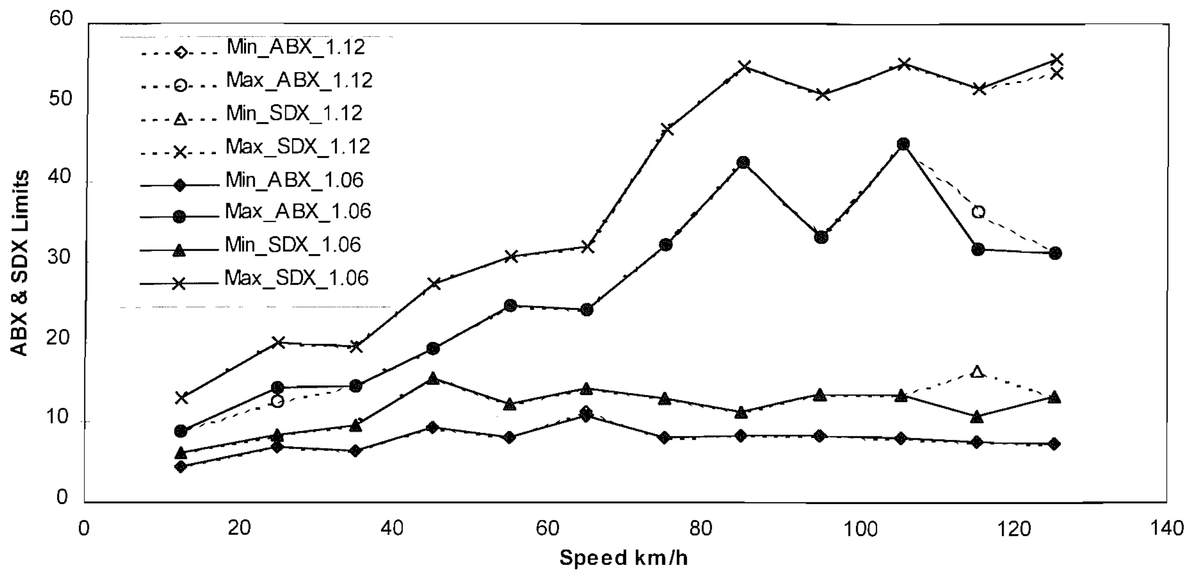


Figure a3.1: The maximum and minimum Limits for ABX & SDX considering two JND critical Values

Looking back at the data very little drivers were dismissed while the values of their ABX and SDX has changed slightly. On the other hand, Figure (a3.2) shows the (SDX/ABX0 percentage limits over the speed range and we can see how little these limits are affected.

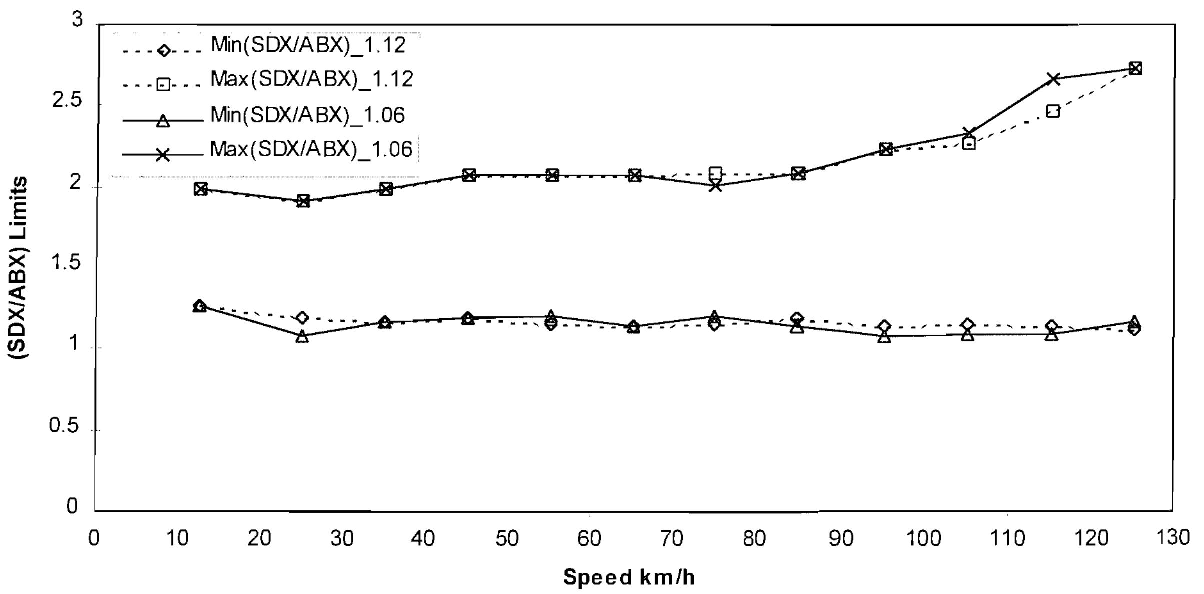


Figure a3.2: The maximum and minimum Limits for the (SDX/ABX) percentage considering two JND critical Values

3. THE ABX DISTRIBUTIONS :

Figure (a3.3) shows the ABX distribution over the speed range for each of the JND critical value (0.06 and 0.12). Although, the general shape is the same for the two ways of filtering the data, but, there are insignificant differences between most the values represented on the graph.

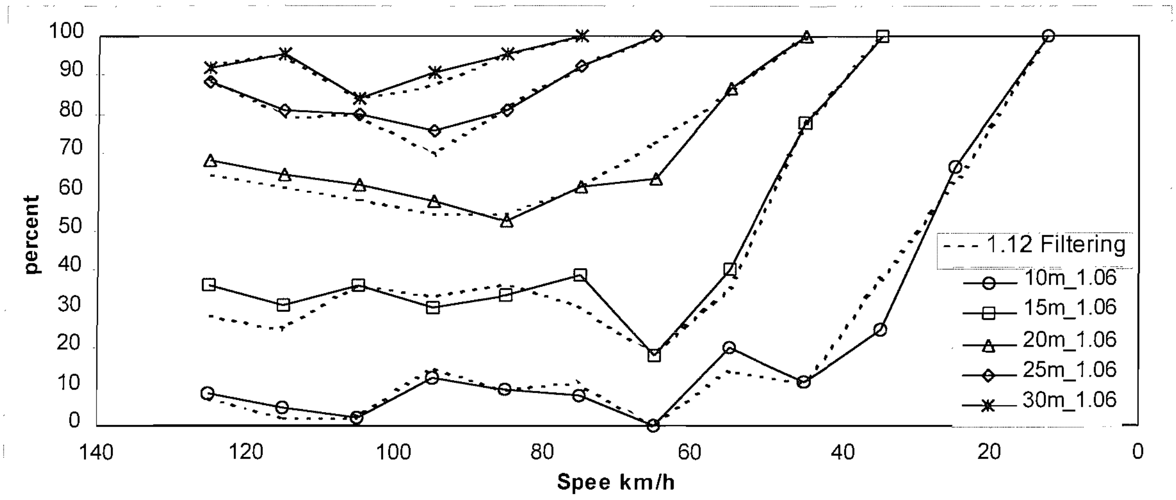


Figure (a3.3) : The ABX Distribution over the speed range considering two JND critical Values for filtering.

4. THE (SDX/ABX) RATIO DISTRIBUTIONS :

Figure (a3.4) shows the (SDX/ABX) ratio distribution over the speed range for each of the JND critical value (0.06 and 0.12). It can be seen that there are some difference between the two sets of data, but the general shape is mainly the same.

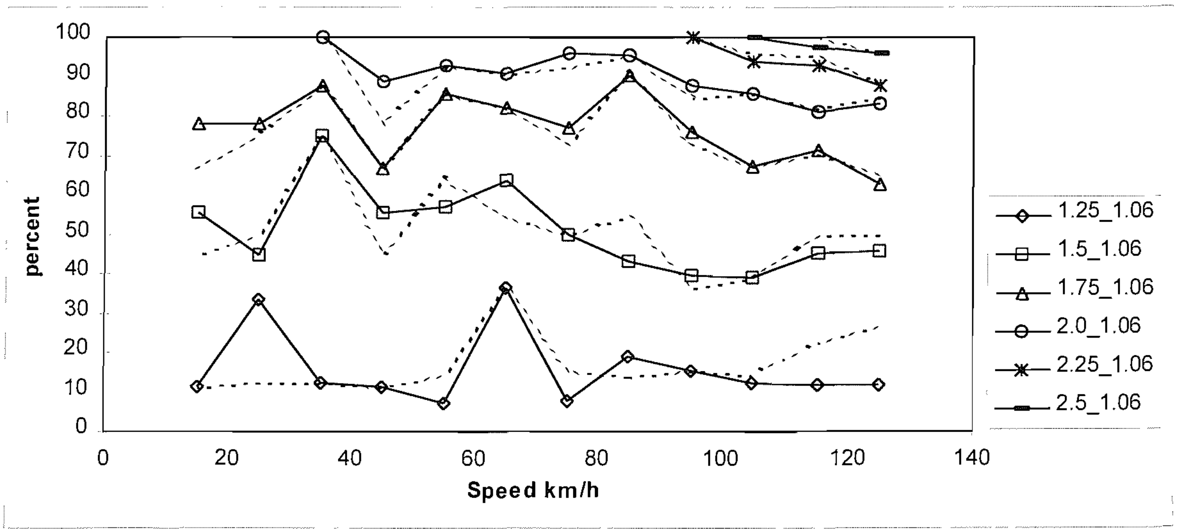


Figure (a3.4) : The (SDX/ABX) ratio Distribution over the speed range considering two JND critical Values for filtering.

5. CONCLUSION :

The justification for considering the just noticeable distance (JND) restriction on the data filtering was explained in Chapter (8). Nevertheless, the critical value of [JND] was taken as [0.06] instead of [0.12] for several reasons :

1- The JND (0.06) value allows more data to inter the analysis process and minimises the defect of discarding real measurements from the following distance, which means it is better and more safe for distribution studies.

2- The general shape of the outcome thresholds ABX and SDX is the same for (0.06) and (0.12) JND critical value.

3- There is no evidence to support one critical value over the other.

APPENDIX (4)

THE SIMULATION PROGRAM CODE & ITERFACE

The simulation program was developed using Delphy programming language, which uses the Turbo Pascal code with the Widows Objects Oriented way. The program interface is presented in Figure a4.1, which shows how it is easy to change the initial configurations in the simulation.

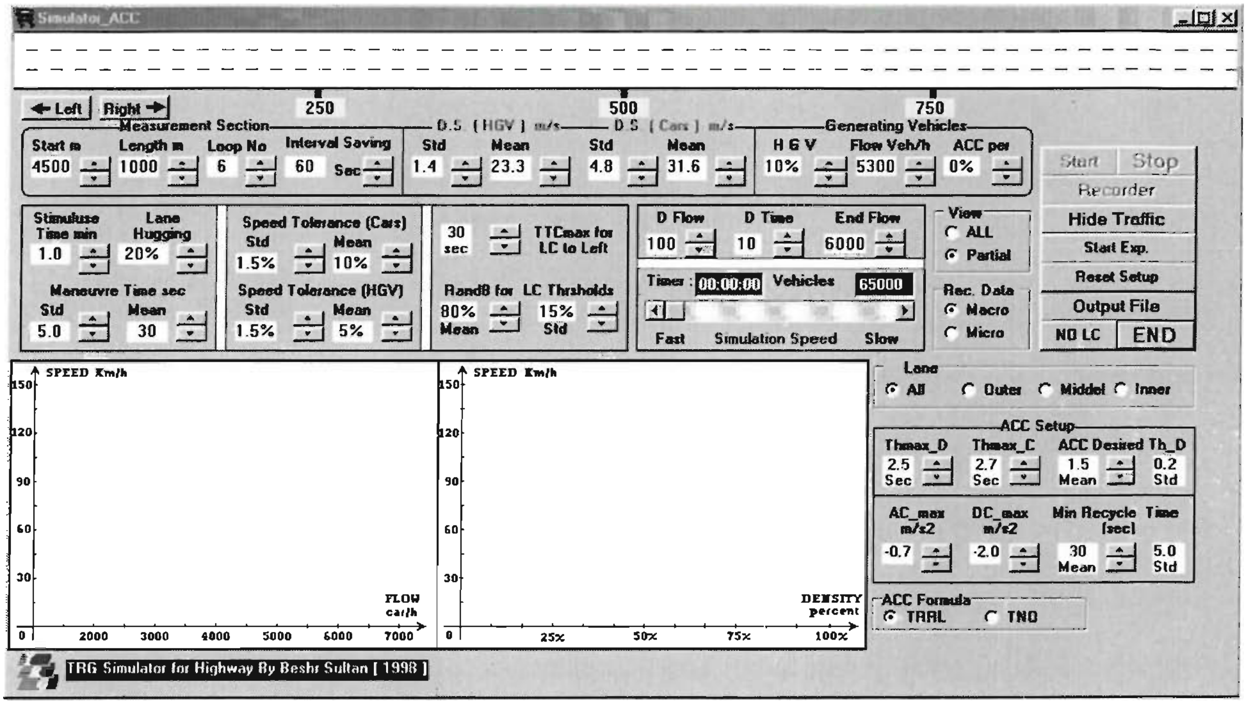


Figure a4.1: The simulation model interface.

The CODE LIST

```
unit simulator_ACC;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs,
  ExtCtrls, StdCtrls, Buttons, Menus, ComCtrls,
  data_ACC,variab_ACC,gener_ACC,folow_ACC,lane_ACC;
type
  TForm1 = class(TForm)
    Button1: TButton;
    Shape1: TShape;
    Shape2: TShape;
    Shape3: TShape;
    Edit4: TEdit;
    Label1: TLabel;
```

```
Btnleft: TBitBtn;
btnright: TBitBtn;
CheckBox1: TCheckBox;
PaintBox1: TPaintBox;
Button2: TButton;
Button4: TButton;
Label2: TLabel;
Edit5: TEdit;
Label4: TLabel;
Edit7: TEdit;
Shape8: TShape;
Label5: TLabel;
Label7: TLabel;
Label10: TLabel;
Label13: TLabel;
Label3: TLabel;
UpDown1: TUpDown;
UpDown2: TUpDown;
Label15: TLabel;
Shape10: TShape;
Button5: TButton;
CheckBox2: TCheckBox;
Shape6: TShape;
Image1: TImage;
Button6: TButton;
CheckBox3: TCheckBox;
Label18: TLabel;
UpDown3: TUpDown;
Label19: TLabel;
Label20: TLabel;
Label23: TLabel;
UpDown5: TUpDown;
Label24: TLabel;
Label26: TLabel;
UpDown6: TUpDown;
Label27: TLabel;
Label28: TLabel;
Label14: TLabel;
Label29: TLabel;
UpDown7: TUpDown;
Label11: TLabel;
Label32: TLabel;
PaintBox2: TPaintBox;
UpDown8: TUpDown;
Label12: TLabel;
Label17: TLabel;
Label36: TLabel;
OpenDialog1: TOpenDialog;
Shape7: TShape;
Shape9: TShape;
UpDown9: TUpDown;
Label25: TLabel;
Label38: TLabel;
UpDown10: TUpDown;
Label39: TLabel;
Label40: TLabel;
UpDown11: TUpDown;
```

```
Label41: TLabel;
Label42: TLabel;
UpDown12: TUpDown;
Label43: TLabel;
Label44: TLabel;
Label45: TLabel;
Label46: TLabel;
CheckBox4: TCheckBox;
RadioGroup2: TRadioGroup;
Shape13: TShape;
Shape14: TShape;
Label52: TLabel;
Label53: TLabel;
Label54: TLabel;
Label55: TLabel;
Label56: TLabel;
UpDown13: TUpDown;
UpDown14: TUpDown;
RadioGroup1: TRadioGroup;
Button7: TButton;
UpDown4: TUpDown;
UpDown15: TUpDown;
Label16: TLabel;
Label8: TLabel;
Label21: TLabel;
Label22: TLabel;
UpDown16: TUpDown;
Label35: TLabel;
UpDown17: TUpDown;
Label57: TLabel;
UpDown18: TUpDown;
Label58: TLabel;
Label59: TLabel;
UpDown19: TUpDown;
Label61: TLabel;
Label62: TLabel;
UpDown20: TUpDown;
Label63: TLabel;
Label64: TLabel;
Button8: TButton;
Label65: TLabel;
Label66: TLabel;
Label67: TLabel;
Label6: TLabel;
Shape16: TShape;
Button3: TButton;
Label9: TLabel;
Label30: TLabel;
UpDown21: TUpDown;
Button9: TButton;
ScrollBar1: TScrollBar;
Label31: TLabel;
Label33: TLabel;
Label34: TLabel;
RadioGroup3: TRadioGroup;
Label47: TLabel;
UpDown22: TUpDown;
```

```
    Label48: TLabel;
    Label68: TLabel;
    Label69: TLabel;
    Label70: TLabel;
    UpDown24: TUpDown;
    UpDown25: TUpDown;
    UpDown26: TUpDown;
    Label71: TLabel;
    Label72: TLabel;
    Label73: TLabel;
    Label74: TLabel;
    Label75: TLabel;
    Label76: TLabel;
    Label77: TLabel;
    UpDown27: TUpDown;
    Thmax_D: TLabel;
    Label78: TLabel;
    UpDown28: TUpDown;
    Label79: TLabel;
    Label80: TLabel;
    UpDown29: TUpDown;
    Shape4: TShape;
    Label37: TLabel;
    Label51: TLabel;
    Label60: TLabel;
    Label82: TLabel;
    Label83: TLabel;
    Label84: TLabel;
    Label85: TLabel;
    UpDown30: TUpDown;
    UpDown31: TUpDown;
    UpDown32: TUpDown;
    Shape15: TShape;
    Shape5: TShape;
    Label81: TLabel;
    RadioGroup4: TRadioGroup;
    procedure RadioGroup1Click(Sender: TObject);
    procedure RadioGroup3Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure BtnleftClick(Sender: TObject);
    procedure btnrightClick(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure ApplicationActivate(Sender: TObject);
    procedure Edit5Change(Sender: TObject);
    procedure FormResize(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure UpDown1Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown2Click(Sender: TObject; Button: TUDBtnType);
    procedure Button5Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure UpDown3Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown4Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown5Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown6Click(Sender: TObject; Button: TUDBtnType);
```

```
    procedure UpDown7Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown8Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown9Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown10Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown11Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown12Click(Sender: TObject; Button: TUDBtnType);
    procedure RadioGroup2Click(Sender: TObject);
    procedure UpDown13Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown14Click(Sender: TObject; Button: TUDBtnType);
    procedure Button7Click(Sender: TObject);
    procedure UpDown15Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown16Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown17Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown18Click(Sender: TObject; Button: TUDBtnType);
    procedure Button8Click(Sender: TObject);
    procedure UpDown19Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown20Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown21Click(Sender: TObject; Button: TUDBtnType);
    procedure Button9Click(Sender: TObject);
    procedure AppDeactivate(Sender: TObject);
    procedure ScrollBar1Change(Sender: TObject);
    procedure UpDown22Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown24Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown25Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown26Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown27Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown28Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown29Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown32Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown31Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown30Click(Sender: TObject; Button: TUDBtnType);
    procedure RadioGroup4Click(Sender: TObject);
 private
    { Private declarations }
 public
    { Public declarations }
end;
var
  Form1: TForm1;
  car,car1,Van,Van1,Road,Road1,car2,car3,Van2,Van3   : Tbitmap;
  Det1,GraphB,GraphC,Dot,carACC                      : Tbitmap;
  F,setF                                             : TextFile;
  fname1,Fname,SetFname                              : string;
implementation
{$R *.DFM}

procedure savedata_mac;
var
 Speed3         : array[0..3] of single;
 Flow3          : array[0..3] of single;
 Density3       : array[0..3] of single;
 Journey        : array[0..3] of single;
 Speed2         : array[0..3] of string[10];
 Journey2       : array[0..3] of string[10];
 Flow2          : array[0..3] of string[10];
 Density2       : array[0..3] of string[10];
 LCnumber2      : array[1..4] of string[10];
```

```
 rectim          : string[10];
 counter2        : single;
 lan             : byte;
 x1,x2,y1        : integer;
begin
 if round(int(T))>(2*RCtime)then begin
  lan :=1;
  while lan <=3 do begin
   p1 := lastv[Lan];
   while (p1<> nil)and(p1^.x[0]>=loopF[1]) do begin
    for x1:=1 to loopD do begin
     if (p1^.x[0]>=loopF[x1])and(p1^.x[4] < loopF[x1]) then begin
      speed1[lan,x1] :=speed1[lan,x1]+p1^.v[4];
      counter1[lan,x1]:=counter1[lan,x1]+1;
     end;
    end;
    p1:=p1^.back;
   end;
   p1 := lastv[Lan];counter2:=0;
   while (p1<> nil)and(p1^.x[0]>=loopF[1]) do begin
    for x1:=1 to (loopD-1) do begin
     if (p1^.x[0]>=loopF[x1])and(p1^.x[4] < loopF[x1+1]) then begin
      Density1[lan,x1] := Density1[lan,x1]+p1^.veh.lenV/(loopF[x1+1]-
loopF[x1]);
     end;
    end;
    p1:=p1^.back;
   end;
   lan :=lan+1;
  end;
 {--------}
  intim:=intim+0.5;
  if (int(intim)=intim)and (round(intim)= RCtime) then begin
   {-----------}
   str(round(T),S1); S := S1; rectim := S;
   {-----------}
   lan :=1;
   while lan <=3 do begin
     Speed3[lan]:=0.0;Flow3[lan]:=0.0;density3[lan]:=0.0;
     for x1:=1 to loopD do begin
      if counter1[lan,x1]<>0.0 then
       speed1[lan,x1]:=speed1[lan,x1]*3.6/counter1[lan,x1]
      else begin Speed1[lan,x1]:=0.0;end;
      density1[lan,x1]:=density1[lan,x1]/(RCTime*2);
      Speed3[lan]:=speed3[lan]+Speed1[lan,x1];
      Flow3[lan]:=Flow3[lan]+counter1[lan,x1]*3600/RCtime;
     end;
     for x1:=1 to (loopD-1) do begin
      density3[lan]:=density3[lan]+density1[lan,x1];
     end;
     Speed3[lan]:=Speed3[lan]/loopd;
     if speed3[lan]<> 0.0 then
      Journey[lan]:=(Dend-Dstar)*3.6/Speed3[lan]
     else Journey[lan]:=0.0;
     Flow3[lan]:=flow3[lan]/loopd;
     density3[lan]:=density3[lan]/(loopd-1);
   {-----------}
```

```
      lan:=lan+1;
     end;
     Speed3[0]:=(speed3[1]+speed3[2]+speed3[3])/3;
     Journey[0]:=(Journey[1]+Journey[2]+Journey[3])/3;
     Flow3[0]:=Flow3[1]+Flow3[2]+Flow3[3];
     density3[0]:=(density3[1]+density3[2]+density3[3])/3;
     {-----------}
     LCnumber[1]:=round(LCnumber[1]*3600*1000/(RCtime*(Dend-Dstar)));
     STR(round(LCnumber[1]),S);LCnumber2[1]:=S;
     LCnumber[2]:=round(LCnumber[2]*3600*1000/(RCtime*(Dend-Dstar)));
     STR(round(LCnumber[2]),S);LCnumber2[2]:=S;
     LCnumber[3]:=round(LCnumber[3]*3600*1000/(RCtime*(Dend-Dstar)));
     STR(round(LCnumber[3]),S);LCnumber2[3]:=S;
     LCnumber[4]:=round(LCnumber[4]*3600*1000/(RCtime*(Dend-Dstar)));
     STR(round(LCnumber[4]),S);LCnumber2[4]:=S;
     {-----------}
     if (Flow3[LaneRG]<>0.0)then begin
     {if (form1.checkbox2.checked =true)and (counter1[laneRG]<>0.0)then
begin}
      if speed3[LaneRG] <= 150 then y1:= round((150-speed3[LaneRG])*20/15)
+20 else y1:=20;
      if laneRG <>0 then begin
       if Flow3[LaneRG] <=3000  then x1:=round(Flow3[LaneRG]/10)+20 else
x1:=320;
      end
      else begin
       if (Flow3[LaneRG] <=7000)and(Flow3[LaneRG] >=1000) then
        x1:=round((Flow3[LaneRG]-1000)/20)+20
       else begin
         if (Flow3[LaneRG] >7000)then x1:=320;
         if (Flow3[LaneRG] <1000)then x1:=20;
       end;
      end;
      x2:= round(density3[laneRG]*3 )+370;
      form1.PaintBox2.Canvas.Draw(x1,y1,dot);
      form1.PaintBox2.Canvas.Draw(x2,y1,dot);
     end;
     if form1.checkbox1.checked =true then begin
      lan :=0;
      while lan <=3 do begin
       str(round(int(Speed3[lan])),S1); S := S1+'.';
       str(round(frac(Speed3[lan])*100),S1); S:=S+S1;
       Speed2[lan] := S;
       str(round(int(Journey[lan])),S1); S := S1+'.';
       str(round(frac(Journey[lan])*100),S1); S:=S+S1;
       Journey2[lan] := S;
       STR(round(Flow3[lan]),S); Flow2[lan]:=S;
       STR(round(density3[lan]),S); Density2[lan]:=S;
       lan:=lan+1;
      end;
     {-----------}
      write(F,rectim,',',Density2[1],',',Flow2[1],',',Speed2[1],',');
      write(F,Density2[2],',',Flow2[2],',',Speed2[2],',');
      write(F,Density2[3],',',Flow2[3],',',Speed2[3],',');
      write(F,Density2[0],',',Flow2[0],',',Speed2[0],',');

Write(F,LCnumber[1],',',LCnumber[2],',',LCnumber[3],',',LCnumber[4],',');
```

```
Writeln(F,Journey2[1],',',Journey2[2],',',Journey2[3],',',Journey2[0]);
   end;
   LCnumber[1]:=0;LCnumber[2]:=0;LCnumber[3]:=0;LCnumber[4]:=0;
   lan:=1;intim:=0.0;density3[0]:=0.0;
   while lan <=3 do begin
    for x1 :=1 to 10 do begin
     counter1[lan,x1]:=0;speed1[lan,x1]:=0.0;density1[lan,x1]:=0.0;
    end;
    lan :=lan+1;
   end;
  end;
 end;
end;{ end of procedure
---------------------------------------}
procedure savedata_mic;
var
desSpeed, ABX,SDX,Vspeed,Vleader          :array[1..3]of single;
desSpeed1, ABX1,SDX1,Vspeed1,Vleader1,TK  :array[1..3]of string[10];
TimerL                                    :string[10];
Lan,place                                 :word;
begin
 if (round(int(T))>(2*RCtime))and(form1.checkbox1.checked =true)then begin
  lan :=1;TimerL:='';place:=0;
  while lan <=3 do begin
   ABX[lan]:=0.0;  SDX[lan]:=0.0;
DesSpeed[lan]:=0.0;Vleader[lan]:=0.0;Vspeed[lan]:=0.0;
   ABX1[lan]:='';  SDX1[lan]:='';
DesSpeed1[lan]:='';Vleader1[lan]:='';TK[lan]:='';Vspeed1[lan]:='';
   p1 := lastv[Lan];
   while (p1<> nil)and(p1^.x[0]>=Dstar) do begin
    if (p1^.x[0]>= Dstar)and(p1^.x[4] <Dstar) then begin
      Place:=1;
      DesSpeed[lan] :=p1^.dvr.DSpeed;
      s:='';S1:='';
      str(round(int(DesSpeed[lan])),S1); S := S1+'.';
      str(round(frac(DesSpeed[lan])*100),S1); S:=S+S1;
      DesSpeed1[lan]:= S;
      s:='';S1:='';
      str(round(int(T)),S1); S := S1+'.';
      str(round(frac(T)*100),S1); S:=S+S1;
      TimerL:= S;
      Vspeed[lan] :=p1^.v[0];
      s:='';S1:='';
      str(round(int(Vspeed[lan])),S1); S := S1+'.';
      str(round(frac(Vspeed[lan])*100),S1); S:=S+S1;
      Vspeed1[lan]:= S;
      if p1^.ahead <> nil then begin
        ABX[lan]:=ABXthresh(p1^.ahead^.v[4],p1^.dvr.rand[1]);
        SDX[lan]:=SDXthresh(ABX[lan],p1^.ahead^.v[4],p1^.dvr.rand[2]);
        Vleader[lan]:=p1^.ahead^.v[0];
        s:='';S1:='';
        str(round(int(Vleader[lan])),S1); S := S1+'.';
        str(round(frac(Vleader[lan])*100),S1); S:=S+S1;
        Vleader1[lan]:= S;
        s:='';S1:='';
        str(round(int(ABX[lan])),S1); S := S1+'.';
```

```
        str(round(frac(ABX[lan])*100),S1); S:=S+S1;
        ABX1[lan]:= S;
        s:='';S1:='';
        str(round(int(SDX[lan])),S1); S := S1+'.';
        str(round(frac(SDX[lan])*100),S1); S:=S+S1;
        SDX1[lan]:= S;
        s:='';
        str(p1^.veh.kind,S);
        TK[lan]:=S;
      end;
      Break;
    end;
    p1:=p1^.back;
   end;
   lan:=lan +1;
  end;
  if place=1 then begin
write(F,TimerL,',',DesSpeed1[1],',',ABX1[1],',',SDX1[1],',',Vspeed1[1],','
,Vleader1[1],',',TK[1],',');
write(F,DesSpeed1[2],',',ABX1[2],',',SDX1[2],',',Vspeed1[2],',',Vleader1[2
],',',TK[2],',');
writeln(F,DesSpeed1[3],',',ABX1[3],',',SDX1[3],',',Vspeed1[3],',',Vleader1
[3],',',TK[3]);
  end;
 end;
end;{ end of procedure
---------------------------------------}
procedure TForm1.ApplicationActivate(Sender: TObject);
begin
 if checkbox2.checked =true then begin
  if radioGroup1.itemIndex=0 then
   form1.PaintBox1.Canvas.Draw(0,0,road1)
  else form1.PaintBox1.Canvas.Draw(0,0,road);
 end;
 if radioGroup3.itemIndex=0 then begin
  if radioGroup2.itemIndex=0 then  begin
   LaneRG:=0;
   form1.PaintBox2.Canvas.Draw(0,0,GraphC);
  end
  else begin
   if radioGroup2.itemIndex=1 then LaneRG:=1;
   if radioGroup2.itemIndex=2 then LaneRG:=2;
   if radioGroup2.itemIndex=3 then LaneRG:=3;
   form1.PaintBox2.Canvas.Draw(0,0,GraphB);
  end;
 end;
  {Put code for your Application.OnActivate here}
end;
procedure updatime;
var
 hour,min,sec :integer;
begin
  if int(T)=T then begin
   hour :=round(int(T/3600));
   min:=round(INT((T-(hour*3600))/60));
   sec :=round(T-(hour*3600)-(min*60));
    S:='';
```

```
    if hour<1 then
     S:='00:'
    else begin
     if hour<10 then S:='0';
     str(hour,S1);S:=S+S1+':';
    end;
    if min<1 then
     S:=S+'00:'
    else begin
     if min<10 then S:=S+'0';
     str(min,S1);S:=S+S1+':';
    end;
    if sec<1 then
     S:=S+'00'
    else begin
     if sec<10 then S:=S+'0';
     str(sec,S1);S:=S+S1;
    end;
    Form1.edit4.text:=S;  form1.edit4.refresh;
    str(round(int(Number)),S); Form1.edit7.text:=S;  form1.edit7.refresh;
   end;
end;
{-------------------------------------------------------------}
procedure updategraph;
var
 pos1,pos2: integer;
 xg,xg1,yg,yg1,L: integer;
begin
if Form1.radioGroup1.itemIndex=1 then begin
  pos1 := 1000*(posi-1); pos2 := 1000*posi;
  L := 1;
  while L <= 3 do begin
   p1 := lastv[L];
   while (p1<> nil) do begin
    if (p1^.x[2] <= pos2)and (p1^.x[4] >(pos1+10)) then begin
     case p1^.veh.kind of
     1,2 :begin
           xg := round(p1^.x[1]-(5+pos1));yg :=(p1^.olane -1)*15 +5;
           form1.PaintBox1.Canvas.Draw(xg,yg,car1);
           Xg1:= round(p1^.x[0] -(5+pos1));yg1:=(p1^.lane - 1)*15 +5;
           if (p1^.ACC.Eq = 'Y')and(p1^.ACC.opp ='on') then
            form1.PaintBox1.Canvas.Draw(xg1,yg1,carACC)
           else
            form1.PaintBox1.Canvas.Draw(xg1,yg1,car);
          end;
     3,4 :begin
           Xg:= round(p1^.x[1]-(10+pos1));yg :=(p1^.olane -1)*15 +5;
           form1.PaintBox1.Canvas.Draw(xg,yg,van1);
           Xg1:=round(p1^.x[0] -(10+pos1));yg1:=(p1^.lane - 1)*15 +5;
           form1.PaintBox1.Canvas.Draw(xg1,yg1,van);
          end;
     end;
    end;
    if (p1^.x[2] <= pos2)and (p1^.x[0] >pos2) then begin
     case p1^.veh.kind of
     1,2 :begin
           xg := round(p1^.x[1]-(5+pos1));yg :=(p1^.olane -1)*15 +5;
```

```
            form1.PaintBox1.Canvas.Draw(xg,yg,car1);
           end;
      3,4 :begin
            Xg:= round(p1^.x[1]-(10+pos1));yg :=(p1^.olane -1)*15 +5;
            form1.PaintBox1.Canvas.Draw(xg,yg,van1);
           end;
      end;
     end;
     p1 := p1^.back;
    end;
    L:=L+1;
   end;
 { if  (loopd < pos2) and (loopd > pos1) then begin
   form1.PaintBox1.Canvas.Draw((loopD-pos1-2),0,Det1);
  end;}
   if  (Dstar < pos2) and (DStar > pos1) then begin
    form1.PaintBox1.Canvas.Draw((Dstar-pos1-2),0,Det1);
   end;
   if  (Dend < pos2) and (Dend > pos1) then begin
    form1.PaintBox1.Canvas.Draw((Dend-pos1-2),0,Det1);
   end;

 end
 else begin
   L := 1;
   while L <= 3 do begin
    p1 := lastv[L];
    while (p1<> nil) do begin
     case p1^.veh.kind of
     1,2 :begin
           xg := round(p1^.x[1]/10);yg :=(p1^.olane -1)*6 +3;
           Xg1:= round(p1^.x[0] /10);yg1:=(p1^.lane - 1)*6 +3;
           form1.PaintBox1.Canvas.Draw(xg,yg,car3);
           form1.PaintBox1.Canvas.Draw(xg1,yg1,car2);
          end;
     3,4 :begin
           Xg:= round(p1^.x[1]/10);yg :=(p1^.olane -1)*6 +3;
           form1.PaintBox1.Canvas.Draw(xg,yg,van3);
           Xg1:=round(p1^.x[0]/10);yg1:=(p1^.lane - 1)*6 +3;
           form1.PaintBox1.Canvas.Draw(xg1,yg1,van2);
          end;
     end;
     p1 := p1^.back;
    end;
    L:=L+1;
   end;
   form1.PaintBox1.Canvas.Draw(round(Dstar/10),0,Det1);
   form1.PaintBox1.Canvas.Draw(Round(Dend/10),0,Det1);

end;
end;{end procedure
--------------------------------}
procedure TForm1.Button1Click(Sender: TObject);
var
 lan,i,delays1,delays2 :integer;
begin
 if Fname <> edit5.text then begin
```

```
  closefile (F);
  fname := edit5.text;
  Assignfile(F,fname);
  rewrite(F);
  if radioGroup3.itemIndex=0 then  begin
   Write(F,'Timer,D1_car/km,F1_car/h,S1_km/h,D2_car/km,F2_car/h,S2_km/h');
   Write(F,'D3_car/km,F3_car/h,S3_km/h,D_car/km,F_car/h,S_km/h,');
   Writeln(F,'LC12/km/h,LC23/km/h,LC21/km/h,LC32/km/h');
  end
  else begin

Writeln(F,'Timer,DSpeed1,ABX1,SDX1,V1,VL1,type,DSpeed2,ABX2,SDX2,V2,VL2,ty
pe,DSpeed3,ABX3,SDX3,V3,VL3,type');
  end;
                    end;
 button6.enabled :=false;button9.enabled :=false;
 edit5.enabled := false;
 Updown15.enabled:=False; Updown16.enabled:=False;Updown4.enabled:=False;
 { edit8.enabled := false;edit9.enabled := false;edit10.enabled := false;}
 Label65.caption:='250'; Label66.caption:='500'; label67.caption:='750';
 str(round(FlowQ),S);
 Label20.caption := S;
 {PaintBox1.Canvas.Draw(0,0,road);}
 button1.enabled := false; button3.enabled := false;
 button2.enabled := true; button4.enabled := true;
 while button1.enabled = false do begin
   T := T +Dt;
   updatime;
   generator;
   follow;
   if (LCfactor=0)and(int(T)=(T)) then begin
    detectlane;
    lanchang;
   end;
   updatep;
   if checkbox2.checked =true then updategraph;
   if int(2*T)=(2*T) then begin
    if radioGroup3.itemIndex=0 then savedata_mac else savedata_mic ;
   end;
   final;
   for delays1:=0 to (100*form1.scrollbar1.position) do begin
    for delays2:=1 to 200 do begin end;
   end;
   if int(T/3)=(T/3) then   Application.ProcessMessages;

   if (Flowfactor<>1)and((T-startTime)>=(DFTime*60)) then begin
     StartTime:=T;
     updown1.position:=updown1.position+updown19.position;
     if updown1.position >50 then updown1.position :=50;
     FlowQ:=2000+updown1.position*100;
     if FlowQ <FlowEnd then begin
      Qflow[1]:=FlowQ/3; Qflow[3]:=FlowQ/3; Qflow[2]:=FlowQ/3;
      Qflow[1]:=Qflow[1]/3600; Qflow[2]:=Qflow[2]/3600;
Qflow[3]:=Qflow[3]/3600;
      HDmin[1]:=1;HDmin[2]:=1;HDmin[3]:=1;
      HDavr[1]:=1/Qflow[1];HDavr[2]:=1/Qflow[2];HDavr[3]:=1/Qflow[3];
      for i:=1 to 3 do begin
```

```
        if Qflow[i] >(1/FMH[i]) then FMH[i]:=1/QFlow[i];
        Lampda[i] :=QFlow[i]*(1-0.5*sqrt(QFlow[i]));
        phai[i] :=(FMH[i]*QFlow[i]);
        phai[i]:= phai[i]-(0.5*(phai[i]-1)*sqrt(QFlow[i]));
       end;
       str((round(FlowQ)),S);
       Label20.caption := S;
      end
      else begin
       closefile(F);
       lan:=1;
       while lan <=3 do begin
         if lastv[lan]<>nil then begin
          p1 := lastv[lan];p2:=p1^.back;
          while (p2<> nil) do begin
           dispose(p1); p1 := p2;p2:=p1^.back;
          end;
          dispose(p1);
         end;
         lan :=lan+1;
       end;
       det1.free; van.free;car.free;car1.free;van1.free;GraphB.free;
       van2.free;car2.free;car3.free;van3.free;GraphC.free;
       road.free;road1.free;dot.free;carAcc.free;
       button1.enabled := true;
       SetFname:='Setup.dat';Assignfile(setF, SetFname);REwrite(setF);

writeln(SetF,Dstar);writeln(SetF,Dend);writeln(SetF,loopD);writeln(SetF,RC
time);
       writeln(SetF,LChug);writeln(SetF,LCTime);writeln(SetF,SpeedT);
       writeln(SetF,SpeedTH);writeln(SetF,DFtime);writeln(SetF,Dflow);
writeln(SetF,FlowEnd);writeln(SetF,HGVQ);writeln(SetF,flowQ);writeln(SetF,
LCtimeS);
writeln(SetF,stimuT);writeln(SetF,ADS);writeln(SetF,ADSH);writeln(SetF,sdA
DS);
       writeln(SetF,sdADSH);writeln(SetF,SpeedTS);writeln(SetF,SpeedTSH);
       writeln(SetF,TTCLCL);  writeln(SetF,R8);writeln(SetF,R8std);
       writeln(SetF,ACC_per); writeln(SetF,Th_D);writeln(SetF,Hmax_D);
       writeln(SetF,Hmax_C);
writeln(SetF,ACC_DCmax);writeln(SetF,ACC_Acmax);
       writeln(SetF,Tmin_R);
       closefile(setF);
      end;
    end;
 end;
 if FlowQ >=FlowEnd then  Application.Terminate;
end; {end Procesure
--------------------------------}
procedure graph1;
var d  :integer;
begin
 if form1.radioGroup1.itemIndex=0 then begin
  form1.PaintBox1.Canvas.Draw(0,0,road1);
  d:= 2500;  str(d,S);  form1.label65.caption:=S;
  d:= 5000;  str(d,S);  form1.label66.caption:=S;
  d:= 7500;  str(d,S);  form1.label67.caption:=S;
 end
```

```
  else begin
   form1.PaintBox1.Canvas.Draw(0,0,road);
   d:= ((posi-1)*1000)+250;  str(d,S);  form1.label65.caption:=S;
   d:= ((posi-1)*1000)+500;  str(d,S);  form1.label66.caption:=S;
   d:= ((posi-1)*1000)+750;  str(d,S);  form1.label67.caption:=S;
  end;
 end;{end Procesure
 -------------------------------}
 procedure graph2;
 begin
   form1.PaintBox2.Canvas.Draw(0,0,GraphB);
 end;{end Procesure
 -------------------------------}
 procedure TForm1.BtnleftClick(Sender: TObject);
 begin
   if posi >1 then begin
    posi := posi -1; graph1;
    end;
 end;{end Procesure
 -------------------------------}
 procedure TForm1.btnrightClick(Sender: TObject);
 begin
   if posi <10 then begin
    posi := posi +1;graph1;
    end;
 end;{end Procesure
 -------------------------------}
 procedure TForm1.Button2Click(Sender: TObject);
 begin
  button1.enabled := true;
  button3.enabled := true;
  button6.enabled := true;
  button9.enabled := true;
  edit5.enabled := true;
  button2.enabled := false;
  button4.enabled := false;
 end;{end Procesure
 -------------------------------}
 procedure TForm1.Button4Click(Sender: TObject);
 begin
  if checkbox1.checked =false then begin
    checkbox1.checked :=true; button4.caption :='Stop Rec';
  end
  else begin
   if checkbox1.checked =true then begin
   checkbox1.checked :=false;button4.caption :='Recorder';
   end;
  end;
 end;{end Procesure
 -------------------------------}
 procedure TForm1.FormCreate(Sender: TObject);
 var i: integer;
 begin
 fname1 :='';
  Application.OnActivate := ApplicationActivate;
    road:= Tbitmap.Create;  dot:=Tbitmap.Create;
    car:= TBitmap.Create;   car1:= TBitmap.Create;
```

```
    Van:= Tbitmap.Create;    Van1:= Tbitmap.Create;
    car2:= TBitmap.Create;    car3:= TBitmap.Create;
    Van2:= Tbitmap.Create;    Van3:= Tbitmap.Create;
    carACC:= Tbitmap.Create;
    Det1:=Tbitmap.Create;    GraphB:=Tbitmap.Create;
    GraphC:=Tbitmap.Create;
    Road1:=Tbitmap.Create;
    road.LoadFromFile('road.bmp');
    road1.LoadFromFile('road1.bmp');
    car.LoadFromFile('car.bmp');
    car1.LoadFromFile('car1.bmp');
    Van.LoadFromFile('van.bmp');
    Van1.LoadFromFile('van1.bmp');
    car2.LoadFromFile('car2.bmp');
    car3.LoadFromFile('car3.bmp');
    carACC.LoadFromFile('carACC.bmp');
    Van2.LoadFromFile('van2.bmp');
    Van3.LoadFromFile('van3.bmp');
    Det1.LoadFromFile('loopD.bmp');
    GraphB.LoadFromFile('Graph.bmp');
    GraphC.LoadFromFile('Graph1.bmp');
    dot.LoadFromFile('dot.bmp');
  randomize;
 {------------------}
  SetFname:='Setup.dat';Assignfile(setF, SetFname);
   {$I-}
   Reset(setF);
   {$I+}
   if IOResult = 0 then begin
    if filesize(setF)<>0 then begin

readln(SetF,Dstar);readln(SetF,Dend);readln(SetF,loopD);readln(SetF,RCtime
);
     readln(SetF,LChug);readln(SetF,LCTime);readln(SetF,SpeedT);
     readln(SetF,SpeedTH);readln(SetF,DFtime);readln(SetF,Dflow);
readln(SetF,FlowEnd);readln(SetF,HGVQ);readln(SetF,flowQ);readln(SetF,LCti
meS);
readln(SetF,stimuT);readln(SetF,ADS);readln(SetF,ADSH);readln(SetF,sdADS);
     readln(SetF,sdADSH);readln(SetF,SpeedTS);readln(SetF,SpeedTSH);
     readln(SetF,TTCLCL); readln(SetF,R8);readln(SetF,R8std);
     readln(SetF,ACC_per);readln(SetF,Th_D);readln(SetF,Hmax_D);;
readln(SetF,Hmax_C);
     readln(SetF,ACC_DCmax);readln(SetF,ACC_ACmax); readln(SetF,Tmin_R);
     str(ACC_per,S);  label71.caption := S+'%';
Updown26.position:=ACC_per;
     str(Tmin_R,S);   label83.caption := S+' Mean';
Updown32.position:=Tmin_R;
     i:=round(int(Th_D)); str(I,S);i:=round(((Th_D)-
int(Th_D))*10);str(I,S1);
     label77.caption := S+'.'+S1+' Mean';
Updown27.position:=round(10*Th_D);
     i:=round(int(Hmax_D)); str(I,S);i:=round(((Hmax_D)-
int(Hmax_D))*10);str(I,S1);
     label78.caption := S+'.'+S1+' Sec';
Updown28.position:=round(10*Hmax_D);
     i:=round(int(Hmax_C)); str(I,S);i:=round(((Hmax_C)-
int(Hmax_C))*10);str(I,S1);
```

```
    label80.caption := S+'.'+S1+' Sec';
Updown29.position:=round(10*Hmax_C);
    i:=round(int(ABS(ACC_DCmax))); str(I,S);i:=round((int(ACC_DCmax)-
(ACC_DCmax))*10);str(I,S1);
    label85.caption := '-'+S+'.'+S1; Updown31.position:=round(-
10*ACC_DCmax);
    i:=round(int(ACC_ACmax)); str(I,S);i:=round(((ACC_ACmax)-
int(ACC_ACmax))*10);str(I,S1);
    label84.caption := '-'+S+'.'+S1;
Updown30.position:=round(10*ACC_ACmax);
    str(R8,S);  label69.caption := S+'%';  Updown24.position:=R8;
    str(R8std,S);  label70.caption := S+'%';  Updown25.position:=R8std;
    str(TTCLCL,S);  label47.caption := S+' sec';
Updown22.position:=TTCLCL;
    str(Dstar,S);  label22.caption := S;
Updown16.position:=round(Dstar/50);
    str((DEnd-Dstar),S);label21.caption:=S;Updown15.position:=round((DEnd-
Dstar)/50);
    str(loopD,S);  label16.caption := S;   Updown4.position:=round(loopD);
    str(RCtime,S); label12.caption := S;
updown8.position:=round(RCTime/10);
    str(LChug,S);  Label57.caption:=S+'%'; updown17.position:=LChug;
    str(LCTime,S); label18.caption := S;   updown3.position:=LCTime;
    str(SpeedT,S); label24.caption:=S+'%'; updown5.position:= SpeedT;
    str(SpeedTH,S);label53.caption:=S+'%'; updown13.position:= SpeedTH;
    str(DFTime,S); Label64.caption := S;   updown20.position:=DFtime;
    str(DFlow,S);  Label62.caption := S;
updown19.position:=round(Dflow/100);
    str(FlowEnd,S);Label9.caption := S;
updown21.position:=round((flowEnd-2000)/100);
    str(HGVQ,S);    label19.caption:=S+'%'; updown2.position:= HGVQ;
    str(FlowQ,S);Label20.caption:=S;updown1.position:=round((FlowQ-
2000)/100);
    i:=round(int(LCtimeS/10)); str(I,S); i:=round(((LCtimeS/10)-
int(LCtimeS/10))*10);str(I,S1);
    label11.caption := S+'.'+S1;    updown7.position:= LCTimeS;
    i:=round(int(StimuT/2)); str(I,S);i:=round(((StimuT/2)-
int(StimuT/2))*10);str(I,S1);
    label58.caption := S+'.'+S1; Updown18.position:=StimuT;
    i:=round(int(ADS)); str(I,S);  i:=round((ADS-int(ADS))*10);
str(I,S1);
    label25.caption := S+'.'+S1;   updown9.position:=i+10;
    i:=round(int(ADSH)); str(I,S);i:=round((ADSH-int(ADSH))*10);str(I,S1);
    label41.caption := S+'.'+S1; updown11.position:=i+10;
    i:=round(int(SDADS)); str(I,S);i:=round((SDADS-
int(SDADS))*10);str(I,S1);
    label39.caption := S+'.'+S1;   updown10.position:=round(SDADS*10);
    i:=round(int(SDADSH)); str(I,S);i:=round((SDADSH-
int(SDADSH))*10);str(I,S1);
    label43.caption := S+'.'+S1;  updown12.position:=round(SDADSH*10);
    i:=round(int(SpeedTS/10)); str(I,S); i:=round(((SpeedTS/10)-
int(SpeedTS/10))*10);str(I,S1);
    label27.caption := S+'.'+S1+'%'; updown6.position:=SpeedTS;
    i:=round(int(SpeedTSH/10)); str(I,S); i:=round(((SpeedTSH/10)-
int(SpeedTSH/10))*10);str(I,S1);
    label55.caption := S+'.'+S1+'%'; updown14.position:=SpeedTSH;
  end
```

```
    else begin
     rewrite(SetF);

Dstar:=3000;Dend:=6000;loopd:=6;RCtime:=60;LChug:=0;LCtime:=30;SpeedT:=10;
     SpeedTH:=5;DFtime:=15;Dflow:=100;FlowEnd:=7000;HGVQ:=10;FlowQ:=2000;
     LCtimeS:=25;stimuT:=2;ADS:=31.6;ADSH:=23.3;sdADS:=4.6;sdADSH:=1.43;
     SpeedTS:=15;SpeedTSH:=15;TTCLCL:=30;R8:=65;R8std:=25;
     ACC_per :=20; Th_D:=1.5; Hmax_D:=2.5; Hmax_C:=2.7;
     ACC_DCmax :=-2.0;ACC_ACmax :=0.7; Tmin_R :=30;


writeln(SetF,Dstar);writeln(SetF,Dend);writeln(SetF,loopD);writeln(SetF,RC
time);
     writeln(SetF,LChug);writeln(SetF,LCTime);writeln(SetF,SpeedT);
     writeln(SetF,SpeedTH);writeln(SetF,DFtime);writeln(SetF,Dflow);

writeln(SetF,FlowEnd);writeln(SetF,HGVQ);writeln(SetF,flowQ);writeln(SetF,
LCtimeS);

writeln(SetF,stimuT);writeln(SetF,ADS);writeln(SetF,ADSH);writeln(SetF,sdA
DS);
     writeln(SetF,sdADSH);writeln(SetF,SpeedTS);writeln(SetF,SpeedTSH);
     writeln(SetF,TTCLCL); writeln(SetF,R8);writeln(SetF,R8std);
     writeln(SetF,ACC_per); writeln(SetF,Th_D);writeln(SetF,Hmax_D);
     writeln(SetF,Hmax_C); writeln(SetF,ACC_DCmax);writeln(SetF,ACC_Acmax);
     writeln(SetF,Tmin_R);
    end;
   end
   else begin
     rewrite(SetF);

Dstar:=3000;Dend:=6000;loopd:=6;RCtime:=60;LChug:=0;LCtime:=30;SpeedT:=10;
     SpeedTH:=5;DFtime:=15;Dflow:=100;FlowEnd:=7000;HGVQ:=10;FlowQ:=2000;
     LCtimeS:=25;stimuT:=2;ADS:=31.6;ADSH:=23.3;sdADS:=4.6;sdADSH:=1.43;
     SpeedTS:=15;SpeedTSH:=15;TTCLCL:=30;R8:=65;R8std:=25;
     ACC_per :=20; Th_D:=1.5; Hmax_D:=2.5; Hmax_C:=2.7;
     ACC_DCmax :=-2.0;ACC_ACmax :=0.7; Tmin_R :=30;


writeln(SetF,Dstar);writeln(SetF,Dend);writeln(SetF,loopD);writeln(SetF,RC
time);
     writeln(SetF,LChug);writeln(SetF,LCTime);writeln(SetF,SpeedT);
     writeln(SetF,SpeedTH);writeln(SetF,DFtime);writeln(SetF,Dflow);

writeln(SetF,FlowEnd);writeln(SetF,HGVQ);writeln(SetF,flowQ);writeln(SetF,
LCtimeS);

writeln(SetF,stimuT);writeln(SetF,ADS);writeln(SetF,ADSH);writeln(SetF,sdA
DS);
     writeln(SetF,sdADSH);writeln(SetF,SpeedTS);writeln(SetF,SpeedTSH);
     writeln(SetF,TTCLCL);writeln(SetF,R8);writeln(SetF,R8std);
     writeln(SetF,ACC_per); writeln(SetF,Th_D);writeln(SetF,Hmax_D);
     writeln(SetF,Hmax_C); writeln(SetF,ACC_DCmax);writeln(SetF,ACC_Acmax);
     writeln(SetF,Tmin_R);
  end;
  closeFile(setF);
 {--------------------------------}
```

```
 if radioGroup3.itemIndex=0 then  begin
  fname:=edit5.text; Assignfile(F, Fname); rewrite(F);
  Write(F,'Timer,D1_car/km,F1_car/h,S1_km/h,D2_car/km,F2_car/h,S2_km/h,');
  Write(F,'D3_car/km,F3_car/h,S3_km/h,D_car/km,F_car/h,S_km/h,');

Writeln(F,'LC12/km/h,LC23/km/h,LC21/km/h,LC32/km/h,J1_sec,J2_sec,J3_sec,J_
sec');
 end
 else begin

Writeln(F,'Timer,DSpeed1,ABX1,SDX1,V,VL,type,DSpeed2,ABX2,SDX2,V,VL,type,D
Speed3,ABX3,SDX3,V,VL,type');
 end;
 getdata;
 count := 1;LCfactor:=0;Flowfactor:=1;
 FMH[1]:=1.6; FMH[2]:=1.3; FMH[3]:=1.3;
 Qflow[1]:=FlowQ/3; Qflow[3]:=FlowQ/3; Qflow[2]:=FlowQ/3;
 Qflow[1]:=Qflow[1]/3600; Qflow[2]:=Qflow[2]/3600;
Qflow[3]:=Qflow[3]/3600;
 HDmin[1]:=1;HDmin[2]:=1;HDmin[3]:=1;
 HDavr[1]:=1/Qflow[1];HDavr[2]:=1/Qflow[2];HDavr[3]:=1/Qflow[3];
 count:=1;
 while count <=3 do begin
  if Qflow[count] >(1/FMH[count]) then FMH[count]:=1/QFlow[count];
  Lampda[count] :=QFlow[count]*(1-0.5*sqrt(QFlow[count]));
  phai[count] :=(FMH[count]*QFlow[count]);
  phai[count]:= phai[count]-(0.5*(phai[count]-1)*sqrt(QFlow[count]));
  count:=count+1;
 end;
 count :=0; while count <=21 do begin norm1[count]:=1000.0;count:=count+1;
end;
 count :=3; while count <=7 do begin
lognorm1[count]:=1000.0;count:=count+1; end;
 recorder:='of';
 HGV[1]:=round((FlowQ*HGVQ/200)/(36*QFlow[1]));
 HGV[2]:=round((FlowQ*HGVQ/200)/(36*QFlow[2]));
 HGV[3]:=0;
 Stimuhead:=30;
 count :=0;
 while count <= 4 do begin
  start[count]:=nil;lastV[count]:=nil;count := count +1;
 end;
 count :=1;
 while count <= 3 do begin
  turn:=1;
  while turn <=10 do begin
   counter1[count,turn]:=0;Speed1[count,turn]:=0.0;
Density1[count,turn]:=0.0;
   turn:=turn+1;
  end;
  count := count +1;
 end;
 intim:=0.0; Number :=0; posi :=1;
 T:=0.0;Dt:=0.125;TF[1,2] :=0.25;TF[2,2] :=1.0;TF[3,2] :=2.0;
 TF[1,1] :=0.0;TF[2,1] :=0.0;TF[3,1] :=0.0;ACC_Formula:=0;
 Number :=0; recorder :='of';lanet1:=1;lanet2:=2;
 laneR:=1;LaneRG:=0;
```

```
  count:=1;
  while count<=loopD do begin
   loopF[count]:=Dstar+((count-1)*(Dend - Dstar)/(loopD-1));
   count:=count+1;
  end;
  count :=0;
 end;{end Procesure
 -------------------------------}
 procedure TForm1.Edit5Change(Sender: TObject);
 begin
  if Edit5.text <> fname then begin
   Updown15.enabled := true;  Updown16.enabled := true;
  end
  else begin
   Updown15.enabled := false;
   Updown16.enabled := false;
  end;
 end;{end Procesure
 -------------------------------}
 procedure TForm1.FormResize(Sender: TObject);
 begin
 if checkbox2.checked =true then begin
  if radioGroup1.itemIndex=0 then
   form1.PaintBox1.Canvas.Draw(0,0,road1)
  else form1.PaintBox1.Canvas.Draw(0,0,road);
 end;
 if radioGroup3.itemIndex=0 then  begin
  if radioGroup2.itemIndex=0 then  begin
   LaneRG:=0;
   form1.PaintBox2.Canvas.Draw(0,0,GraphC);
  end
  else begin
   if radioGroup2.itemIndex=1 then LaneRG:=1;
   if radioGroup2.itemIndex=2 then LaneRG:=2;
   if radioGroup2.itemIndex=3 then LaneRG:=3;
   form1.PaintBox2.Canvas.Draw(0,0,GraphB);
  end;
 end;
 end;{end Procesure
 -------------------------------}
 procedure TForm1.Button3Click(Sender: TObject);
 var
 lan : word;
 begin
 closefile(F);
 lan:=1;
 while lan <=3 do begin
  if lastv[lan]<>nil then begin
   p1 := lastv[lan];p2:=p1^.back;
   while (p2<> nil) do begin
    dispose(p1); p1 := p2;p2:=p1^.back;
   end;
   dispose(p1);
  end;
  lan :=lan+1;
 end;
 det1.free; van.free;car.free;car1.free;van1.free;GraphB.free;
```

```
van2.free;car2.free;car3.free;van3.free;GraphC.free;
road.free;road1.free;dot.free;carACC.free;
SetFname:='Setup.dat';Assignfile(setF, SetFname);rewrite(SetF);
writeln(SetF,Dstar);writeln(SetF,Dend);writeln(SetF,loopD);writeln(SetF,RC
time);
writeln(SetF,LChug);writeln(SetF,LCTime);writeln(SetF,SpeedT);
writeln(SetF,SpeedTH);writeln(SetF,DFtime);writeln(SetF,Dflow);
writeln(SetF,FlowEnd);writeln(SetF,HGVQ);writeln(SetF,flowQ);writeln(SetF,
LCtimeS);
writeln(SetF,stimuT);writeln(SetF,ADS);writeln(SetF,ADSH);writeln(SetF,sdA
DS);
writeln(SetF,sdADSH);writeln(SetF,SpeedTS);writeln(SetF,SpeedTSH);
writeln(SetF,TTCLCL);writeln(SetF,R8);writeln(SetF,R8std);
writeln(SetF,ACC_per); writeln(SetF,Th_D);writeln(SetF,Hmax_D);
writeln(SetF,Hmax_C); writeln(SetF,ACC_DCmax);writeln(SetF,ACC_Acmax);
writeln(SetF,Tmin_R);
closefile(setF);
Application.Terminate;
end;{end Procesure
-------------------------------}
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
var
 lan :word;
begin
 button1.enabled := true;
 button3.enabled := true;
 button6.enabled := true;
 edit5.enabled := true;
 button2.enabled := false;
 button4.enabled := false;
 closefile(F);
 lan:=1;
 while lan <=3 do begin
  if lastv[lan]<>nil then begin
   p1 := lastv[lan];p2:=p1^.back;
   while (p2<> nil) do begin
    dispose(p1); p1 := p2;p2:=p1^.back;
   end;
   dispose(p1);
  end;
  lan :=lan+1;
 end;
 det1.free; van.free;car.free;car1.free;van1.free;GraphB.free;
 van2.free;car2.free;car3.free;van3.free;GraphC.free;
 road.free;road1.free;dot.free;carACC.free;
 SetFname:='Setup.dat';Assignfile(setF, SetFname);rewrite(SetF);

writeln(SetF,Dstar);writeln(SetF,Dend);writeln(SetF,loopD);writeln(SetF,RC
time);
 writeln(SetF,LChug);writeln(SetF,LCTime);writeln(SetF,SpeedT);
 writeln(SetF,SpeedTH);writeln(SetF,DFtime);writeln(SetF,Dflow);

writeln(SetF,FlowEnd);writeln(SetF,HGVQ);writeln(SetF,flowQ);writeln(SetF,
LCtimeS);

writeln(SetF,stimuT);writeln(SetF,ADS);writeln(SetF,ADSH);writeln(SetF,sdA
DS);
```

```
  writeln(SetF,sdADSH);writeln(SetF,SpeedTS);writeln(SetF,SpeedTSH);
  writeln(SetF,TTCLCL);
  writeln(SetF,R8);writeln(SetF,R8std);
  writeln(SetF,ACC_per); writeln(SetF,Th_D);writeln(SetF,Hmax_D);
  writeln(SetF,Hmax_C); writeln(SetF,ACC_DCmax);writeln(SetF,ACC_Acmax);
  writeln(SetF,Tmin_R);
  closefile(setF);
 end;{end Procesure
 ------------------------------}
 procedure TForm1.UpDown1Click(Sender: TObject; Button: TUDBtnType);
 var i :integer;
 begin
  if updown1.position >50 then updown1.position :=50;
  if updown1.position <0 then updown1.position :=0;
  FlowQ:=2000+updown1.position*100;
  Qflow[1]:=FlowQ/3;
  Qflow[2]:=FlowQ/3;
  Qflow[3]:=FlowQ/3;
  Qflow[1]:=Qflow[1]/3600; Qflow[2]:=Qflow[2]/3600;
 Qflow[3]:=Qflow[3]/3600;
  HDmin[1]:=1;HDmin[2]:=1;HDmin[3]:=1;
  HDavr[1]:=1/Qflow[1];HDavr[2]:=1/Qflow[2];HDavr[3]:=1/Qflow[3];
  for i:=1 to 3 do begin
  if Qflow[i] >(1/FMH[i]) then FMH[i]:=1/QFlow[i];
   Lampda[i] :=QFlow[i]*(1-0.5*sqrt(QFlow[i]));
   phai[i] :=(FMH[i]*QFlow[i]);
   phai[i]:= phai[i]-(0.5*(phai[i]-1)*sqrt(QFlow[i]));
  end;
  str((round(FlowQ)),S);
  Label20.caption := S;
 end;{end Procesure
 ------------------------------}
 procedure TForm1.UpDown2Click(Sender: TObject; Button: TUDBtnType);
 Var
  QFH    : single;
 begin
  if updown2.position >60 then updown2.position :=60;
  if updown2.position <0 then updown2.position :=0;
  HGVQ := updown2.position;
  str(HGVQ,S);
  label19.caption := S+'%';
  QFH  := FlowQ*HGVQ/200;
  HGV[1]:=round(QFH/(36*QFlow[1]));
  HGV[2]:=round(QFH/(36*QFlow[2]));
  HGV[3]:=0;
 end;{end Procesure
 ------------------------------}
 procedure TForm1.Button5Click(Sender: TObject);
 begin
  if checkbox2.checked =false then begin
   checkbox2.checked :=true; button5.caption :='Hide Traffic';;
   paintBox1.visible := false;  {paintBox2.visible := false;}
   label65.visible :=True; label66.visible :=true; label67.visible :=true;
   shape1.visible :=True; shape2.visible :=True; shape3.visible :=True;
   btnleft.visible := True; btnright.visible :=True;
   RadioGroup1.enabled:=true;
   graph1;
```

```
  end
  else begin
   if checkbox2.checked =true then begin
   checkbox2.checked :=false;button5.caption :='Show Traffic';
   paintBox1.visible := true;  {paintBox2.visible := true;}
   label65.visible :=false; label66.visible :=false; label67.visible
:=false;
   shape1.visible :=false; shape2.visible :=false; shape3.visible :=false;
   btnleft.visible := false; btnright.visible :=false;
   RadioGroup1.enabled:=false;
   end;
  end;
end;{end Procesure
-------------------------------}
procedure makesure;
begin
   closefile(F);
   Assignfile(F, Fname1);
   {$I-}
   Reset(F);
   {$I+}
   if IOResult = 0 then begin
    if filesize(F)<>0 then begin
     if MessageDlg('Warning... This will delet the existing file?',
        mtwarning, [mbOk, mbCancel], 0) = mrOK then
     begin
      rewrite(F);fname:=fname1;
      if form1.radioGroup3.itemIndex=0 then  begin

Write(F,'Timer,D1_car/km,F1_car/h,S1_km/h,D2_car/km,F2_car/h,S2_km/h,');
      Write(F,'D3_car/km,F3_car/h,S3_km/h,D_car/km,F_car/h,S_km/h,');

Writeln(F,'LC12/km/h,LC23/km/h,LC21/km/h,LC32/km/h,J1_sec,J2_sec,J3_sec,J_
sec');
      end
      else begin

Writeln(F,'Timer,DSpeed1,ABX1,SDX1,V1,VL1,type,DSpeed2,ABX2,SDX2,V2,VL1,ty
pe,DSpeed3,ABX3,SDX3,V3,VL3,type');
      end;
      form1.button6.enabled := false;
      form1.button1.enabled := true;
      form1.checkbox3.checked := true;
      form1.edit5.visible :=true; Form1.label2.visible :=true;
      form1.edit5.text := fname;
      form1.updown15.enabled := true;form1.updown16.enabled := true;
     end
     else begin
      closefile(F);Assignfile(F, Fname);  reset (F);
     end;
    end
    else begin
     rewrite(F);fname:=fname1;
     if form1.radioGroup3.itemIndex=0 then  begin

Write(F,'Timer,D1_car/km,F1_car/h,S1_km/h,D2_car/km,F2_car/h,S2_km/h,');
```

```
Writeln(F,'D3_car/km,F3_car/h,S3_km/h,LC12/km/h,LC23/km/h,LC21/km/h,LC32/k
m/h,J1_sec,J2_sec,J3_sec,J_sec');
     end
     else begin

Writeln(F,'Timer,DSpeed1,ABX1,SDX1,V1,VL1,type,DSpeed2,ABX2,SDX2,V2,VL2,ty
pe,DSpeed3,ABX3,SDX3,V3,VL3,type');
     end;
     form1.button6.enabled := false;
     form1.button1.enabled := true;
     form1.checkbox3.checked := true;
     form1.edit5.visible :=true; Form1.label2.visible :=true;
     form1.edit5.text := fname;
     form1.updown15.enabled := true;form1.updown16.enabled := true;
    end;
   end
   else begin
     rewrite(F);fname:=fname1;
     if form1.radioGroup3.itemIndex=0 then  begin

Write(F,'Timer,D1_car/km,F1_car/h,S1_km/h,D2_car/km,F2_car/h,S2_km/h,');
      Write(F,'D3_car/km,F3_car/h,S3_km/h,D_car/km,F_car/h,S_km/h,');

Writeln(F,'LC12/km/h,LC23/km/h,LC21/km/h,LC32/km/h,J1_sec,J2_sec,J3_sec,J_
sec');
     end
     else begin

Writeln(F,'Timer,DSpeed1,ABX1,SDX1,V1,VL1,type,DSpeed2,ABX2,SDX2,V2,VL2,ty
pe,DSpeed3,ABX3,SDX3,V3,VL3,type');
     end;
     form1.button6.enabled := false;
     form1.button1.enabled := true;
     form1.checkbox3.checked := true;
     form1.edit5.visible :=true; Form1.label2.visible :=true;
     form1.edit5.text := fname;
     form1.updown15.enabled := true;form1.updown16.enabled := true;
   end;
end;{end Procesure
--------------------------------}
procedure TForm1.Button6Click(Sender: TObject);
begin
 Fname1:='';
 if opendialog1.execute then begin
  Fname1:=opendialog1.filename
 end;
 if Fname1<>'' then makesure;
 if checkbox2.checked =true then begin
  if radioGroup1.itemIndex=0 then
   form1.PaintBox1.Canvas.Draw(0,0,road1)
  else form1.PaintBox1.Canvas.Draw(0,0,road);
 end;
 if radioGroup2.itemIndex=0 then
  form1.PaintBox2.Canvas.Draw(0,0,GraphC)
 else form1.PaintBox2.Canvas.Draw(0,0,GraphB);
end;{end Procesure
```

```
--------------------------------}
procedure TForm1.UpDown3Click(Sender: TObject; Button: TUDBtnType);
begin
 if updown3.position >99 then updown3.position :=99;
 if updown3.position <0 then updown3.position :=0;
 LCTime:=updown3.position;
 str(updown3.position,S);
 label18.caption := S;
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown4Click(Sender: TObject; Button: TUDBtnType);
var I : integer;
begin
 if updown4.position >10 then updown4.position :=10;
 if updown4.position <2 then updown4.position :=2;
 loopD:=updown4.position;I:=loopD;
 for I:=1 to loopD do begin
  loopF[I]:=Dstar+((I-1)*(Dend - Dstar)/(loopD-1));
 end;
 i:=LoopD; str(I,S); label16.caption := S;
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown5Click(Sender: TObject; Button: TUDBtnType);
begin
 if updown5.position >99 then updown5.position :=25;
 if updown5.position <0 then updown5.position :=0;
 SpeedT:=updown5.position;
 str(SpeedT,S);
 label24.caption := S+'%';
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown6Click(Sender: TObject; Button: TUDBtnType);
 var
  i :integer;
begin
 if updown6.position >30 then updown6.position :=30;
 if updown6.position <5 then updown6.position :=5;
 SpeedTS:=updown6.position;
 i:=round(int(SpeedTS/10)); str(I,S);
 i:=round(((SpeedTS/10)-int(SpeedTS/10))*10);str(I,S1);
 label27.caption := S+'.'+S1+'%';
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown7Click(Sender: TObject; Button: TUDBtnType);
var i : integer;
begin
 if updown7.position >50 then updown7.position :=50;
 if updown7.position <5 then updown7.position :=5;
 LCTimeS:=updown7.position;
 i:=round(int(LCtimeS/10)); str(I,S);
 i:=round(((LCtimeS/10)-int(LCtimeS/10))*10);str(I,S1);
 label11.caption := S+'.'+S1;
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown8Click(Sender: TObject; Button: TUDBtnType);
begin
 if updown8.position >30 then updown8.position :=30;
```

```
 if updown8.position <2 then updown8.position :=2;
 RCTime:=updown8.position*10;
 str(updown8.position*10,S);
 label12.caption := S;intim:=0;
end;{end Procesure
-------------------------------}
procedure TForm1.UpDown9Click(Sender: TObject; Button: TUDBtnType);
var i :integer;
begin
 if updown9.position >19 then begin
  if ADS >39.8 then begin
   ADS:=39.9;updown9.position:=19;
  end
  else begin
   ADS:=int(ADS)+1;updown9.position:=10;
  end;
 end;
 if updown9.position <10 then begin
  if ADS <10 then begin
   ADS:=10;updown9.position:=10;
  end
  else begin
   ADS:=int(ADS)-1;updown9.position:=19;
  end;
 end;
 ADS:=int(ADS)+(updown9.position-10)/10;
 i:=round(int(ADS)); str(I,S);
 i:=updown9.position-10;str(I,S1);
 label25.caption := S+'.'+S1;
end;{end Procesure
-------------------------------}
procedure TForm1.UpDown10Click(Sender: TObject; Button: TUDBtnType);
var i :integer;
begin
 if updown10.position >90 then updown10.position :=90;
 if updown10.position <0 then updown10.position :=0;
 SDADS:=updown10.position/10;
 i:=round(int(SDADS)); str(I,S);
 i:=round((SDADS-int(SDADS))*10);str(I,S1);
 label39.caption := S+'.'+S1;
end;{end Procesure
-------------------------------}
procedure TForm1.UpDown11Click(Sender: TObject; Button: TUDBtnType);
var i :integer;
begin
 if updown11.position >19 then begin
  if ADSH >29.8 then begin
   ADSH :=29.9;updown11.position:=19;
  end
  else begin
   ADSH:=int(ADSH)+1;updown11.position:=10;
  end;
 end;
 if updown11.position <10 then begin
  if ADSH <10 then begin
   ADSH :=10;updown11.position:=10;
  end
```

```
   else begin
    ADSH:=int(ADSH)-1;updown11.position:=19;
   end;
  end;
  ADSH:=int(ADSH)+(updown11.position-10)/10;
  i:=round(int(ADSH)); str(I,S);
  i:=updown11.position-10;str(I,S1);
  label41.caption := S+'.'+S1;
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown12Click(Sender: TObject; Button: TUDBtnType);
var i :integer;
begin
 if updown12.position >90 then updown12.position :=90;
 if updown12.position <0 then updown12.position :=0;
 SDADSH:=updown12.position/10;
 i:=round(int(SDADSH)); str(I,S);
 i:=round((SDADSH-int(SDADSH))*10);str(I,S1);
 label43.caption := S+'.'+S1;
end;{end Procesure
--------------------------------}
procedure TForm1.RadioGroup2Click(Sender: TObject);
begin
 if radioGroup2.itemIndex=0 then  begin
  LaneRG:=0;
  form1.PaintBox2.Canvas.Draw(0,0,GraphC);
 end
 else begin
  if radioGroup2.itemIndex=1 then LaneRG:=1;
  if radioGroup2.itemIndex=2 then LaneRG:=2;
  if radioGroup2.itemIndex=3 then LaneRG:=3;
  form1.PaintBox2.Canvas.Draw(0,0,GraphB);
 end;
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown13Click(Sender: TObject; Button: TUDBtnType);
begin
 if updown13.position >99 then updown13.position :=25;
 if updown13.position <0 then updown13.position :=0;
 SpeedTH:=updown13.position;
 str(SpeedTH,S);
 label53.caption := S+'%';
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown14Click(Sender: TObject; Button: TUDBtnType);
 var i :integer;
begin
 if updown14.position >30 then updown14.position :=30;
 if updown14.position <5 then updown14.position :=5;
 SpeedTSH:=updown14.position;
 i:=round(int(SpeedTSH/10)); str(I,S);
 i:=round(((SpeedTSH/10)-int(SpeedTSH/10))*10);str(I,S1);
 label55.caption := S+'.'+S1+'%';
end;{end Procesure
--------------------------------}
procedure TForm1.RadioGroup1Click(Sender: TObject);
begin
```

```
  if radioGroup1.itemIndex=0 then  begin
   btnleft.visible := false; btnright.visible :=false;
   graph1;
  end
  else begin
   {posi:=1;}
   btnleft.visible := True; btnright.visible :=True;
   graph1;
  end;
end;{end Procesure
-------------------------------}
procedure TForm1.Button7Click(Sender: TObject);
begin
if LCfactor =1 then begin
 LCfactor :=0;
 Button7.caption:='NO LC'
end
else begin
 LCfactor :=1;
 Button7.caption:='Yes LC'
end;
end;{end Procesure
-------------------------------}
procedure TForm1.UpDown15Click(Sender: TObject; Button: TUDBtnType);
Var I:integer;
begin
 if updown15.position >99 then updown15.position :=99;
 if updown15.position >((9000-Dstar)/50) then updown15.position
:=round((9000-Dstar)/50);
 if updown15.position <2 then updown15.position :=2;
 I:=50*Updown15.position;
 str(I,S); label21.caption := S;
 DEnd:=Dstar+I;
end;{end Procesure
-------------------------------}
procedure TForm1.UpDown16Click(Sender: TObject; Button: TUDBtnType);
var I:integer;
begin
 if updown16.position >99 then updown16.position :=99;
 if (50*(updown15.position+updown16.position))>9000 then
  updown16.position:=180-(updown15.position);
 if updown16.position <1 then updown16.position :=1;
 I:=50*Updown16.position;
 str(I,S); label22.caption := S;
 Dstar:=I;
 DEnd:=Dstar+50*updown15.position;
end;{end Procesure
-------------------------------}
procedure TForm1.UpDown17Click(Sender: TObject; Button: TUDBtnType);
begin
 if updown17.position >90 then updown17.position :=90;
 if updown17.position <0 then updown17.position :=0;
 LChug:=updown17.position;
 str(LChug,S);
 S:=S+'%';
 Label57.caption:=S;
end;{end Procesure
```

```
-------------------------------}
procedure TForm1.UpDown18Click(Sender: TObject; Button: TUDBtnType);
var i : integer;
begin
 if updown18.position >20 then updown18.position :=20;
 if updown18.position <0 then updown18.position :=0;
 StimuT:=Updown18.position;
 i:=round(int(StimuT/2)); str(I,S);
 i:=round(((StimuT/2)-int(StimuT/2))*10);str(I,S1);
 label58.caption := S+'.'+S1;
end;{end Procesure
-------------------------------}
procedure TForm1.Button8Click(Sender: TObject);
begin
if Flowfactor =1 then begin
 Flowfactor :=0;StartTime:=T;
 Button8.caption:='Stop Exp.'
end
else begin
 Flowfactor :=1;
 Button8.caption:='Start Exp.'
end;
end;{end Procesure
-------------------------------}
procedure TForm1.UpDown19Click(Sender: TObject; Button: TUDBtnType);
begin
 if updown19.position >10 then updown19.position :=10;
 if updown19.position <0 then updown19.position :=0;
 Dflow:=updown19.position*100;
 str(DFlow,S);
 Label62.caption := S;
end;{end Procesure
-------------------------------}
procedure TForm1.UpDown20Click(Sender: TObject; Button: TUDBtnType);
begin
 if updown20.position >60 then updown20.position :=60;
 if updown20.position <5 then updown20.position :=5;
 DFtime:=updown20.position;
 str(DFTime,S);
 Label64.caption := S;
end;{end Procesure
-------------------------------}
procedure TForm1.UpDown21Click(Sender: TObject; Button: TUDBtnType);
begin
 if updown21.position >50 then updown21.position :=50;
 if updown21.position <0 then updown21.position :=0;
 flowEnd:=2000+updown21.position*100;
 str(FlowEnd,S);
 Label9.caption := S;
end;{end Procesure
-------------------------------}
procedure TForm1.Button9Click(Sender: TObject);
var i : integer;
begin

Dstar:=3000;Dend:=6000;loopd:=6;RCtime:=60;LChug:=0;LCtime:=30;SpeedT:=10;
  SpeedTH:=5;DFtime:=15;Dflow:=100;FlowEnd:=7000;HGVQ:=10;FlowQ:=2000;
```

```
  LCtimeS:=25;stimuT:=2;ADS:=31.6;ADSH:=23.3;sdADS:=4.6;sdADSH:=1.43;
  SpeedTS:=15;SpeedTSH:=15;R8:=65;R8std:=25;Stimuhead:=30;TTCLCL:=30;
  ACC_per :=20; Th_D:=1.5; Hmax_D:=2.5; Hmax_C:=2.7;
  ACC_DCmax :=-2.0;ACC_ACmax :=0.7; Tmin_R :=30;

  str(ACC_per,S);  label71.caption := S+'%';  Updown26.position:=ACC_per;
  str(Tmin_R,S);   label83.caption := S+' Mean';
Updown32.position:=Tmin_R;

  i:=round(int(Th_D)); str(I,S);i:=round(((Th_D)-int(Th_D))*10);str(I,S1);
  label77.caption := S+'.'+S1+' Mean'; Updown27.position:=round(10*Th_D);
  i:=round(int(Hmax_D)); str(I,S);i:=round(((Hmax_D)-
int(Hmax_D))*10);str(I,S1);
  label78.caption := S+'.'+S1+' Sec'; Updown28.position:=round(10*Hmax_D);
  i:=round(int(Hmax_C)); str(I,S);i:=round(((Hmax_C)-
int(Hmax_C))*10);str(I,S1);
  label80.caption := S+'.'+S1+' Sec'; Updown29.position:=round(10*Hmax_C);
  i:=round(int(ABS(ACC_DCmax))); str(I,S);i:=round((int(ACC_DCmax)-
(ACC_DCmax))*10);str(I,S1);
  label85.caption := '-'+S+'.'+S1; Updown31.position:=round(-
10*ACC_DCmax);
  i:=round(int(ACC_ACmax)); str(I,S);i:=round(((ACC_ACmax)-
int(ACC_ACmax))*10);str(I,S1);
  label84.caption := '-'+S+'.'+S1; Updown30.position:=round(10*ACC_ACmax);

  str(TTCLCL,S);  label47.caption := S+' sec';
Updown22.position:=TTCLCL;
  str(R8,S);  label69.caption := S+'%';   Updown24.position:=R8;
  str(R8std,S);  label70.caption := S+'%';   Updown25.position:=R8std;

  str(Dstar,S);  label22.caption := S;
Updown16.position:=round(Dstar/50);
  str((DEnd-Dstar),S);label21.caption:=S;Updown15.position:=round((DEnd-
Dstar)/50);
  str(loopD,S);  label16.caption := S;   Updown4.position:=round(loopD);
  str(RCtime,S); label12.caption := S;
updown8.position:=round(RCTime/10);
  str(LChug,S);  Label57.caption:=S+'%'; updown17.position:=LChug;
  str(LCTime,S); label18.caption := S;   updown3.position:=LCTime;
  str(SpeedT,S); label24.caption:=S+'%'; updown5.position:= SpeedT;
  str(SpeedTH,S);label53.caption:=S+'%'; updown13.position:= SpeedTH;
  str(DFTime,S); Label64.caption := S;   updown20.position:=DFtime;
  str(DFlow,S);  Label62.caption := S;
updown19.position:=round(Dflow/100);
  str(FlowEnd,S);Label9.caption := S;
updown21.position:=round((flowEnd-2000)/100);
  str(HGVQ,S);   label19.caption:=S+'%'; updown2.position:= HGVQ;
  str(FlowQ,S);Label20.caption:=S;updown1.position:=round((FlowQ-
2000)/100);
  i:=round(int(LCtimeS/10)); str(I,S); i:=round(((LCtimeS/10)-
int(LCtimeS/10))*10);str(I,S1);
  label11.caption := S+'.'+S1;    updown7.position:= LCTimeS;

  i:=round(int(StimuT/2)); str(I,S);i:=round(((StimuT/2)-
int(StimuT/2))*10);str(I,S1);
  label58.caption := S+'.'+S1; Updown18.position:=StimuT;
```

```
  i:=round(int(ADS)); str(I,S);  i:=round((ADS-int(ADS))*10);
str(I,S1);
  label25.caption := S+'.'+S1;   updown9.position:=i+10;
  i:=round(int(ADSH)); str(I,S);i:=round((ADSH-int(ADSH))*10);str(I,S1);
  label41.caption := S+'.'+S1; updown11.position:=i+10;
  i:=round(int(SDADS)); str(I,S);i:=round((SDADS-
int(SDADS))*10);str(I,S1);
  label39.caption := S+'.'+S1;   updown10.position:=round(SDADS*10);
  i:=round(int(SDADSH)); str(I,S);i:=round((SDADSH-
int(SDADSH))*10);str(I,S1);
  label43.caption := S+'.'+S1;  updown12.position:=round(SDADSH*10);
  i:=round(int(SpeedTS/10)); str(I,S); i:=round(((SpeedTS/10)-
int(SpeedTS/10))*10);str(I,S1);
  label27.caption := S+'.'+S1+'%'; updown6.position:=SpeedTS;
  i:=round(int(SpeedTSH/10)); str(I,S); i:=round(((SpeedTSH/10)-
int(SpeedTSH/10))*10);str(I,S1);
  label55.caption := S+'.'+S1+'%'; updown14.position:=SpeedTSH;

  Qflow[1]:=FlowQ/3; Qflow[3]:=FlowQ/3; Qflow[2]:=FlowQ/3;
  Qflow[1]:=Qflow[1]/3600; Qflow[2]:=Qflow[2]/3600;
Qflow[3]:=Qflow[3]/3600;
  HDmin[1]:=1;HDmin[2]:=1;HDmin[3]:=1;
  HDavr[1]:=1/Qflow[1];HDavr[2]:=1/Qflow[2];HDavr[3]:=1/Qflow[3];
  count:=1;
  while count <=3 do begin
   if Qflow[count] >(1/FMH[count]) then FMH[count]:=1/QFlow[count];
   Lampda[count] :=QFlow[count]*(1-0.5*sqrt(QFlow[count]));
   phai[count] :=(FMH[count]*QFlow[count]);
   phai[count]:= phai[count]-(0.5*(phai[count]-1)*sqrt(QFlow[count]));
   count:=count+1;
  end;
  HGV[1]:=round((FlowQ*HGVQ/200)/(36*QFlow[1]));
  HGV[2]:=round((FlowQ*HGVQ/200)/(36*QFlow[2]));
  HGV[3]:=0;
  count:=1;
  while count<=loopD do begin
   loopF[count]:=Dstar+((count-1)*(Dend - Dstar)/(loopD-1));
   count:=count+1;
  end;
  count :=0;
end;{end Procesure
--------------------------------}
procedure TForm1.AppDeactivate(Sender: TObject);
begin
 Application.minimize;
end;{end Procesure
--------------------------------}
procedure TForm1.ScrollBar1Change(Sender: TObject);
begin
 str(100-scrollbar1.position,S);
 Label37.caption := S+'%';
end;{end Procesure
--------------------------------}
procedure TForm1.RadioGroup3Click(Sender: TObject);
begin
 if radioGroup3.itemIndex=1 then  begin
  radiogroup2.visible :=false;
```

```
  end
  else begin
   radiogroup2.visible :=True;
  end;
 end;{end Procesure
--------------------------------}
procedure TForm1.UpDown22Click(Sender: TObject; Button: TUDBtnType);
begin
 if updown22.position >99 then updown22.position :=99;
 if updown22.position <0 then updown22.position :=0;
 TTCLCL:=updown22.position;
 str(updown22.position,S);
 label47.caption := S+' sec';
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown24Click(Sender: TObject; Button: TUDBtnType);
begin
 if updown24.position <50 then updown24.position :=50;
 R8 := updown24.position;
 if (100-R8)<R8std then begin
  R8std :=100-R8;
  str(R8std,S);label70.caption := S+'%';
  updown25.position:=R8std;
 end;
 str(R8,S);
 label69.caption := S+'%';
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown25Click(Sender: TObject; Button: TUDBtnType);
begin
 if updown25.position >50 then updown25.position :=50;
 if updown25.position >(100-R8) then updown25.position :=(100-R8);
 if updown25.position >R8 then updown25.position :=R8;
 if updown25.position <0 then updown25.position :=0;
 R8std :=updown25.position;
 str(R8std,S);
 label70.caption := S+'%';
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown26Click(Sender: TObject; Button: TUDBtnType);
begin
 if updown26.position >99 then updown26.position :=99;
 if updown26.position <0 then updown26.position :=0;
 ACC_per :=updown26.position;
 str(ACC_per,S); Label71.caption := S+'%';
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown27Click(Sender: TObject; Button: TUDBtnType);
var i:integer;
begin
 if updown27.position >25 then updown27.position :=25;
 if updown27.position <10 then updown27.position :=10;
 Th_D:=updown27.position/10;
 i:=round(int(Th_D)); str(I,S);
 i:=round(((Th_D)-int(Th_D))*10);str(I,S1);
 label77.caption := S+'.'+S1+' Mean';
end;{end Procesure
```

```
--------------------------------}
procedure TForm1.UpDown28Click(Sender: TObject; Button: TUDBtnType);
var i : integer;
begin
 if updown28.position >30 then updown28.position :=30;
 if updown28.position <20 then updown28.position :=20;
 Hmax_D:=updown28.position/10;
 i:=round(int(Hmax_D)); str(I,S);
 i:=round(((Hmax_D)-int(Hmax_D))*10);str(I,S1);
 label78.caption := S+'.'+S1+' sec';
 if Hmax_C < Hmax_D then begin
  Hmax_C:=Hmax_D;
  updown29.position:=updown28.position;
  label80.caption := S+'.'+S1+' sec';
 end;
 if Th_D > Hmax_D then begin
  Th_D:=Hmax_D;
  updown27.position:=updown28.position;
  label77.caption := S+'.'+S1+' Mean';
 end;
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown29Click(Sender: TObject; Button: TUDBtnType);
var i : integer;
begin
 if updown29.position >30 then updown29.position :=30;
 if updown29.position <20 then updown29.position :=20;
 Hmax_C:=updown29.position/10;
 i:=round(int(Hmax_C)); str(I,S);
 i:=round(((Hmax_C)-int(Hmax_C))*10);str(I,S1);
 label80.caption := S+'.'+S1+' sec';
 if Hmax_C < Hmax_D then begin
  Hmax_D:=Hmax_C;
  updown28.position:=updown29.position;
  label78.caption := S+'.'+S1+' sec';
 end;
 if Th_D > Hmax_C then begin
  Th_D:=Hmax_C;
  updown27.position:=updown29.position;
  label77.caption := S+'.'+S1+' Mean';
 end;
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown32Click(Sender: TObject; Button: TUDBtnType);
begin
 if updown32.position >60 then updown32.position :=60;
 if updown32.position <10 then updown32.position :=10;
 Tmin_R :=updown32.position;
 str(updown32.position, S);
 label83.caption := S+ ' Mean';
end;{end Procesure
--------------------------------}
procedure TForm1.UpDown31Click(Sender: TObject; Button: TUDBtnType);
var i  : integer;
begin
 if updown31.position >30 then updown31.position :=30;
 if updown31.position <5 then updown31.position :=5;
```

```
  ACC_DCmax :=0.1*updown31.position;
  i:=round(int(ACC_DCmax)); str(I,S);
  i:=round(((ACC_DCmax)-int(ACC_DCmax))*10);str(I,S1);
  label85.caption := '-'+S+ '.'+S1;
  ACC_DCmax:=-1*ACC_DCmax;
 end;{end Procesure
 -------------------------------}
 procedure TForm1.UpDown30Click(Sender: TObject; Button: TUDBtnType);
 var i :integer;
 begin
  if updown30.position >10 then updown30.position :=10;
  if updown30.position <5 then updown30.position :=5;
  ACC_ACmax :=0.1*updown30.position;
  i:=round(int(ACC_ACmax)); str(I,S);
  i:=round(((ACC_ACmax)-int(ACC_ACmax))*10);str(I,S1);
  label84.caption := S+ '.'+S1;
 end;{end Procesure
 -------------------------------}
 procedure TForm1.RadioGroup4Click(Sender: TObject);
 begin
  if radioGroup4.itemIndex=0 then ACC_Formula:=0 else ACC_Formula:=1;
 end;{end Procesure
 -------------------------------}
 End.
 {------***********************************************************-------}
 unit Variab_ACC;
 interface
  type
    vptr = ^vehicle;

    featur = record   {for Vehicl}
     Number    : word;
     lenV      : word;
     widV      : word;
     kind      : byte;
            end;
    {-----------------------------}
    featur1 = record   {for Driver}
     dspeed    : single;{Desierd speed}
     Cdspeed   : single;{current Desierd speed}
     HBLC      : Single; {Headway threshold to be forced to do LC to the
Left }
     RAND      : array[1..8] of single;
     NULL      : single;
     NULL1     : single;
            end;
    {-----------------------------}
    featur2 = record   {for ACC}
     EQ       : char;      { Equiped with ACC either Y or N}
     Opp      : string[2];{ opperation signe either on or of}

     Th_D      : single; { desired time headway to be adopted under distance
control mode [normaly distributed STD =0.2sec]}
     Thmax_D   : single; { maximum time headway to enter distance control
mode [2.5sec]}
     Thmax_C   : single; { maximum time headway to remain distance control
mode [2.75]}
```

```
    Tmin_R    : single; { minimum time between leaving and re-entering
distance control mode;
                          also minimum time to be spent in distance control
model before an unforced lane change is made.
                          The model will adapt the Lc time manuevre }

    ACmax     : single;{ the maximum acceleration [0.75m/s2]}
    DCmax     : single;{ the maximum  deceleration [-2m/s2]}
    Dc_Eng    : single;{ the deceleration of the vehicle under engine
braking   [DC = 0.03169*V -0.132](m/s2)}

    Vmin_D    : single;{ minimum speed to enter distance control mode
[13.89 m/s]}
    Vmax_D    : single;{ maximum speed in excess of desired speed for
remaining in
                          distance control mode [1.1*desiered speed]}
    Vmin_Ac   : single;{ minimum speed for any acceleration uder distance
control mode}
    Timer     : single;{the timer  for the switch on and off for the ACC}
 end;
   {-------------------------------}
 vehicle =record
    veh             : featur;
    dvr             : featur1;
    ACC             : featur2;
    overtake        : byte; { 0: no overtaking
                              1: overtake to the right
                              2: overtake to the left
                             11: He is during overtaking to the right
                             22: He is during overtaking to the left       }
    overtake1       : byte;{ 1: Has stimuluos to the right                     }
    overtake2       : byte;{ 1: Has stimulous to the left                      }
    OvertakeTime    : single;{ Lane changing time}
    LCT             : integer; { the maximum length of the LC maneuver in
secunds}

    NO_ahead,ABXTimer                          : word;
    olane,lane,Flane                           : byte;
    factor,factor2,factor3,factor4             : byte;
    factorth,F2,F3                             : byte;
    TTA,TTAmin,CLDV,OPDV,stimu                 : single;
    x,v,dc                                     : array[0..12] of Single;
    theta,PDC,EST,odv1                         : array[0..2] of Single;
    DVper                                      : array[0..2] of char;

    ahead,ah_right,ah_left                     : vptr;
    back ,ba_right,ba_left                     : vptr;
    takeveh                                    : vptr;

 end;
var
 recorder            : string[2];
 S,S1                : string;
 Gd,Gm               : integer;
 loopD,Dstar,Dend    : integer;
 LCTime,LCHug        : integer;
```

```
 LCtimeS,stimuT      : integer;
 SpeedT,SpeedTS      : Integer;
 SpeedTH,SpeedTSH    : Integer;
 RCTime,DFTime       : Integer;
 intim,Starttime     : single;
 C                   : char;
 count,turn,F2,F3    : byte;
 number,LCfactor     : word;
 posi,Flowfactor     : word;
 ACC_Formula         : word;
 laneR,lane          : byte;
 laneRG              : byte;
 laneT1,lanet2       : byte;
 Dt,theta            : single;
 T,Speed,Space       : single;
 Rand,mean,std       : Single;
 Tmin_R,ACC_per      : Integer;
 Th_D                : Single;
 ACC_ACmax           : Single;
 ACC_DCmax           : Single;
 Hmax_D,Hmax_C       : Single;
 p1,p2,p3            : vptr ;
 FlowQ,DFlow         : integer;
 FlowEnd             : integer;
 HGVQ,TTCLCL         : word;
 StimuHead           : word;
 R8,R8std            : word;
 ADS                 : single;
 sdADS               : single;
 ADSH                : single;
 sdADSH              : single;
 LCnumber            : array[1..4] of integer;
 density1            : array[0..3,1..10] of Single;
 Speed1              : array[1..3,1..10] of Single;
 loopF               : array[1..10]of Single;
 counter1            : array[1..3,1..10] of integer;
 start               : array[0..4] of vptr;
 lastv               : array[0..4] of vptr;
 HGV                 : array[1..3] of word;
 QFlow               : array[1..3] of single;
 Lampda              : array[1..3] of single;
 Phai                : array[1..3] of single;
 FMH                 : array[1..3] of single;
 HDmin,HDavr         : array[1..3] of single;
 PNorm               : array[0..100]of single;
 lognorm1            : array[3..7] of single;
 norm1               : array[0..21]of single;
 TF                  : array[1..3,1..2] of single;
 DVClose             : array[1..5] of single;
 DVOpen              : array[1..3] of single;
implementation
end.
{------***************************************************-------}
unit Randno_ACC;
interface
 uses variab_ACC;
```

```
  function Lognormal :single;
  function Normal    :single;
  Function Maxac (var V,r1:single; K :byte) :single;
 implementation
 {-----------------------------------------------------------------}
 Function Maxac (var V,r1:single; K :byte) :single;
 begin
  case K of
  1,2:begin
          if V <= 22.9 then
            Maxac := 1.2+r1
          else begin
           if V <= 27.5 then
             Maxac := 0.9+r1
           else begin
             Maxac := 0.55+r1;
           end;
          end;
      end;
  3,4:begin
          if V <= 22.9 then
            Maxac := 0.75*(1.2+r1)
          else begin
           if V <= 27.5 then
             Maxac := 0.75*(0.9+r1)
           else begin
             Maxac := 0.75*(0.55+r1);
           end;
          end;
      end;
  end;
 end; { Function Maxac
 -----------------------------------------------------------------}
 function Lognormal :single;
 var Y1,X1,r1,r2,N1,N2 : single;
 begin
   Y1:=ln(sqr(std/mean)+1);
   X1:=ln(mean)-Y1/2;
   r1:=0.0;r2:=0.0;
   repeat
    repeat r1:=random;r2:=random; until (r1<>0.0)and(r2<>0.0);
    N1:=sqrt(-2*Y1*ln(r1))*cos(2*pi*r2);
    N2:=sqrt(-2*Y1*ln(r1))*sin(2*pi*r2);
   until ((ABS(N1)<(sqrt(Y1)*2))and(ABS(N2)<(Sqrt(Y1)*2)));
   Lognormal:=exp(X1+N1);
   case turn of
    3: lognorm1[3]:=exp(X1+N2);
    4: lognorm1[4]:=exp(X1+N2);
    5: lognorm1[5]:=exp(X1+N2);
    6: lognorm1[6]:=exp(X1+N2);
    7: lognorm1[7]:=exp(X1+N2);
   end;
 end;{end Procesure
 --------------------------------}
 function Normal:single;
 var r1,r2,N1,N2 : single;
 begin
```

```
 r1:=0.0;r2:=0.0;
 repeat
  repeat r1:=random;r2:=random; until (r1<>0.0)and(r2<>0.0);
  N1:=sqrt(-2*sqr(std)*ln(r1))*cos(2*pi*r2);
  N2:=sqrt(-2*sqr(std)*ln(r1))*sin(2*pi*r2);
 until (((ABS(N1)<(2.5*std))and(ABS(N2)<(2.5*std))))or(turn=9);
 Normal:=mean+N1;
 case turn of
   0: norm1[0]:=1000;
   1: norm1[1]:=mean + N2;{TTA}
   2: norm1[2]:=mean + N2;{Car Length}
   3: norm1[3]:=mean + N2;{Car Width}
   4: norm1[4]:=mean + N2;{Car Dspeed lane 1}
   14: norm1[14]:=mean + N2;{Car Dspeed lane 2}
   15: norm1[15]:=mean + N2;{Car Dspeed lane 3}
   5: norm1[5]:=mean + N2;{HGV Length}
   6: norm1[6]:=mean + N2;{HGV W}
   7: norm1[7]:=mean + N2;{HGV Dspeed}
   8: norm1[8]:=mean + N2;{DFD}
   9: norm1[9]:=mean + N2;{Vehicle kind}
  10: norm1[10]:=mean + N2;{Vehicle kind}
  11: norm1[11]:=mean + N2; { Speed Tolerance car}
  13: norm1[13]:=mean + N2; { Speed Tolerance HGV}
  12: norm1[12]:=mean + N2; {LC Maneuver }
  16: norm1[16]:=mean + N2; { Desiered Speed Tolerance }
  17: norm1[17]:=mean + N2; { NULLP }
  18: norm1[18]:=mean + N2; { R8 }
  19: norm1[19]:=mean + N2; { ACC_per}
  20: norm1[20]:=mean + N2; { the ACC desiered headway}
  21: norm1[21]:=mean + N2; { the ACC Recycle time}
 end;
end;{end Procesure
--------------------------------}
begin
end.
{------*******************************************************-------}
unit Data_ACC;
interface
uses variab_ACC;
 procedure getdata;

implementation

procedure getdata;
begin
 PNorm [0]:=10;
 pnorm [1] :=2.576;pnorm [2] :=2.326;pnorm [3] :=2.170;pnorm [4] :=2.054;
 pnorm [5] :=1.959;pnorm [6] :=1.880;pnorm [7] :=1.811;pnorm [8] :=1.750;
 pnorm [9] :=1.695;pnorm [10]:=1.644;pnorm [11]:=1.598;pnorm [12]:=1.554;
 pnorm [13]:=1.514;pnorm [14]:=1.475;pnorm [15]:=1.439;pnorm [16]:=1.405;
 pnorm [17]:=1.372;pnorm [18]:=1.340;pnorm [19]:=1.310;pnorm [20]:=1.281;
 pnorm [21]:=1.253;pnorm [22]:=1.226;pnorm [23]:=1.200;pnorm [24]:=1.174;
 pnorm [25]:=1.150;pnorm [26]:=1.126;pnorm [27]:=1.103;pnorm [28]:=1.080;
 pnorm [29]:=1.058;pnorm [30]:=1.036;pnorm [31]:=1.015;pnorm [32]:=0.994;
 pnorm [33]:=0.974;pnorm [34]:=0.954;pnorm [35]:=0.934;pnorm [36]:=0.915;
 pnorm [37]:=0.896;pnorm [38]:=0.877;pnorm [39]:=0.859;pnorm [40]:=0.841;
 pnorm [41]:=0.823;pnorm [42]:=0.806;pnorm [43]:=0.789;pnorm [44]:=0.772;
```

```
 pnorm [45]:=0.755;pnorm [46]:=0.738;pnorm [47]:=0.722;pnorm [48]:=0.706;
 pnorm [49]:=0.690;pnorm [50]:=0.674;pnorm [51]:=0.658;pnorm [52]:=0.643;
 pnorm [53]:=0.628;pnorm [54]:=0.612;pnorm [55]:=0.597;pnorm [56]:=0.583;
 pnorm [57]:=0.568;pnorm [58]:=0.553;pnorm [59]:=0.539;pnorm [60]:=0.524;
 pnorm [61]:=0.510;pnorm [62]:=0.495;pnorm [63]:=0.483;pnorm [64]:=0.467;
 pnorm [65]:=0.454;pnorm [66]:=0.440;pnorm [67]:=0.426;pnorm [68]:=0.413;
 pnorm [69]:=0.400;pnorm [70]:=0.386;pnorm [71]:=0.371;pnorm [72]:=0.358;
 pnorm [73]:=0.346;pnorm [74]:=0.330;pnorm [75]:=0.318;pnorm [76]:=0.309;
 pnorm [77]:=0.290;pnorm [78]:=0.280;pnorm [79]:=0.265;pnorm [80]:=0.253;
 pnorm [81]:=0.240;pnorm [82]:=0.231;pnorm [83]:=0.213;pnorm [84]:=0.200;
 pnorm [85]:=0.188;pnorm [86]:=0.175;pnorm [87]:=0.163;pnorm [88]:=0.150;
 pnorm [89]:=0.138;pnorm [90]:=0.125;pnorm [91]:=0.113;pnorm [92]:=0.100;
 pnorm [93]:=0.088;pnorm [94]:=0.075;pnorm [95]:=0.063;pnorm [96]:=0.050;
 pnorm [97]:=0.038;pnorm [98]:=0.025;pnorm [99]:=0.013;pnorm [100]:=0.0;
 {--------------------------------}
  DVClose[1]:=39.4;DVClose[2]:=63.4;DVClose[3]:=80.7;
  DVClose[4]:=92;  DVClose[5]:=97.9;
  DVOPen[1]:=50.1;  DVOPen[2]:=71.6;  DVOPen[3]:=86.2;
end;{end Procesure
--------------------------------}
begin
end.
{-----*****************************************************-------}
unit Folow_ACC;
interface
uses
 variab_ACC,gener_ACC,RANDNO_ACC;

procedure follow;
procedure follow1;
procedure follow2;
procedure follow_ACC;

implementation

procedure follow1;
var
  DFD,ABX,SDX,AX,TTC,TTA,TTAmin,DTC,NULL,NULL1,NULLP,lenV : Single;
  V,r1,vd,EST,DCmax,VL,OPDV,CLDV,F1                       : Single;
  ov2,ox2,V2,X2,ACmax2,dc2,odc2,odc21,odc22               : Single;
  v1,x1,ox1,ov1,odc1,odc12,PDC                            : Single;
  odv,dv,dx,odx,Eodv,Eodx,DCeng                           : Single;
  factor, factor2,factorth,factor4,i                      : byte;
  DVper                                                   : char;
label 20;
begin
 p2:=p1^.ahead;
 lenV  :=0.0;NULL1:=0.0;NULL:=0.0;TTAmin:=0.0;DCmax:=0.0;acmax2:=0.0;
 DFD   :=0.0;ABX  :=0.0;SDX :=0.0;TTA   :=0.0;DTC  :=0.0;V     :=0.0;
 r1    :=0.0;Vd   :=0.0;EST :=0.0;CLDV  :=0.0;F1   :=0.0;
 ov2   :=0.0;ox2  :=0.0;v2  :=0.0;x2    :=0.0;OPDV :=0.0;dc2   :=0.0;
 odc2  :=0.0;odc21:=0.0;v1  :=0.0;odc22 :=0.0;x1   :=0.0;ox1   :=0.0;
 ov1   :=0.0;odc1 :=0.0;odv :=0.0;odc12 :=0.0;dv   :=0.0;dx    :=0.0;
 odx   :=0.0;Eodx :=0.0;Eodv:=0.0;factor4:=0;
 factor :=0;factor2:=0; factorth:=0;DVper  :='s';

 ov2:= p1^.v[4]; ox2:= p1^.x[4];
```

```
  v2 := p1^.v[2] ; x2 := p1^.x[2] ;
  odc2  := p1^.dc[2];
  odc21 := p1^.dc[4];
  odc22 := p1^.dc[6];

 {ov2:= p1^.ov; ox2:= p1^.ox;
  v2 := p1^.v ; x2 := p1^.x ;
  odc2  := p1^.odc1;
  odc21 := p1^.odc2;
  odc22 := p1^.odc3;}

  Acmax2:= maxac(p1^.v[2],p1^.dvr.RAND[4],p1^.veh.kind);
  NULL  := p1^.dvr.NULL; NULL1 := p1^.dvr.NULL1;

  if (p1^.V[1]*3.6)>=50 then begin
   DCeng:=-0.0317*p1^.V[1] +0.132;
  end else begin
   DCeng:=NULL1;
  end;
  repeat
   turn:=17;
   if norm1[turn]<>1000 then begin
    NULLP:=norm1[turn];norm1[turn]:=1000;
   end
   else begin
    mean:=0.0;std:=0.07;NULLP:=Normal;
   end;
  until (ABS(NULLP)<0.15);
 {---- Determining the Desierd Speed -------}
  if p1.overtake =11 then  vd:=(1+0.5*p1^.dvr.RAND[6])*p1^.dvr.dspeed else
vd:=p1^.dvr.dspeed;
  vd:=p1^.dvr.dspeed;
   case p1^.lane of
  1: begin
      p2:=p1^.ah_right;p3:=nil;
      while(p2<>nil)and((p2^.x[4]-p1^.x[4])<100) do begin
p3:=p2;p2:=p2^.ahead; end;
      if p3<>nil then begin
       dx:=0;dv:=0;p2:=p3;
       while (p2<>nil)and(p2^.x[4]>p1^.x[4]) do begin
        dv:=dv+p2^.v[4];dx:=dx+1;p2:=p2^.back;
       end;
       if dx>2 then begin
        dv:=dv/dx;
        if (dv+6)<Vd then begin
         {if (dv+4)>=15 then vd:=dv+4+p1^.dvr.Rand[7] else
vd:=15+p1^.dvr.Rand[7];}
         vd:=dv+4+p1^.dvr.Rand[7];
        end;
       end;
      end;
     end;
  2: begin
      p2:=p1^.ah_right;p3:=nil;
      while(p2<>nil)and((p2^.x[4]-p1^.x[4])<100) do begin
p3:=p2;p2:=p2^.ahead; end;
      if p3<>nil then begin
```

```
       dx:=0;dv:=0;p2:=p3;
       while (p2<>nil)and(p2^.x[4]>p1^.x[4]) do begin
        dv:=dv+p2^.v[4];dx:=dx+1;p2:=p2^.back;
       end;
       if dx>2 then begin
        dv:=dv/dx;
        if (dv+6)<Vd then begin
         {if (dv+4)>=15 then vd:=dv+4+p1^.dvr.Rand[7] else
vd:=15+p1^.dvr.Rand[7];}
         vd:=dv+4+p1^.dvr.Rand[7];
        end;
       end;
      end;
     end;
(*
 3: begin
      p2:=p1^.ah_left;p3:=nil;
      while(p2<>nil)and((p2^.x[4]-p1^.x[4])<100) do begin
p3:=p2;p2:=p2^.ahead; end;
      if p3<>nil then begin
       dx:=0;dv:=0;p2:=p3;
       while (p2<>nil)and(p2^.x[4]>p1^.x[4]) do begin
        dv:=dv+p2^.v[4];dx:=dx+1;p2:=p2^.back;
       end;
       if dx>2 then begin
        dv:=dv/dx;
        if (dv+6)<Vd then begin
         if (dv+4)>=15 then vd:=dv+4+p1^.dvr.Rand[7] else
vd:=15+p1^.dvr.Rand[7];
         {vd:=dv+4+p1^.dvr.Rand[7];}
        end;
       end;
      end;
     end;
     *)
 end;
 p1^.dvr.Cdspeed:=vd;
 p2:=p1^.ahead;
 factor4 := p1^.factor4;  { looking for more than one vehicle ahead}
 factor  := p1^.factor;    { the type of process }
 {factor1 := p1^.factor1;   { the brake light indecator}
 factor2 := p1^.factor2;   { the delay time for perceive braking}
 factorth:= p1^.factorth;
{-------------------------}
 if p1 = lastv[p1^.lane] then begin
  factor:=0;
  dc2 := Acmax2;
  VL := v2 + 0.25 * dc2+ 0.25 * odc2;
  if (VL > vd)and(VL < vd+1.15)and(odc2>0) then dc2 :=ABS(NULLP);
  if (VL > vd)and(VL < vd+1.15)and(odc2<0) then dc2 :=-1*ABS(NULLP);
  if (VL < vd)and(VL > vd-1.15)and(odc2<0) then dc2 :=-1*ABS(NULLP);
  if (VL < vd)and(VL > vd-1.15)and(odc2>0) then dc2 :=ABS(NULLP);
  if (VL > (vd+1.15))then dc2 :=NULL1;
  if (VL < (vd-1.15))and (dc2>-0.2) then dc2 :=NULL;
  goto 20;
 end;
 if factor <=1 then
```

```
    p1^.stimu :=0
   else begin
    p1^.stimu:=p1^.stimu+(1/(StimuT*120));
    if p1^.stimu > 1 then p1^.stimu:=1;
   end;

   {-------------------------}
   if p2 <> nil then begin
   { -------------- Entering Data  -----------}
    v1 := p2^.v[2];x1 := p2^.x[2];ov1 := p2^.v[4];ox1 := p2^.x[4];
    lenV := p2^.veh.lenV/100;
    odc1  :=p2^.dc[4];
    odc12 :=p2^.dc[6];
    dx:=x1-x2-lenV; odx:=ox1-ox2-lenV; dv :=v1-v2; odv:=ov1-ov2;
 {----------------------------------}
    if p1^.NO_ahead <> p2^.veh.Number then begin
     p1^.F2:=0;p1^.F3:=0;
     repeat
      turn:=1;
      if norm1[turn]<>1000 then begin
       p1^.TTA:=norm1[turn];norm1[turn]:=1000;
      end
      else begin
       mean:=11.7326;std:=2.292;p1^.TTA:=Normal;
      end;
     until (p1^.TTA>=8);
     repeat
      turn:=7;
      if lognorm1[turn]<>1000 then begin
        p1^.TTAmin:=lognorm1[turn];lognorm1[turn]:=1000;
      end
      else begin
       mean:=8.41; std:=2.1;
       p1^.TTAmin:=Lognormal;
      end;
     until ((p1^.TTAmin>=6) and (p1^.TTAmin<=(p1^.TTA-2)));
     p1^.NO_ahead := p2^.veh.Number;
     P1^.dvr.NULL:=ACnull; P1^.dvr.NULL1:=DCnull(space,speed);
     NULL  := p1^.dvr.NULL; NULL1 := p1^.dvr.NULL1;
   end;
   TTA:=p1^.TTA; TTAmin:=p1^.TTAmin;F2:=p1^.F2;F3:=p1^.F3;
   {----------------------------------}
   if ov1 > 1.0 then begin
    IF p2^.ahead <> nil then begin
     if (p2^.factor<>0)and((p2^.ahead^.dc[4]<-0.7)or(p2^.ahead^.v[4]<5))
then begin
      if factor4 =0 then factor4 :=1 else factor4 :=2;
     end else factor4 :=0;
    End
    Else factor4 :=0;
   end
   else factor4 :=0;
   {----------------------------------}
   theta := (0.04*odv*(p2^.veh.widV))/(4*sqr(odx)+
sqr((p2^.veh.widV)/100));
   space:= odx;Speed:= odv;
   DVper:= DVperception(space,speed,theta,p1^.theta[2],p1^.DVper[2]);
```

```
for i:=2 downto 1 do begin
 p1^.theta[i]:=p1^.theta[i-1];p1^.DVper[i]:=p1^.DVper[i-1];
end;
p1^.theta[0]:=theta;p1^.DVper[0]:=DVper;
AX:=1.5+1.5*random;
if p1^.ACC.EQ='Y' then p1^.ACC.Timer:=p1^.ACC.Timer+0.125;
if ((p1^.ACC.EQ ='Y')and (p1^.ACC.Opp ='of')and(p1^.ACC.Timer
<=p1^.ACC.Tmin_R))
   or(p1^.ACC.EQ ='N')then begin
 if factor =0 then p1^.dvr.rand[1]:=0.5;
 speed:=ov1;ABX:=ABXthresh(speed,p1^.dvr.rand[1]);
 space:=ABX;SDX:=SDXthresh(space,speed,p1^.dvr.rand[2]);
 if ABX < (AX+2) then ABX:=AX+2;
 if (SDX-ABX)<5 then begin
  SDX := ABX+5;
  DFD := ABX+2.5;
 end
 else begin DFD:=(SDX+ABX)/2;end;
end
else begin
 speed:=ov1;
 DFD := (speed*p1^.ACC.Th_D)-lenV; ABX := (DFD*0.8);
 space:=ABX;SDX:=SDXthresh(space,speed,p1^.dvr.rand[2]);
 if ABX < (AX+2) then ABX:=AX+2;DFD:=ABX*1.25;
 if (SDX-DFD)<5 then SDX := DFD+5;
end;
CLDV:=p1^.CLDV; OPDV:=p1^.OPDV;
NULL  := p1^.dvr.NULL; NULL1 := p1^.dvr.NULL1;
 {----------------------------------}
if (ov2 <=0.0) and (ov1<=0.0) then begin
  dc2:=0.0;p1^.factor3:=1;{p1^.PDC:=0.0;}
  for i:=0 to 2 do p1^.PDC[i]:=0.0;
  if odx> AX then begin dc2:= null; p1^.factor3:=0;end;
  goto 20;
end;
if (p1^.factor3 >0) and (p1^.factor3 <5) then  begin
 p1^.factor3 := p1^.factor3 +1;
 dc2 := 0.0; p1^.factor := 0;{p1^.PDC:=0.0;}
 for i:=0 to 2 do p1^.PDC[i]:=0.0;
 goto 20;
end;
{-----------------------}
PDC := p1^.PDC[2];{ PDC := p1^.PDC;}
{----------------------}
 if (DVper='c')and (odv>=0.0) then DVper:='s';
 if (DVper='o')and (odv<=0.0) then DVper:='s';
{--------------------------------}
 if DVper='c' then begin
  PDC :=0.0;
  if (ov1 > 0.0) then begin
   if (Eodx>ABX) then begin
    if(odc12<=DCeng)and((factor2=2)or(factor2=22))then begin
factor2:=22;PDC:=odc12;end;
    if (odc12<=DCeng)and(factor2<>22)then factor2 :=2 ;
    if (odc12>DCeng)and(factor2=22) then factor2 :=2 ;
    if (odc12>DCeng)and(factor2<>22) then factor2 :=0 ;
   end;
```

```
     if (Eodx<=ABX) then begin
      if(odc12<=-0.25)and((factor2=2)or(factor2=22))then begin
factor2:=22;PDC:=odc12;end;
      if (odc12<=-0.25)and(factor2<>22)then factor2 :=2 ;
      if (odc12>0.0)and(factor2 =22) then factor2 :=2 ;
      if (odc12>0.0)and(factor2 <>22) then factor2 :=0 ;
     end;
    end
    else begin
     factor2 := 0;
    end;
   end;
{-----------------------------------}
   if DVper='o' then begin
    PDC :=0.0;
    if (ov1 > 0.0) then begin
     if (Eodx>ABX) then begin
      if(odc12>=0.5)and((factor2=1)or(factor2=11))then begin
factor2:=11;PDC:=odc12;end;
      if (odc12>=0.5)and(factor2<>11)then factor2 :=1 ;
      if (odc12<0.5)and(factor2=11) then factor2 :=1 ;
      if (odc12<0.5)and(factor2<>11) then factor2 :=0 ;
     end;
     if (Eodx<=ABX) then begin
      if(odc12>=0.25)and((factor2=1)or(factor2=11))then begin
factor2:=11;PDC:=odc12;end;
      if (odc12>=0.25)and(factor2<>11)then factor2 :=1 ;
      if (odc12< 0.25)and(factor2=11) then factor2 :=1 ;
      if (odc12< 0.25)and(factor2<>11) then factor2 :=0 ;
     end;
    end
    else begin
     factor2 := 0;
    end;
   end;

{-----------------------------------}
   if DVper<>'s' then begin
    Eodx:=odx+odv*0.5+(-odc2)*0.125;
    if (odv<=-0.5)and (odx>ABX)and(DVper='c') then begin
     TTC:= -odx/odv; est:=(-0.18+0.767*TTC);
     if (Est < 1)or(EST > TTC) then Est:=TTC;
     if (EST < 0.4*TTc) then Est :=0.4*TTC;
     Eodv :=-odx/EST;
     if
(odx>(ABX))and(p1^.EST[2]<20)and((factor2=11)or(factor2=22))and(ov1>5)
then
      PDC := odc22 -((EST-
p1^.EST[2]+0.25)*p1^.odv1[2]/((EST+0.125)*0.25));

     if (PDC < (odc12*1.3))and(odc12<0.0) then PDC :=1.3*odc12;
     {p1^.EST:=EST;p1^.odv1:=odv;}
     for i:=1 to 2 do begin
      p1^.EST[i]:=p1^.EST[i-1];p1^.odv1[i]:=p1^.odv1[i-1];
     end;
    p1^.EST[0]:=EST;p1^.odv1[0]:=odv;
   end
```

```
    else begin
     EST:=100;Eodv :=odv; {p1^.EST:=EST;p1^.odv1:=odv;}
     for i:=1 to 2 do begin
      p1^.EST[i]:=p1^.EST[i-1];p1^.odv1[i]:=p1^.odv1[i-1];
     end;
     p1^.EST[0]:=EST;p1^.odv1[0]:=odv;
    end;
{-----------------------------------}
  end
  else begin
   if (ov1 <= 0.0) then begin
    PDC:=0.0;Factor2:=0;{p1^.PDC :=0.0;}
    for i:=0 to 2 do p1^.PDC[i]:=0.0;
   end;
   Eodx:=odx+odv*0.5+(-odc2)*0.125;
   EST:=100;Eodv :=odv;{p1^.EST:=EST;p1^.odv1:=odv;}
   for i:=1 to 2 do begin
    p1^.EST[i]:=p1^.EST[i-1];p1^.odv1[i]:=p1^.odv1[i-1];
   end;
   p1^.EST[0]:=EST;p1^.odv1[0]:=odv;
  end;
{-----------------------------------}
  IF p2^.ahead <> nil then begin
   if factor =11 then begin
    case factor of
     6: begin PDC := PDC *0.7; if PDC<NULL then PDC:=NULL; end;
   3,5: begin if factor4=2 then PDC := PDC *0.5;end;
    end;
   end;
  end;
{-----------------------------------}
  {p1^.PDC:=PDC;}
  for i:=2 downto 1 do p1^.PDC[i]:=p1^.PDC[i-1];
  p1^.PDC[0]:=PDC;
  p1^.factor2 := factor2;p1^.factor4 := factor4;
{-------------- Following vehicle -----------}
 {-----------------------------------}
  if (DVper='c')and(odx>SDX) then begin
    factorth:=0;
    if dv <0.0 then TTC:= -dx/dv else TTC:=100;
    if ov1<=1.0 then begin
     if (TTC <= TTA)or(factor<>0)or((dx/ov2)<5) then begin
       factor:=1; dc2 := 0.5*sqr(Eodv) / (ABX-Eodx);
       if dc2 > NULL1 then dc2:= ABS(NULLP);
     end
     else begin
       if odx >100 then dc2:=NULL else dc2:=-1*ABS(NULLP);
     end;
    end
    else begin
     if (factor2=22)and(Eodv<0.0)and (ov2>0.0) then begin
       DTC:=(eodx-AX)/ov2;if DTC<0.0 then DTC :=0.0;
       DCmax := pdc*SQR(ov2)/(SQR(Eodv+ov2)-(2*pdc*ov2*DTC));
       if DTC=0.0 then DCmax:=DCmax-0.145;
       if DCmax>(0.5*sqr(Eodv)/((ABX)-odx)) then DCmax
:=0.5*(DCmax+0.5*sqr(Eodv)/(ABX-odx));
       if DCmax > NULL1 then DCmax :=null1;
```

```
      end
      else DCmax :=null1;
      case factor of
      3,6:begin
           factor := 6;
           if (TTC < TTA)or(F2<>0) then begin
            F2:=1; dc2:=(0.45*sqr(Eodv)/(ABX-odx));
            if dc2 > NULL1 then dc2:=ABS(NULLP);
           end
           else begin
             if ov1<5 then dc2:=ABS(NULLP) else dc2:=NULL;
           end;
           case factor2 of
           22: begin
                dc2 := Dcmax;
               end;
           11: Begin
                 dc2 := PDC +dc2;
               end;
           end;
          end;
{-----------------------}
        0:begin
           if (TTC>TTA)and (F2=0) then begin
            if (Eodx<=(50))or(Eodx<=(2*ABX)) then begin
              if factor2=22 then dc2:=NULL1 else dc2:=0.145;
            end
            else begin dc2:=0.5*Acmax2;end;
           end
           else begin
            factor:=1;
            if F2 =0 then F2:=1;
            F1:=0.45*EXP(((TTC-TTA)/(TTamin-TTA))-1);
            if (EST <>0.0)and (Eodv<0.0)then begin
             dc2:=F1*Eodv/(EST-0.5);
             if dc2> NULL1 then dc2:=ABS(NULLP);
            end
            else begin
             dc2:=NULL;
            end;
            if TTC <=TTAmin then F3:=1;
            if ((Eodx<(2*ABX))or(f3<>0))and(Eodv<0.0) then begin
              dc2:=2*odv/(2*TTC+0.25);
             if dc2>(0.5*sqr(Eodv)/(ABX-odx)) then
dc2:=(0.5*sqr(Eodv)/(ABX-odx));
            end;
            case factor2 of
              2: begin dc2:= dc2 +NULL1; end;
             22: begin
                   if factor4=2 then dc2:=dcmax;{dc2:=0.6*PDC+dc2;}
                   if dc2 < Dcmax then dc2 := Dcmax;
                 end;
            end;
          end;
         end;
{-----------------------}
       1:begin
```

```
          if (TTC>TTA)and (F2=0) then begin
           F3:=0;
           if (Eodx<(50))or(Eodx<=(2*ABX))  then begin
             if factor2=22 then dc2:=NULL1 else dc2:=0.145;
           end
           else begin dc2:=0.5*Acmax2;end;
          end
          else begin
           if F2 =0 then F2:=1;
           F1:=0.45*EXP(((TTC-TTA)/(TTamin-TTA))-1);
           if (EST <>0.0)and (Eodv<0.0)then begin
            dc2:=F1*Eodv/(EST-0.5);
            if dc2> NULL1 then dc2:=AbS(NULLP);
           end
           else begin
            dc2:=NULL;
           end;
           if TTC <=TTAmin then F3:=1;
           if ((Eodx<(2*ABX))or(f3<>0))and(Eodv<0.0) then begin
             dc2:=2*odv/(2*TTC+0.25);
            if dc2>(0.5*sqr(Eodv)/(ABX-odx)) then
dc2:=(0.5*sqr(Eodv)/(ABX-odx));
           end;
           if dc2> NULL1 then begin F2:=0;F3:=0;end;
           case factor2 of
             22: begin
                    if factor4=2 then dc2:=dcmax;{dc2:=0.6*PDC+dc2;}
                    if dc2 < Dcmax then dc2 := Dcmax;
                 end;
              2: begin dc2:= dc2 +NULL1; end;
           end;
           {if dc2>Dceng then dc2:=dceng;}
          end;
         end;
{-----------------------}
     end;
    end;
  end;
{-----------------------------------}
  if (DVper='s')and (Eodx > SDX)  then begin
   factorth:=0; F2:=0;F3:=0;
   if ov1<=1.0 then begin
    if ((odx-AX) > 50) then begin
     factor :=0;dc2:= Acmax2;
    end
    else dc2:=NULL;
   end
   else begin
    if ((odx-AX) > 75) then begin
     if factor <> 0 then p1^.NO_ahead:=0;
     factor :=0;dc2:= Acmax2;
     if factor4 =2 then dc2:= 0.5*Acmax2;
    end
    else begin
     if factor=1 then factor:=0;
     if (factor<>0)and (factor<>6) then begin
      p1^.dvr.rand[1]:=random;p1^.dvr.rand[2]:=random;
```

```
       p1^.ABXTimer:=0;
       factor :=6;
      end;
      if factor = 0 then begin
       if factor4=2 then dc2 := 0.5*acmax2 else dc2 := acmax2;
      end;
      if factor = 6 then begin
       if factor4=2 then dc2 := 0.25*acmax2 else dc2 := 0.5*acmax2;
      end;
     end;
    end;
   end;

 {--------------------------------}
   if (DVper='o')and(Eodx>SDX)  then begin
    factorth:=0;F2:=0;F3:=0;
    if ((Eodx-AX) >(1.2*SDX))  then begin
      if factor <> 0 then p1^.NO_ahead:=0;
      factor :=0;
      if ov1 < 5 then dc2:=NULL else dc2:= Acmax2;
    end
    else begin
     case factor of
      5,3,6:begin
             if factor<>6 then begin
              factor :=6;p1^.ABXTimer:=0;
              p1^.dvr.rand[1]:=random;p1^.dvr.rand[2]:=random;
             end ;
             dc2:= 0.5*ACmax2;dc2:=dc2+PDC;
             if dc2 > Acmax2 then begin dc2:=Acmax2; factor :=0;end;
             if (odx > (ABX*2)) and (odx>50) then begin
factor:=0;dc2:=ACmax2;end;
            end;
        1,0:begin
             dc2:=ACmax2;
            end;
     end;
    end;
   end;
  {--------------------------------}
  if(Eodx<=SDX)and(Eodx>ABX)and(DVper='c') then begin
   if ov1<=1.0 then begin
     dc2 := 0.65*sqr(Eodv)/(AX-Eodx);
     if dc2 > NULL1 then dc2:= ABS(NULLP);
     {if DC2>DCeng then DC2:=DCeng;}
   end
   else begin
    if (factor2=22)and(ov2 >0.0)then begin
      DTC:=(odx+Eodv*0.5+PDC*0.125-AX)/ov2;if DTC <0 then DTC:=0;
      if (Eodv+ov2)<=0 then
       Dcmax:= 0.5*sqr(Eodv)/(AX-Eodx)
      else begin
       DCmax := PDC*SQR(ov2)/(SQR(Eodv+ov2)-(2*PDC*ov2*DTC));
      end;
      DCmax:=DCmax-0.145;
      if DCmax>(0.5*sqr(Eodv)/(AX-Eodx)) then DCmax:=0.5*sqr(Eodv)/(AX-
Eodx);
```

```
      if DCmax>NULL1 then DCmax:=NULL1;
     end
     else {Dcmax :=DCeng;}Dcmax:=NULL1;
     case factor of
      1,0: begin
            factorth:=1;
            if (odx<>ABX)then begin
             dc2:=(-0.5*sqr(Eodv)/(odx-ABX));if dc2<(0.5*Eodv)then
dc2:=0.5*Eodv;
            end
            else begin dc2:=0.5*Eodv;end;
            if factor2=22 then begin
             if factor4=2 then dc2:=PDC+dc2 else dc2:=0.8*PDC+dc2;
             if(dc2<Dcmax)then dc2 := Dcmax;
            end;
           end;
    6,5,3: begin
            if dv <= CLDV then begin
             Space:=Eodx-AX;speed:=dv;
             p1^.OPDV:=OPDVthresh(Space);
             p1^.dvr.NULL1:=DCnull(space,speed);NULL1:=p1^.dvr.NULL1;
             factorth:=1;
             dc2:=(-0.5*sqr(Eodv)/(odx-((ABX+AX)/2)));if dc2 > NULL1 then
dc2:=NULL1;
             if factor2=22 then begin
               if factor4=2 then dc2:=PDC+dc2 else dc2:=0.8*PDC+dc2;
               if(dc2<Dcmax)then dc2 := Dcmax;
             end;
            end
            else begin
             case factorth of
              1:begin
                 dc2:=(-0.5*sqr(Eodv)/(odx-((ABX+AX)/2)));if dc2 > NULL1
then dc2:=NULL1;
                 if factor2=22 then begin
                   if factor4=2 then dc2:=PDC+dc2 else dc2:=0.8*PDC+dc2;
                   if(dc2<Dcmax)then dc2 := Dcmax;
                 end;
                end;
            2,0:begin
                 factorth:=0;dc2:=ABS(NULLP);
                 if factor2=22 then begin
                  Space:=Eodx-AX;speed:=dv;
                  p1^.dvr.NULL1:=DCnull(space,speed);NULL1:=p1^.dvr.NULL1;
                  dc2:=NULL1;factorth:=1;
                 end;
                end;
             end;
            end;
           end;
    end;
   end;
  end;
 {----------------------------}
  if(Eodx>=ABX)and(Eodx<=SDX)and(DVper='o')then begin
   case factor of
1,5,6,3:begin
```

```
          factor := 3;
          if (dv >= OPDV) then begin
           space := Eodx-AX; P1^.CLDV:=CLDVthresh(Space);
           p1^.dvr.NULL:=ACnull;NULL:=p1^.dvr.NULL;
           factorth:=2;
           if Eodx < DFD then dc2 :=NULL else dc2:=0.37*Eodv;
           if dc2 < NULL then dc2:=NULL;
           if factor2 =11 then begin
            dc2 := PDC+dc2; if dc2 < odc2 then dc2:=odc2;
           end;
          end
          else begin
           case factorth of
          2:begin
             if Eodx < DFD then dc2 :=NULL else dc2:=0.37*Eodv;
             if dc2 < NULL then dc2:=NULL;
              dc2:=NULL;
              if factor2=11 then begin
               dc2 := PDC+dc2; if dc2 < odc2 then dc2:=odc2;
              end;
            end;
        1,0:begin
             factorth:=0;dc2:=-1*ABS(NULLP);
             if factor2=11 then begin dc2 :=NULL;factorth:=2;end;
            end;
           end;
          end;
         end;
       0:begin dc2:=0.5*Acmax2;end;
    end;
   end;
          {----------------}
   if(Eodx<=SDX)and(Eodx>=ABX)and(DVper='s')then begin
    factor:=3;
    if ov1<=1.0 then begin
     dc2:=ABS(NULLP);
    end
    else begin
     case factorth of
       1:begin
          if (factor2=22) and (PDC < -0.5) then begin
           DTC:=(odx-AX)/ov2;if DTC<0.0 then DTC :=0.0;
           DCmax := PDC*SQR(ov2)/(SQR(odv+ov2)-(2*PDC*ov2*DTC));
           if DTC=0.0 then DCmax:=DCmax-0.145;
           dc2 :=DCmax-0.145;
          end
          else dc2:=-1*ABS(NULLP);
         end;
       2:begin
          if (factor2=11) and (PDC>0.5) then dc2:=odc12+0.145 else
dc2:=ABS(NULLP);
         end;
       0:begin
          if (factor2=22) and (PDC < -0.5) then begin
           DTC:=(odx-AX)/ov2;if DTC<0.0 then DTC :=0.0;
           DCmax := PDC*SQR(ov2)/(SQR(odv+ov2)-(2*PDC*ov2*DTC));
           if DTC=0.0 then DCmax:=DCmax-0.145;
```

```
          dc2 :=DCmax-0.145;
         end
         else begin
          if (factor2=11) and (odc12>0.5) then
           dc2:=PDC+0.145
          else begin
           if odc2< 0.0 then dc2:=-1*ABS(NULLP) else dc2:= ABS(NULLP);
          end;
         end;
        end;
    end;
   end;
  end;
{-------------------------------}
  if (Eodx <= ABX)and(DVper='c')then begin
   factorth:=0;
   if ov1<=1.0 then begin
    if (dx+dv*0.25)> AX then dc2 :=0.75*sqr(dv)/(Ax-dx) else dc2:=-
v2/0.25;
    if dc2 < (-v2/0.25) then dc2:=-v2/0.25;
   end
   else begin
    if  Eodx <=AX then begin
       dc2:=-Eodv/0.25+NULL1;
    end
    else begin
     if (dx-AX) < (((ABX-AX)/2)+1) then
       dc2 :=sqr(dv)/(Ax-Eodx)+NULL1
     else dc2:=-0.65*sqr(dv)/(Eodx-((ABX+AX)/2));
     {if DC2>DCeng then DC2:=Dceng;}
     if dc2>(NULLP-0.45) then dc2:=NULLP-0.45;
     if(factor2=22)then begin
      DTC:=(dx+dv+PDC*0.125-AX)/ov2; if DTC <0 then DTC:=0;
      if (SQR(dv+ov2)-(2*odc12*ov2*DTC))<>0 then begin
       DCmax := odc12*SQR(ov2)/(SQR(dv+ov2)-(2*odc12*ov2*DTC));
       dcmax:=DCmax-0.145;
       if (dx-AX) < ((ABX-AX)/2) then begin
         dcmax := dcmax-0.145;
         if DCmax > 0.75*sqr(dv)/(AX-Eodx) then DCmax := 0.75*sqr(dv)/(AX-
Eodx);
       end
       else begin
         if DCmax > 0.65*sqr(dv)/(AX-Eodx) then DCmax := 0.65*sqr(dv)/(AX-
Eodx);
       end;
       {if DCmax>DCeng then DCmax:=Dceng;}
       if dcmax>(NULLP-0.45) then dcmax:=NULLP-0.45;
      end
      else begin
        DCmax := dc2+PDC;
      end;
      if (dc2+PDC)<DCmax then dc2:=Dcmax else dc2:=PDC+dc2;
     end;
    end;
    factor :=5;
   end;
  end;
```

```
{-------------------------------}
  if (Eodx < ABX)and(DVper='o')  then begin
   factorth:=0;
   if ov1<=1.0 then begin
    if dx >Ax then dc2:=-0.145 else dc2:=-v2/0.25;
   end
   else begin
    factor :=5;
     if (odv > 1)and(odv >OPDV) then begin
       if factor2 =22 then
         dc2:=odc12+NULLP
       else begin
        dc2:=0.5*odc12+NULLP; if (factor=11)then dc2:=odc12+NULLP;
       end;
     end else begin
       if factor2 =22 then
         dc2:=odc12-0.145
       else begin
        dc2:=-0.145; if (factor=11) then dc2:=odc12-0.145;
       end;
     end;

     if (dx-AX)<((ABX-AX)/2) then dc2:=dc2-0.145;
   end;
  end;
{-------------------------------}
  if (Eodx < ABX)and(DVper='s')  then begin
   factorth:=0;
   if ov1<=1.0 then begin
    if dx >Ax then dc2:=NULL1 else dc2:=-v2/0.25;
   end
   else begin
    factor :=5;dc2:=-0.45+NULLP+p2^.dc[0];
    if dc2 > NULL1 then dc2:=NULL1;
   end;
   {if DC2>DCeng then DC2:=Dceng;}
  end;
{-------------------------------}
  p1^.ABXTimer:=p1^.ABXtimer+1;
  if dc2 > Acmax2 then dc2 := Acmax2;
  if (dx > AX) then begin
   if dc2 < (0.5*sqr(v2)/(ax-dx)) then dc2 :=0.5*sqr(v2)/(ax-dx);
  end;
  if dc2< -9 then dc2:=-9;
  VL := v2 + 0.25*dc2+ 0.25*odc2;
  if VL < 0.0 then begin
   dc2:=-4*(v2+0.25*odc2);
  end;
 end
 else begin
  factor:=0;
  dc2 := Acmax2;
  VL := v2 + 0.25 * dc2+ 0.25 * odc2;
 end;
  {------- UPDATING -----------}
   if ((p1^.ahead<>nil)and(factor = 0))or(p1^.ahead=nil)then begin
     if (VL > vd)and(VL < vd+1.15)and(odc2>0) then dc2 :=ABS(NULLP);
```

```
      if (VL > vd)and(VL < vd+1.15)and(odc2<0) then dc2 :=-1*ABS(NULLP);
      if (VL < vd)and(VL > vd-1.15)and(odc2<0) then dc2 :=-1*ABS(NULLP);
      if (VL < vd)and(VL > vd-1.15)and(odc2>0) then dc2 :=ABS(NULLP);
      if (VL > (vd+1.15))and (odc2>NULL1)then dc2 :=NULL1;
      if (VL < (vd-1.15))and (odc2<NULL) then dc2 :=NULL;
      if (VL > (vd+2))and(dc2>(-0.2+NULLP)) then dc2:=-0.2+NULLP;
    end
    else begin
      if (VL>(vd+2))and(dc2>(-0.2+NULLP)) then dc2:=-0.2+NULLP;
      if (VL > vd)and(VL < vd+1.15) then begin
       if dc2>ABS(NULLP) then dc2:=NULLP;
      end;
      if (VL < vd)and(VL > vd-1.15) then begin
       if dc2>ABS(NULLP) then dc2:=NULLP;
      end;
    end;

 20:
   VL := v2 + 0.25 * dc2+ 0.25 * odc2;
   if VL < 0.0 then begin
    p1^.factor3 :=1;
    dc2:=-4*(v2+0.25*odc2);
   end;
   if p2 <> nil then begin
    x2 := x2+v2*0.25+0.03125*odc2;
    x1 := p2^.x[2]+p2^.v[2]*0.25+0.03125*p2^.dc[2] -LenV;
    if x2 >(x1) then begin
     p1^.v[2]:=p2^.v[2]-0.25;
     dc2 := p2^.dc[0]+(NULLP-0.45);
     if p1^.v[2] < 0.0 then p1^.v[2]:=0.0;dc2:=0.0;
    end;

   end;
   p1^.dc[0] := dc2;
   p1^.F2 := F2;  p1^.F3 := F3;
   p1^.factor := factor; p1^.factorth := factorth;
  {-------------------}
end; { procedure
-----------------------------------------------------------------}
procedure follow2;
var
 i :byte;
begin
    if p1^.ACC.EQ='Y' then p1^.ACC.Timer:=p1^.ACC.Timer+0.125;
    for i:=1 to 2 do begin
     p1^.EST[i]:=p1^.EST[i-1];
     p1^.odv1[i]:=p1^.odv1[i-1];
     p1^.PDC[i]:=p1^.PDC[i-1];
     p1^.Dvper[i]:=p1^.Dvper[i-1];
     p1^.theta[i]:=p1^.theta[i-1];
    end;
end; { procedure
-----------------------------------------------------------------}
procedure follow_ACC;
var
 VL,XL,VF,XF,headway,DCeng,DCacc    :single;
 DX1,DX2,DC_CAS,DV                  :single;
```

```
begin
 if p1^.ACC.EQ ='N' then Exit;
 if p1^.ACC.Opp ='of' then begin
  if p1^.ACC.timer < p1^.ACC.Tmin_R then Exit;
  if p1^.V[1] < p1^.ACC.Vmin_D then Exit;
  if (p1^.ahead =nil) then exit;
  VL:=p1^.ahead^.V[1]; XL:=p1^.ahead^.X[1]; VF:=p1^.V[1]; XF:=p1^.X[1];
  headway :=(XL-XF)/VF;
  DCeng := -0.0317*VF+0.132;if Dceng >-0.4 then DCeng :=-0.4;
  if headway > p1^.ACC.Thmax_D then Exit;
  if p1^.ahead^.oLane<>p1^.ahead^.Lane then Exit;
  if p1^.dc[0] < Dceng then exit;
  if p1^.dc[0] > p1^.ACC.ACmax then exit;
  if VL > p1^.ACC.Vmax_D then Exit;
  if (XF < 100) or (XF >9900) then Exit;
  p1^.ACC.Opp:='on';p1^.dc[0]:=0.0;p1^.ACC.Timer:=0.0;
  Exit;
 end;
 if p1^.ahead =nil then exit;
 if p1^.ahead^.oLane<>p1^.ahead^.Lane then
 begin
  p1^.ACC.Opp:='of';p1^.ACC.Timer:=0.0; Exit;
 end;
 VL:=p1^.ahead^.V[1]; XL:=p1^.ahead^.X[1]; VF:=p1^.V[1]; XF:=p1^.X[1];
 headway :=(XL-XF)/VF;
 {-------------}
 if headway > p1^.ACC.Thmax_C then
 begin
  p1^.ACC.Opp:='of';p1^.ACC.Timer:=0.0; Exit;
 end;
 {-------------}
 if p1^.dc[0] < p1^.ACC.DCmax then
 begin
  p1^.ACC.Opp:='of';p1^.ACC.Timer:=0.0; Exit;
 end;
 {-------------}
 if ACC_formula =1 then begin
  DX1:=p1^.ahead.X[1]-p1^.X[1] -(p1^.ahead^.veh.LenV/100);
  DX2:=p1^.V[1]*p1^.ACC.Th_D-(p1^.ahead^.veh.LenV/100);
  DX1:=DX1-DX2;
  DX2:=p1^.ahead.V[1]-p1^.V[1];
  DCacc:=0.2*DX1+3*DX2;
 end else begin
  DX1:=p1^.ahead.X[0]-p1^.X[0] -(p1^.ahead^.veh.LenV/100);
  DX2:=p1^.ahead.X[1]-p1^.X[1] -(p1^.ahead^.veh.LenV/100);
  VF:=p1^.V[0];
  DCacc := (((DX1-DX2)*500/0.125)+(31.25*DX1)-31.25*VF*p1^.ACC.Th_D)/2000;
 end;
 {-------------}
 VF := p1^.V[0]+DCacc*0.125;
 DV:=p1^.ahead.V[1]-p1^.V[1];
 if DV<0.0 then begin
  DC_CAS:=-0.5*SQR(DV)/(DX2-2);
  if DCacc>DC_CAS then DCacc:=DC_CAS;
 end;
 if VF > p1^.ACC.Vmax_D then
 begin
```

```
  p1^.ACC.Opp:='of';p1^.ACC.Timer:=0.0; Exit;
 end;
 if DCacc > p1^.ACC.ACmax then DCacc :=p1^.ACC.Acmax;
 if DCacc < p1^.ACC.DCmax then DCacc :=p1^.ACC.Dcmax;
 p1^.dc[0]:=DCacc;
end; { procedure
----------------------------------------------------------------}
procedure follow;
begin
 if int(T/0.25)=(T/0.25) then begin
  p1 := lastv[1]; while (p1<> nil) do begin follow1; Follow_ACC; p1 :=
p1^.back; end;
  p1 := lastv[2]; while (p1<> nil) do begin follow1; Follow_ACC; p1 :=
p1^.back; end;
  p1 := lastv[3]; while (p1<> nil) do begin follow1; Follow_ACC; p1 :=
p1^.back; end;
 end
 else begin
  p1 := lastv[1]; while (p1<> nil) do begin follow2; Follow_ACC; p1 :=
p1^.back; end;
  p1 := lastv[2]; while (p1<> nil) do begin follow2; Follow_ACC; p1 :=
p1^.back; end;
  p1 := lastv[3]; while (p1<> nil) do begin follow2; Follow_ACC; p1 :=
p1^.back; end;
 end;
end; { procedure
----------------------------------------------------------------}
begin
end.
{------*****************************************************-------}
unit Lane_ACC;
interface
uses variab_ACC,folow_ACC,Randno_ACC,gener_ACC;
 procedure detectlane;
 procedure lanchang;

implementation
procedure detect1;
var
 DV,DV1,ABX,dx,dx1,odx,odv  :single;
begin
 p1^.overtake1 :=0;p1^.overtake2:=0;
 if p1^.x[2] < 75 then exit;
 if (p1^.overtaketime >= p1^.LCT) then begin
  If(p1^.overtake=11)then begin
   if(p1^.ah_left <> p1^.takeveh)then begin
    p1^.overtake:=0;p1^.overtaketime:=0;
   end
   else begin
    if(p1^.ah_left<>nil)then begin
     if (p1^.ah_left.x[4] >(p1^.x[4]+200))or(p1^.ah_left^.v[4]>((1-
p1^.dvr.RAND[6])*p1^.dvr.Dspeed)) then begin
      p1^.overtake:=0;p1^.overtaketime:=0;
     end;
    end
    else begin
     p1^.overtake:=0;p1^.overtaketime:=0;
```

```
     end;
    end;
   end;
   If(p1^.overtake=22)then begin
    if(p1^.ba_right <> p1^.takeveh)then begin
     p1^.overtake:=0;p1^.overtaketime:=0;
    end
    else begin
     if(p1^.ba_right<>nil) then begin
      if (p1^.ba_right^.x[4]<(p1^.x[4]-
200))or(p1^.ba_right^.v[4]<p1^.v[4])then begin
       p1^.overtake:=0;p1^.overtaketime:=0;
      end;
     end
     else begin
       p1^.overtake:=0;p1^.overtaketime:=0;
     end;
    end;
   end;
  end;
  if(p1^.overtake=22)or(p1^.overtake=11) then exit;
  if (p1^.ahead <>nil)and((p1^.ahead^.x[2]-p1^.x[2]-
(p1^.ahead^.veh.LenV/100))<200)Then begin
   DV :=p1^.ahead^.x[2]-p1^.x[2];
   DV:=0.0019*sqr(DV)-0.0817*DV+2.5318;
   if DV < 1.8 then DV :=1.8;
   if ((p1^.ahead^.v[2]+DV)<=p1^.dvr.dspeed)or(p1^.DVper[0]='c') then begin
     if(p1^.ahead^.V[4]< ((1-p1^.dvr.Rand[6])*p1^.dvr.dspeed))THEN BEGIN
      if((p1^.ahead^.dvr.dspeed)< p1^.dvr.dspeed) then p1^.overtake1:=1;
     end else begin
      if(p1^.ahead^.dvr.dspeed<(p1^.dvr.dspeed-
2))and((p1^.factor=3)or(p1^.factor=5))and
        ((p1^.ba_right =nil)or((p1^.ba_right
<>nil)and(p1^.ba_right^.X[4]<(p1^.X[4]-200))))
      then begin
        if NOT((p1^.ACC.EQ ='Y')and(p1^.ACC.Opp
='on')and(p1^.ACC.timer<p1^.ACC.Tmin_R)) then p1^.overtake1:=1;
      end;
     end;
   end;
  end;
   if p1^.back<>nil then begin
    DV :=p1^.x[4]-p1^.back^.x[4]-(p1^.veh.lenV/100);
    if DV <200 then begin
     if (p1^.back^.v[4]>0)then begin
      ABX:=EXP(Ln(2.0687)+0.6944*Ln(p1^.back^.v[4]*3.6));
      if (ABX >DV)then begin
       if(p1^.back^.dvr.Dspeed > (p1^.dvr.Dspeed-1))then p1^.overtake2:=1;
      end;
     end;
    end;
   end;

{-----------------}
 if
(p1^.ba_left<>nil)and((p1^.ba_left^.X[4]+75)>p1^.X[4])and(p1^.ba_left.dvr.
Dspeed>(p1^.dvr.Dspeed+2))then begin
```

```
  if NOT((p1^.ACC.EQ ='Y')and(p1^.ACC.Opp
='on')and(p1^.ACC.timer<p1^.ACC.Tmin_R)) then p1^.overtake2:=1;
 end;
{------------------ }
 if (p1^.overtake1=1) and (p1^.overtake2=1) then begin
  if p1^.lane >=2 then begin
    if(p1^.ah_left=nil)then begin
     p1^.overtake2:=1;p1^.overtake1:=0;
    end else begin
     if (p1^.ah_left^.X[4]>(p1^.X[4]+700)) then begin
      p1^.overtake2:=1;p1^.overtake1:=0;
     end else begin
      p1^.overtake2:=0;p1^.overtake1:=1;
     end;
    end;
  end else begin
    p1^.overtake2:=0;p1^.overtake1:=1;
  end;
 end;
{------------------}
end; { procedure
-----------------------------------------------------------------}
procedure detect2;
var
 DV,DV1,ABX,dx,dx1,odx,odv  :single;
begin
 if(p1^.overtake=22)or(p1^.overtake=11) then exit;
{----------------Assessing the Lane hugging behaviour------}
if NOT((p1^.ACC.EQ ='Y')and(p1^.ACC.Opp
='on')and(p1^.ACC.timer<p1^.ACC.Tmin_R)) then begin
 if (p1^.overtake1=0)and(p1^.overtake2=0) then begin
  if (p1^.lane <
p1^.Flane)and(p1^.dvr.rand[5]<LChug)and(p1^.veh.kind<3)then begin
   if NOT((p1^.back<>nil)and(p1^.back^.overtake1=1))then begin
    if(p1^.ba_right=nil)then
     p1^.overtake1:=1
    else begin
     if (p1^.ba_right.X[4]<(p1^.X[4]-200)) then p1^.overtake1:=1;
    end;
   end;
  end;
  if (p1^.lane >
p1^.Flane)and(p1^.dvr.rand[5]<LChug)and(p1^.veh.kind<3)then begin
   if NOT((p1^.ahead<>nil)and(p1^.ahead^.overtake2=1))then begin
    if(p1^.ah_left=nil)then
     p1^.overtake2:=1
    else begin
     if (p1^.ah_left.X[4]>(p1^.X[4]+200)) then p1^.overtake2:=1;
    end;
   end;
  end;
 end;
end;
 {---------------- detecting if the follower can over take or not--------}
 if (p1^.overtake2=1)and(p1^.lane=2)and(p1^.back<>nil) then begin
   if (p1^.back^.overtake1=0)and (p1^.dvr.rand[5]<LChug)
      and (p1^.veh.kind < 3) and (p1^.Flane > 1) then  p1^.overtake2:=0;
```

```
    if (p1^.back^.overtake1=1) then begin
     if (p1^.back^.ba_right=nil) then
      p1^.overtake2:=0
     else begin
      if (p1^.back^.ba_right^.X[4]+200)<(p1^.back^.X[4]) then
p1^.overtake2:=0;
     end;
    end;
  end;
  {----------------Assessing the come back to slwoer Lane behaviour------}
 if NOT((p1^.ACC.EQ ='Y')and(p1^.ACC.Opp
='on')and(p1^.ACC.timer<p1^.ACC.Tmin_R)) then begin
  if (p1^.overtake1=0)and(p1^.overtake2=0) then begin
   if NOT((p1^.lane <= p1^.Flane)and
(p1^.dvr.rand[5]<LChug)and(p1^.veh.kind<3))then begin
      if(p1^.ah_left=nil)then
     p1^.overtake2:=1
    else begin
     if (p1^.ah_left^.X[4]>(p1^.X[4]+200)) then p1^.overtake2:=1;
    end;
   end;
  end;
 end;
{--------------}
 if p1^.lane =1 then p1^.overtake2:=0;

{--------------}
 if (p1^.overtake2=1)and(p1^.ah_left<>nil) then begin
   if (p1^.ah_left^.X[4]<(p1^.X[4]+75))and
(p1^.ah_left^.V[4]<(p1^.dvr.dspeed-2))then p1^.overtake2:=0;
 end;
{--------------}
 if (p1^.overtake1=1)and(p1^.ba_right<>nil)and(p1^.lane=2)then begin
  if (p1^.ba_right^.X[4]>(p1^.X[4]-75))
      and
(p1^.ba_right^.dvr.Dspeed>((1+p1^.dvr.RAND[6])*p1^.dvr.dspeed))then
p1^.overtake1:=0;
 end;
{--------------}
 if (p1^.overtake1=1)and(p1^.lane=2)and(p1^.veh.kind>2)then
p1^.overtake1:=0;
 {--------------}
 if p1^.overtake =1 then begin
  case p1^.lane of
  1:begin
      p2:=p1^.ah_right; p3:=nil;
      while (p2<>nil)and((p2^.x[4]-p1^.x[4])<200)  do begin
       p3:=p2;p2:=p2^.ahead;
      end;
      if p3<>nil then begin
       dx:=0;dv:=0;p2:=p3;
       while (p2<>nil)and(p2^.x[4]>p1^.x[4]) do begin
        if dv> p2^.v[4]then dv :=p1^.V[4];
        dx:=dx+1;p2:=p2^.back;
       end;
       if dx>0 then begin
          if dv<(p1^.v[4]*(1+p1^.dvr.RAND[6])) then p1^.overtake1:=0;
```

```
        end;
       end;
      end;
   2:begin
       p2:=p1^.ah_right; p3:=nil;
       while (p2<>nil)and((p2^.x[4]-p1^.x[4])<200)  do begin
        p3:=p2;p2:=p2^.ahead;
       end;
       if p3<>nil then begin
        dx:=0;dv:=0;p2:=p3;
        while (p2<>nil)and(p2^.x[4]>p1^.x[4]) do begin
         if dv> p2^.v[4]then dv :=p1^.V[4];
         dx:=dx+1;p2:=p2^.back;
        end;
        if dx>6 then
          p1^.overtake1:=0
        else begin
          if (dx>0)and(dv<(p1^.v[4]*(1+p1^.dvr.RAND[6]))) then
p1^.overtake1:=0;
        end;
       end;
      end;
   3:begin
      p2:=p1^.ahead; p3:=nil;dx:=0;p1^.overtake1:=0;dv:=0;
      while (p2<>nil)and((p2^.x[4]-p1^.x[4])<200)  do begin
       if dv> p2^.v[4]then dv :=p1^.V[4];
       p3:=p2;p2:=p3^.ahead;dx:=dx+1;
      end;
      if dx>6 then begin
       p2:=p1^.ah_left; p3:=nil;
       while (p2<>nil)and((p2^.x[4]-p1^.x[4])<300)  do begin
        p3:=p2;p2:=p2^.ahead;
       end;
       if p3<>nil then begin
        dx:=0;dv:=0;p2:=p3;
        while (p2<>nil)and(p2^.x[4]>p1^.x[4]) do begin
         if dv> p2^.v[4]then dv :=p1^.V[4];
         dx:=dx+1;p2:=p2^.back;
        end;
        if dx>0 then begin
         if dv>(p1^.v[4]*(1-p1^.dvr.RAND[6])) then p1^.overtake2:=1;
        end else begin
         p1^.overtake2:=1;
        end;
       end else begin
         p1^.overtake2:=1;
       end;
      end;
     end;
   end;
  end;
{-------------}
end; { procedure
-----------------------------------------------------------}
procedure detectlane;
begin
 p1 := lastv[1]; while (p1<> nil) do begin detect1; p1 := p1^.back;end;
```

```
 p1 := lastv[2]; while (p1<> nil) do begin detect1; p1 := p1^.back;end;
 p1 := lastv[3]; while (p1<> nil) do begin detect1; p1 := p1^.back;end;
 p1 := lastv[1]; while (p1<> nil) do begin detect2; p1 := p1^.back;end;
 p1 := lastv[2]; while (p1<> nil) do begin detect2; p1 := p1^.back;end;
 p1 := lastv[3]; while (p1<> nil) do begin detect2; p1 := p1^.back;end;
end; { procedure
-------------------------------------------------------------}
Function ahead_R :byte;
{ This function to assess the influence of the lead vehicle on lane
changing}
var
  v,v2,v3,DV,DV1,DX,X,X2,X3,ABX                        : single;
begin
 Ahead_R :=0;
 if (p3^.ahead <>nil)and(p3^.ahead^.overtake = 1) then begin Ahead_R
:=0;Exit; end;
 if p3^.ah_right = nil then begin Ahead_R :=1;Exit; end;

 if(p3^.ah_right^.overtake=11)and((p3^.ah_right^.x[2]-p3^.x[2])<50)
    and(p3^.ah_right^.overtaketime<2)then begin Ahead_R :=0;Exit; end;

 if(p3^.ah_right^.overtake=22)and((p3^.ah_right^.x[2]-p3^.x[2])<50)
   and(p3^.ah_right^.overtaketime<2)then begin Ahead_R :=0;Exit; end;

 if (p3^.ahead <>nil) then begin
  V3 :=p3^.ah_right^.V[2];X3 :=p3^.ah_right^.X[2];
  V2 :=p3^.ahead^.V[2];X2 :=p3^.ahead^.X[2];
  V  :=p3^.V[2];X :=p3^.X[2];
  if (X3 <= X2) then begin
   if V3 <= V2 then begin
    Ahead_R :=0;Exit;
   end
   else begin
    DV := V3-V; DX:=X3-X;
    if DV > 0 then begin
     ABX:=ABXthresh(V3,p3^.dvr.rand[1]);
     if (DX-(p3^.ah_right^.veh.LenV/100))>ABX then Ahead_R :=1 else
Ahead_R :=0;
     Exit;
    end
    else begin
     DV1 := p3^.dvr.rand[8]*(-0.001*sqr(DX)-0.00105*DX-2.6741);
     if DV1< DV then begin
      ABX:=ABXthresh(V3,p3^.dvr.rand[1]);
      if (DX-(p3^.ah_right^.veh.LenV/100))>ABX then Ahead_R :=1 else
Ahead_R :=0;
      Exit;
     end
     else begin Ahead_R :=0;Exit; end;
    end;
   end;
  end
  else begin
   if (X3-X2)>100 then begin
    Ahead_R :=1;Exit;
   end
   else begin
```

```
    if V3 >= V2 then Ahead_R :=1 else Ahead_R :=0;
    exit;
   end;
  end;
 end
 else begin
  V3 :=p3^.ah_right^.V[2];X3 :=p3^.ah_right^.X[2];
  V  :=p3^.V[2];X :=p3^.X[2];
  DV := V3-V; DX:=X3-X;
  if DV > 0 then begin
   ABX:=ABXthresh(V3,p3^.dvr.rand[1]);
   if (DX-(p3^.ah_right^.veh.LenV/100))>ABX then Ahead_R :=1 else Ahead_R
:=0;
   Exit;
  end
  else begin
   if ((x3-x)<100) then begin
    DV1 := p3^.dvr.rand[8]*( -0.001*sqr(DX)-0.00105*DX-2.6741);
    if DV1< DV then begin
     ABX:=ABXthresh(V3,p3^.dvr.rand[1]);
     if (DX-(p3^.ah_right^.veh.LenV/100))>ABX then Ahead_R :=1 else
Ahead_R :=0;
     Exit;
    end
    else begin Ahead_R :=0;Exit; end;
   end
   else begin Ahead_R :=1;Exit; end;
  end;
 end;
end;{ Function
-----------------------------------------------------------------}
Function Back_R(var stimu:single) :byte;
var
 V,V1,X,X1,DX,DV,DV1,ABX,SLDV  :single;
 BR                            :Byte;
begin
 Back_R:=0;
 if (p3^.ba_right = nil) then begin Back_R:=1;exit;end;

 if(p3^.ba_right^.overtake=22)and((p3^.x[2]-p3^.ba_right^.x[2])<50)
    and(p3^.ba_right^.overtaketime<2)then begin Back_R :=0;Exit; end;

 V := p3^.v[2]; X := p3^.x[2];
 V1 := p3^.ba_right^.v[2]; X1 := p3^.ba_right^.x[2];
 DX := X-X1 ;
 if ((DX-(p3^.veh.lenV/100))>200)then begin Back_R:=1;exit;end;
 ABX:=ABXthresh(v,p3^.ba_right^.dvr.rand[1]);
 if NOT(((DX-(p3^.veh.lenV/100))>ABX)and((DX-(p3^.veh.lenV/100))>12))then
begin Back_R:=0;exit;end;
 if V1 < V then begin
   Back_R:=1;exit;
 end
 else begin
   DV := p3^.dvr.rand[8] *(2.407*LN(DX)-3.8711);
   DV1 :=p3^.dvr.rand[8] *(4.0805*LN(DX)-8.7619);
   SLDV:=DV+(DV1-DV)*stimu;
   if (V1-V)<=SLDV then begin
```

```
    Back_R :=1;exit;
   end
   else  begin
    if p3^.lane =1 then begin
     BR:=0;
     if (p3^.ba_right^.ba_right =nil) then begin
      if (p3^.ba_right^.ah_right =nil) then
       BR :=1
      else begin
       if (p3^.ba_right^.ah_right^.v[4] > p3^.ba_right^.v[4])and
       (p3^.ba_right^.ah_right^.x[4] > p3^.x[4]) then BR :=1 else BR :=0;
      end;
     end
     else begin
      if (p3^.ba_right^.ba_right^.x[4] < (p3^.ba_right^.x[4]-100)) and
       (p3^.ba_right^.ba_right^.x[4] < (p3^.x[4]-200)) then begin
        if (p3^.ba_right^.ah_right =nil) then
         BR :=1
        else begin
         if (p3^.ba_right^.ah_right^.v[4] > p3^.ba_right^.v[4])and
         (p3^.ba_right^.ah_right^.x[4] > p3^.x[4]) then BR :=1 else BR
:=0;
        end;
      end
      else BR :=0;
     end;
     DV1 := p3^.dvr.rand[8] *(5.6037*LN(DX)-12.051);
     SLDV:=DV+(DV1-DV)*stimu;
     if ((V1-V)>SLDV)and (BR=1) then BR:=0;
     if(BR
<>0)and((p3^.ba_right^.overtake=22)or(p3^.ba_right^.overtake=11))
     then p3^.ba_right^.overtake:=0;
    end
    else BR :=0;
    Back_R:=BR;
  end;
 end;
end;{ Function
----------------------------------------------------------------}
Function Ahead_L :byte;
var
 vd,v,x,x2,v3,x3,ABX,Tc,DX,TTCthreshold,DV   :single;
begin
 Ahead_L:=0;
 if (p3^.ahead<>nil)and(p3^.ahead^.overtake=2) then begin Ahead_L:=0;Exit;
end;
 if (p3^.ah_left =nil) then begin Ahead_L:=1;Exit; end;

 if(p3^.ah_Left^.overtake=22)and((p3^.ah_Left^.x[2]-p3^.x[2])<50)
    and(p3^.ah_Left^.overtaketime<2)then begin Ahead_L :=0;Exit; end;

 if(p3^.ah_Left^.overtake=11)and((p3^.ah_Left^.x[2]-p3^.x[2])<50)
    and(p3^.ah_Left^.overtaketime<2)then begin Ahead_L :=0;Exit; end;

 vd := p3^.dvr.Cdspeed; v := p3^.v[4];x := p3^.x[4];
 v3 := p3^.ah_left^.v[4];x3 := p3^.ah_left^.x[4];
 ABX:=(p3^.ah_left^.veh.lenV/100)+ABXthresh(v3,p3^.dvr.rand[1]);
```

```
 x2:=x3-(p3^.ah_left^.veh.lenV/100)-x;

 if x2 < ABX then begin Ahead_L:=0;Exit; end;
 if v3 >= V then begin Ahead_L:=1;Exit; end;

 TC := (x3-x-(p1^.ah_left^.veh.lenV/100))/(v-v3);
 if (p3^.back <>nil)and(p3^.back^.v[4]>0.0) then begin
   DX:=(p3^.x[4]-p3^.back^.x[4]);
  TTCthreshold := TTCLCL-(TTCLCL-p3^.TTA)*StimuLC_left(p3^.back^.v[4],DX);
  if TC < TTCthreshold then begin Ahead_L:=0; exit;end;
  x2:=x3-x;
  DV:=-0.0019*sqr(x2)+0.0817*x2-2.5318;
  if (v3-v)<DV then begin Ahead_L:=0; exit;end;
 end
 else begin
  if TC < TTCLCL then begin Ahead_L:=0; exit;end;
 end;
 Ahead_L:=1;
end;{ Function
-------------------------------------------------------------------}
Function Back_L :byte;
var
 vd,v,x,v2,x2,ABX,dx,dv   :single;
begin
 Back_L :=0;
 if(p3^.ba_Left=nil)then begin Back_L :=1;Exit; end;

 if(p3^.ba_Left^.overtake=11)and((p3^.x[2]-p3^.ba_Left^.x[2])<50)
   and(p3^.ba_Left^.overtaketime<2)then begin Back_L :=0;Exit; end;

 v := p3^.v[4];x := p3^.x[4];
 ABX:=ABXthresh(v,p3^.dvr.rand[1]);
 v2 := p3^.ba_left^.v[4];x2 := p3^.ba_left^.x[4];
 dx:=x-x2-(p3^.veh.lenV/100);
 if (dx< ABX)  then begin Back_L :=0; exit; end;
 if (dx>200)or((V-v2)>=0)   then begin Back_L :=1; exit; end;
 x2:=p3^.x[2]-p3^.ba_left^.x[2]; vd:= p3^.v[2]-p3^.ba_left^.v[2];
 if (x2< 15)  then begin Back_L :=0; exit; end;
 DV:= -1.8531*ln(x2)+3.4245;
 if Vd <DV then begin Back_L :=0; exit; end;
 Back_L:=1;
end;{ Function
-------------------------------------------------------------------}
Procedure Lanchang2;
var
 A,b,c,d,E :Byte;
 stimu1,ABX,TC,DV  :single;
begin
 if p1^.overtake1 = 1 then begin
   p3 := p1;A:=0;B:=0;p1^.overtake :=0;
   A:=ahead_R;
   if A <>1 then exit;
   B:=Back_R(p3^.stimu);
   if B=1 then begin
    p1^.overtake :=1;
    p1^.factor:=1; p1^.dc[0]:=0.0;{p2:=p1^.ah_right;follow1;
    if (p1^.ACC.Eq ='Y')and(p1^.ACC.OPP ='on')then begin
```

```
      p1^.ACC.Opp :='of';p1^.ACC.Timer :=0.0;
     end;
    end;
    Exit;
  end;
  if p1^.overtake2 = 1 then begin
   p3:=p1;A:=0;B:=0;C:=0;d:=0;p1^.overtake :=0;
   A:=ahead_L;
   B:=Back_L;
   if (A=1)and(b=1) then begin
    p3:=p1^.back;C:=1;
    if (p3<>nil)and(p1^.lane<>3)then begin
      DV :=p1^.x[4]-p3^.x[4]-(p1^.veh.lenV/100);
      if(DV<200)and(p1^.lane <= p1^.Flane)and(p1^.dvr.rand[5]<LChug)then
begin
       stimul :=0.0;
       d:=Back_R(stimul); E:=ahead_R;
       if (d =1)and(E=1)then C:=0;
      end;
    end;
   end;
   p1^.overtake:=0;
   if C=1 then begin
    p1^.overtake:=2;
    if p1^.lane =3 then begin
     p3:=p1^.ba_left;p2:=p1^.ah_left;
     if p3=nil then p3:=start[1] else p3:=p3^.ah_left;
     if p2=nil then p2:=lastV[1] else p2:=p2^.ba_left;
     while (p3<>nil)and(p3^.x[2]<(p1^.x[2]-50)) do begin
      p3:=p3^.ahead;
     end;
     while (p2<>nil)and(p2^.x[2]>(p1^.x[2]+50)) do begin
      p2:=p2^.back;
     end;
     if (p3<>nil)and(p3^.x[2]>(p1^.x[2]+50)) then p3:=nil;
     if (p2<>nil)and(p2^.x[2]<(p1^.x[2]-50)) then p2:=nil;
     if (p3=nil)and(p2=nil) then begin
      p1^.factor:=1; p1^.dc[0]:=0.0;{ p2:=p1^.ah_left;follow1;}
      if (p1^.ACC.Eq ='Y')and(p1^.ACC.OPP ='on')then begin
       p1^.ACC.Opp :='of';p1^.ACC.Timer :=0.0;
      end;
      exit;
     end;
     if (p3<>nil)and(p2<>nil)and(p3^.x[2] >p2^.x[2])then begin
      p1^.factor:=1; p1^.dc[0]:=0.0; {p2:=p1^.ah_left;follow1;}
      if (p1^.ACC.Eq ='Y')and(p1^.ACC.OPP ='on')then begin
       p1^.ACC.Opp :='of';p1^.ACC.Timer :=0.0;
      end;
      exit;
     end;
     while (p2<>nil)and(p2^.X[2] >=(p1^.x[2]-50)) do begin
       if(p2^.overtake=1)then begin p1^.overtake:=0;exit;end;
       p2:=p2^.back;
     end;
     while (p3<>nil)and(p3^.X[2] <=(p1^.x[2]+50)) do begin
       if(p3^.overtake=1)then begin p1^.overtake:=0;exit;end;
       p3:=p3^.ahead;
```

```
    end;
   end;
   p1^.factor:=1; p1^.dc[0]:=0.0;{ p2:=p1^.ah_left;follow1;}
   if (p1^.ACC.Eq ='Y')and(p1^.ACC.OPP ='on')then begin
     p1^.ACC.Opp :='of';p1^.ACC.Timer :=0.0;
   end;
  end;
 end;
end; { procedure
-----------------------------------------------------------------}
procedure lanchang;
begin
p1 := lastv[1];
while (p1<> nil) do begin
 lanchang2;
 if(p1^.overtake=1)and(round(int(T))>(2*RCtime)) then begin
  if(p1^.x[2]>=Dstar)and(p1^.x[2]<=DEnd)then LCnumber[1]:=LCnumber[1]+1;
 end;
 p1 := p1^.back;
end;
p1 := lastv[2];
while (p1<> nil) do begin
 lanchang2;
 if(p1^.overtake=1)and(round(int(T))>(2*RCtime))then begin
   if(p1^.x[2]>=Dstar)and(p1^.x[2]<=DEnd)then LCnumber[2]:=LCnumber[2]+1;
 end;
 if(p1^.overtake=2)and(round(int(T))>(2*RCtime))then begin
   if(p1^.x[2]>=Dstar)and(p1^.x[2]<=DEnd)then LCnumber[3]:=LCnumber[3]+1;
 end;
 p1 := p1^.back;
end;
p1 := lastv[3];
while (p1<> nil) do begin
lanchang2;
if(p1^.overtake=2)and(round(int(T))>(2*RCtime))then begin
 if(p1^.x[2]>=Dstar)and(p1^.x[2]<=DEnd)then LCnumber[4]:=LCnumber[4]+1;
end;
p1 := p1^.back;
end;
end; { procedure
-----------------------------------------------------------------}
begin
end.
{-------*****************************************************-------}
unit gener_ACC;
interface
uses variab_ACC,RandNO_ACC;

 procedure generator;
 procedure final;
 procedure Updatep;
 function DVperception(var space,Speed,theta,theta1:single;
DVper1:char):Char;
 function CLDVThresh(var space:single ):single;
 function OPDVThresh(var space:single ):single;
 function ABXThresh(var speed,rand:single ):single;
 function SDXThresh(var space,speed,rand:single ):single;
```

```
 function StimuLC_left(Var Follower_speed,Follower_DX:single):single;
 function DCnull(var space,speed:single ):single;
 function ACnull : single;
implementation

function DCnull(var space,speed:single ):single;
var
per,DCmin,TCT,Y :single;
begin
 per:=100*random;
 if Speed <>0 then TCT := ABS(Space/Speed) else TCT :=100;
 if TCT>65 then
  DCmin := -0.145
 else begin
  DCmin:=-0.00008876*sqr(TCT)+0.011939*TCT-0.55051;
  if DCmin < -0.45 then DCmin := -0.45;
 end;
 Y:=-0.003054*per-0.14593;
 if Y > DCmin then Y:=DCmin;
 DCnull := Y;
end;{end Function
--------------------------}
function ACnull : single;
var
 per : single;
begin
 per := 100*random;
 ACnull:=0.0000181*sqr(per)+0.0016308*per+0.14236;
end;{end Function
--------------------------}
function StimuLC_left(Var Follower_speed,Follower_DX:single):single;
 var
 j,SDX,ABX : single;
begin
 if (Follower_speed)<>0 then begin
  SDX:=EXP(ln(2.0687)+0.6944*ln(Follower_speed*3.6));
  ABX:=EXP(ln(3.2485)+0.2118*ln(Follower_speed*3.6));

  if SDX <(ABX+5) then SDX:=ABX+5;
  if (Follower_DX>SDX)then
   J:=0.0
  else begin
   if (Follower_DX<ABX)then
    J:=1.0
   else begin
    J := (SDX-Follower_DX)/(SDX-ABX);
   end;
  end;
  StimuLC_left := J;
 end
 else begin
  StimuLC_left := 0.0;
 end;
end; {end of function StimuLC_left
-------------------------------}
function CLDVThresh(var space:single ):single;
var
```

```
 per,DVmin,Y,x,Z1,Z2 :single;
 begin
  per:=100*random; DVmin:=-0.01*Space;
  if PER >= 99 then begin CLDVThresh:=-2.5;Exit;end;
  if space <=20 then begin
   y:=-16.1*(sqr(Dvmin)*DVmin)-96.01*Sqr(DVmin)-191.2*Dvmin-28.733;
   if per <= Y then begin
    CLDVThresh:=DVmin;Exit;
   end
   else begin
    Z2:=-2.5;Z1:=DVmin;
    X:=(Z1+Z2)/2;
    y:=-16.1*(sqr(X)*X)-96.01*Sqr(X)-191.2*X-28.733;
    while ABS(y-per)>0.05 do begin
     if per > Y then z1:=X else z2:=x;
     X:=(Z1+Z2)/2;
     y:=-16.1*(sqr(X)*X)-96.01*Sqr(X)-191.2*X-28.733;
    end;
    CLDVThresh:=X;Exit;
   end;
  end
  else begin
   y:=-6.6715*(sqr(Dvmin)*DVmin)-54.11*Sqr(DVmin)-151.73*Dvmin-45.325;
   if per <= Y then begin
    CLDVThresh:=DVmin;Exit;
   end
   else begin
    Z2:=-2.5;Z1:=DVmin;
    X:=(Z1+Z2)/2;
    y:=-6.6715*(sqr(X)*X)-54.11*Sqr(X)-151.73*X-45.325;
    while ABS(y-per)>0.05 do begin
     if per > Y then z1:=X else z2:=x;
     X:=(Z1+Z2)/2;
     y:=-6.6715*(sqr(X)*X)-54.11*Sqr(X)-151.73*X-45.325;
    end;
    CLDVThresh:=X;Exit;
   end;
  end;
end;{ end of function CLDVThresh
---------------------------------}
function OPDVThresh(var space:single ):single;
var
per,DVmin,Y,x,Z1,Z2 :single;
begin
 per:=100*random; DVmin:=0.01*Space;
 if PER >= 99 then begin OPDVThresh:=2.5;Exit;end;
 if space <=20 then begin
  y:=10.863*(sqr(Dvmin)*DVmin)-73.993*Sqr(DVmin)+167.79*Dvmin-27.066;
  if per <= Y then begin
   OPDVThresh:=DVmin;
  end
  else begin
   Z2:=2.5;Z1:=DVmin;
   X:=(Z1+Z2)/2;
   y:=10.863*(sqr(X)*X)-73.993*Sqr(X)+167.79*X-27.066;
   while ABS(y-per)>0.05 do begin
    if per > Y then z1:=X else z2:=x;
```

```
    X:=(Z1+Z2)/2;
    y:=10.863*(sqr(X)*X)-73.993*Sqr(X)+167.79*X-27.066;
   end;
   OPDVThresh:=X;
  end;
 end
 else begin
  y:=3.744*(sqr(Dvmin)*DVmin)-39.95*Sqr(DVmin)+131.83*Dvmin-38.607;
  if per <= Y then begin
   OPDVThresh:=DVmin;
  end
  else begin
   Z2:=2.5;Z1:=DVmin;
   X:=(Z1+Z2)/2;
   y:=3.744*(sqr(X)*X)-39.95*Sqr(X)+131.83*X-38.607;
   while ABS(y-per)>0.05 do begin
    if per > Y then z1:=X else z2:=x;
    X:=(Z1+Z2)/2;
    y:=3.744*(sqr(X)*X)-39.95*Sqr(X)+131.83*X-38.607;
   end;
   OPDVThresh:=X;
  end;
 end;
end;{ end of function OPDVThresh
---------------------------------}
function DVperception(var space,Speed,theta,theta1:single;
DVper1:char):char;
var
 per,TTC   :single;
 N1        :integer;
begin
 TTC:=Speed/Space;
 if         (TTC < -0.11)        then begin DVperception:='c';Exit;end;
 if (TTC >-0.01) and (TTC <0.01) then begin DVperception:='s';Exit;end;
 if          (TTC >0.09)         then begin DVperception:='o';Exit;end;
 if (theta1 > theta) and (DVper1='c')and (space<100)and(TTC<-0.01) then
 begin DVperception:='c';Exit;end;
 if (theta1 < theta) and (DVper1='o')and (space<100)and(TTC>0.01) then
 begin DVperception:='o';Exit;end;
  {--------------------------------}
 if TTC < -0.01 then begin
  N1:=0;per:=-0.01;repeat n1:=n1+1; until (TTC>(per-(N1*0.02)))or(n1=5);
  per :=100*random;
  if per <=DVClose[n1] then begin
   DVperception:='c';Exit;
  end
  else begin DVperception:='s';Exit;end;
 end
 else begin
  if TTC >0.01 then begin
   N1:=0;per:=0.01;repeat n1:=n1+1; until (TTC<(per+(N1*0.02)))or(n1=3);
   per :=100*random;
   if per <=DVopen[n1] then begin
    DVperception:='o';Exit;
   end
   else begin DVperception:='s';Exit;end;
  end;
```

```
 end;
end;{ End of function DVper
----------------------------------}
function ABXThresh(var speed,rand:single ):single;
var
ABXmin,ABXmax,V,BX  :single;
ABXper  :array[1..5] of single;
begin
 V:=Speed*3.6;
 if V>10.0 then begin

ABXmin:=exp(ln(3.2485)+0.2118*ln(V));ABXmax:=EXP(ln(1.6184)+0.6655*ln(V));
 end
 else begin
  ABXmin:=5.3;ABXmax:=7.5;
 end;
{if V <= 55 then begin
  ABXmin:=-0.001159*SQR(V)+0.174643*V+2.276; ABXmax:=(-7.1772E-
05)*sqr(V)*V+0.01135*sqr(V)-0.129646*V+9.7725;
 end
 else begin
  ABXmin:=0.0198*V+6.3771; ABXmax:=0.2518*V+9.634;
 end;}
 ABXper[1]:=(((7.035E-06)*sqr(V)*Sqr(V))-((2.3742E-
03)*Sqr(V)*V)+0.28992*sqr(V)-15.198*V+297.746)/100;
 if V >125 then begin
  ABXper[1]:=0.07;
  ABXper[2]:=(-((555.36E-06)*sqr(V)*V)+0.1571*sqr(V)-
14.3114*V+450.14)/100;
  if ABXper[2]< 0 then ABXper[2] :=0.0;
  if ABXper[2]< ABXper[1] then ABXper[1]:=ABXper[2];
 end
 else begin
  if ABXper[1] >1 then ABXper[1]:=1;
 end;
 if rand < ABXper[1] then begin
  BX := ABXmin+((10-ABXmin)* RAND / ABXper[1]);
 end
 else begin
  ABXper[2]:=(-((555.36E-06)*sqr(V)*V)+0.1571*sqr(V)-
14.3114*V+450.14)/100;
  if ABXper[2] >1 then ABXper[2]:=1;
  if ABXper[2]< 0 then ABXper[2] :=0.0;
  if rand < ABXper[2] then begin
    if ABXper[2]<1 then
     BX := 10+(5*(RAND-ABXper[1])/(ABXper[2]-ABXper[1]))
    else begin
     if ABXmax <=15 then
      BX := 10+((ABXmax-10)*(RAND-ABXper[1])/(1-ABXper[1]))
     else BX := 10+(5*(RAND-ABXper[1])/(ABXper[2]-ABXper[1]));
    end;
  end
  else begin
   ABXper[3]:=((-(368.84E-06)*sqr(V)*V)+0.11646*sqr(V)-
11.7954*V+442.64)/100;
   if ABXper[3] >1 then ABXper[3]:=1;
   if rand < ABXper[3] then begin
```

```
      if ABXper[3]<1 then
       BX := 15+(5*(RAND-ABXper[2])/(ABXper[3]-ABXper[2]))
      else begin
       if ABXmax <=20 then
        BX := 15+((ABXmax-15)*(RAND-ABXper[2])/(1-ABXper[2]))
       else BX := 15+(5*(RAND-ABXper[2])/(ABXper[3]-ABXper[2]));
      end;
    end
    else begin
     ABXper[4]:=((20.822E-03)*sqr(V)-4.2138*V+290.08)/100;
     if ABXper[4] >1 then ABXper[4]:=1;
     if rand < ABXper[4] then begin
       if ABXper[4]<1 then
        BX := 20+(5*(RAND-ABXper[3])/(ABXper[4]-ABXper[3]))
       else begin
        if ABXmax <=25 then
         BX := 20+((ABXmax-20)*(RAND-ABXper[3])/(1-ABXper[3]))
        else BX := 20+(5*(RAND-ABXper[3])/(ABXper[4]-ABXper[3]));
       end;
     end
     else begin
      ABXper[5]:=((14.521E-03)*sqr(V)-3.0708*V+250.92)/100;
      if ABXper[5] >1 then ABXper[5]:=1;
      if rand < ABXper[5] then begin
        if ABXper[5]<1 then
         BX := 25+(5*(RAND-ABXper[4])/(ABXper[5]-ABXper[4]))
        else begin
         if ABXmax <=30 then
          BX := 25+((ABXmax-25)*(RAND-ABXper[4])/(1-ABXper[4]))
         else BX := 25+(5*(RAND-ABXper[4])/(ABXper[5]-ABXper[4]));
        end;
      end
      else begin
       BX := 30+((ABXmax-30)*(RAND-ABXper[5])/(1-ABXper[5]));
      end;
     end;
    end;
   end;
  end;
  if BX< ABXmin then BX := ABXmin;
  if BX> ABXmax then BX := ABXmax;
  ABXThresh:=BX;
end;{ End of function ABXThresh
-----------------------------------}
function SDXThresh(var space,speed,rand:single ):single;
var
 SDXmin,SDXmax,SDXpermin,SDXpermax,V,SDX     :single;
 SDXper                       : array[1..7] of single;
begin
 V:=Speed*3.6;
 {if rand >0.5 then rand:=(rand-0.25);}
{
 if V <=55 then begin
 SDXmin:=(-1.2582E-03)*sqr(V)+0.221909*V+4.0235;
 SDXmax:=(-2.3317E-03)*sqr(v)+0.719573*V+2.5486;
 if SDXmax < SDXmin then SDXmax := SDXmin;
end
```

```
else begin
 SDXmin:=0.0458*V+8.6894; SDXmax:=0.3931*V+11.195;
end;}
 if (V>10.0) then begin
  SDXmin:=EXP(ln(3.4033)+0.2976*ln(v));
SDXmax:=EXP(ln(2.0687)+0.6944*ln(v));
 end
 else begin
  SDXmin:=6.75; SDXmax:=10.23;
 end;
 SDXpermin:=1.139;  SDXpermax:=(104.96E-06)*sqr(V)-(8.0921E-03)*V+2.0991;
 SDXper[1]:=(-0.06*V+19.68)/100;
 if SDXper[1] <0.0 then SDXper[1]:=0.0;
 if rand < SDXper[1] then
  SDX := SDXpermin+((1.25-SDXpermin)* RAND / SDXper[1])
 else begin
  SDXper[2]:=(-0.17*V+63.2)/100;
  if SDXper[2] <0.0 then SDXper[2]:=0.0;
  if rand < SDXper[2] then begin
    if SDXper[2] <1 then
     SDX := 1.25+(0.25*(RAND-SDXper[1])/(SDXper[2]-SDXper[1]))
    else
     SDX := 1.25+((SDXpermax-1.25)*(RAND-SDXper[1])/(1-SDXper[1]));
  end
  else begin
   SDXper[3]:=(-0.11*V+84.55)/100;
   if SDXper[3] <0.0 then SDXper[3]:=0.0;
   if rand < SDXper[3] then begin
     if SDXper[3] <1 then
      SDX := 1.5+(0.25*(RAND-SDXper[2])/(SDXper[3]-SDXper[2]))
     else
      SDX := 1.5+((SDXpermax-1.5)*(RAND-SDXper[2])/(1-SDXper[2]));
   end
   else begin
    SDXper[4]:=(-0.16*V+102.62)/100;
    if SDXper[4] <0.0 then SDXper[4]:=0.0;
    if rand < SDXper[4] then begin
      if SDXper[4] <1 then
       SDX := 1.75+(0.25*(RAND-SDXper[3])/(SDXper[4]-SDXper[3]))
      else
       SDX := 1.75+((SDXpermax-1.75)*(RAND-SDXper[3])/(1-SDXper[3]));
    end
    else begin
     SDXper[5]:=(-0.37*V+133.4)/100;
     if SDXper[5] <0.0 then SDXper[5]:=0.0;
     if rand < SDXper[5] then begin
       if SDXper[5] <1 then
        SDX := 2+(0.25*(RAND-SDXper[4])/(SDXper[5]-SDXper[4]))
       else begin
         if SDXpermax < 2.25 then
          SDX := 2+((SDXpermax-2)*(RAND-SDXper[4])/(1-SDXper[4]))
         else SDX := 2+(0.25*(RAND-SDXper[4])/(SDXper[5]-SDXper[4]));
       end;
     end
     else begin
      SDXper[6]:=(-0.21*V+121.78)/100;
      if SDXper[6] <0.0 then SDXper[6]:=0.0;
```

```
      if rand < SDXper[6] then begin
        if SDXper[6] <1 then
         SDX := 2.25+(0.25*(RAND-SDXper[5])/(SDXper[6]-SDXper[5]))
        else begin
          if SDXpermax < 2.5 then
           SDX := 2.25+((SDXpermax-2.25)*(RAND-SDXper[5])/(1-SDXper[5]))
          else SDX := 2.25+(0.25*(RAND-SDXper[5])/(SDXper[6]-SDXper[5]));
        end;
      end
      else begin
         SDX := 2.5+((SDXpermax-2.5)*(RAND-SDXper[6])/(1-SDXper[6]));
      end;
     end;
    end;
   end;
  end;
 end;
 SDX := Space*SDX;
 if SDX < SDXmin then SDX :=SDXmin;
 if SDX > SDXmax then begin
  if SDXmax > (1.139*Space) then SDX := SDXmax else SDX:=1.139*Space;
 end;
 SDXThresh:=SDX;
end;{ End of function SDXThresh
---------------------------------}
procedure generator1;
var
 Dspeed,NULL,TTA,TTAmin  : single;
 L,W                     : word;
 K,i                     : Byte;
begin
   turn:=9;
   if norm1[turn]<>1000 then begin
      NULL:=ABS(norm1[turn]);norm1[turn]:=1000;
   end
   else begin
      mean:=0.0;std:=1.0;NULL:=ABS(Normal);
   end;
   p1^.veh.kind :=1;
   if NULL > PNorm[HGV[p1^.lane]] then begin
      p1^.veh.kind :=4;
      if (p1^.lane = 3)then p1^.veh.kind :=1;
   end;
   K:=p1^.veh.kind;
   case k of
    1,2:begin
            turn:=2;
            if norm1[turn]<>1000 then begin
               L:=round(norm1[turn]);norm1[turn]:=1000;
            end
            else begin
             mean:=500;std:=30;L:=round(Normal);
            end;
            turn:=3;
            if norm1[turn]<>1000 then begin
               W:=round(norm1[turn]);norm1[turn]:=1000;
            end
```

```
            else begin
             mean:=200;std:=15;W:=round(Normal);
            end;
            if p1^.lane =1 then turn:=4;
            if p1^.lane =2 then turn:=14;
            if p1^.lane =3 then turn:=15;
            if norm1[turn]<>1000 then begin
               Dspeed:=norm1[turn];norm1[turn]:=1000;
            end
            else begin
             {mean:=ADS-(2-p1^.lane)*sdADS;std:=2*sdADS/3;}
             mean:=ADS;STD:=SDADS;
             Dspeed:=Normal;
            end;
            turn:=11;
            if norm1[turn]<>1000 then begin
              p1^.dvr.RAND[6]:=norm1[turn]/100;norm1[turn]:=1000;
            end
            else begin
              mean:=SpeedT;std:=SpeedTS/10;p1^.dvr.RAND[6]:=Normal/100;
            end;
            turn:=19;
            if norm1[turn]<>1000 then begin
             NULL:=ABS(norm1[turn]);norm1[turn]:=1000;
            end
            else begin
              mean:=0.0;std:=1.0;NULL:=ABS(Normal);
            end;
            p1^.ACC.Opp:='of';
            if NULL>PNorm[ACC_per] then p1^.ACC.Eq:='Y'else
p1^.ACC.Eq:='N';
        end;
    3,4:begin
            turn:=5;
            if norm1[turn]<>1000 then begin
               L:=round(norm1[turn]);norm1[turn]:=1000;
            end
            else begin
             mean:=1100;std:=200;L:=round(Normal);
            end;
            turn:=6;
            if norm1[turn]<>1000 then begin
               W:=round(norm1[turn]);norm1[turn]:=1000;
            end
            else begin
             mean:=240;std:=15;W:=round(Normal);
            end;
            turn:=7;
            if norm1[turn]<>1000 then begin
               Dspeed:=norm1[turn];norm1[turn]:=1000;
            end
            else begin
             mean:=ADSH; std:=sdADSH;
             Dspeed:=Normal;
            end;
            turn:=13;
            if norm1[turn]<>1000 then begin
```

```
             p1^.dvr.RAND[6]:=norm1[turn]/100;norm1[turn]:=1000;
           end
           else begin
             mean:=SpeedTH;std:=SpeedTSH/10;p1^.dvr.RAND[6]:=Normal/100;
           end;
           p1^.ACC.Opp:='of';p1^.ACC.Eq:='N';
       end;
  end;
  repeat
   turn:=1;
   if norm1[turn]<1000 then begin
    TTA:=norm1[turn];norm1[turn]:=1000;
   end
   else begin
    mean:=11.7026;std:=2.292;TTA:=Normal;
   end;
  until (TTA>=8)and(TTA<=20);

  repeat
   turn:=18;
   if norm1[turn]<1000 then begin
    p1^.dvr.RAND[8]:=norm1[turn];norm1[turn]:=1000;
   end
   else begin
    mean:=R8/100;std:=R8std/100;p1^.dvr.RAND[8]:=Normal;
   end;
  until (p1^.dvr.RAND[8]>((R8-R8std)/100))and(p1^.dvr.RAND[8]<1);

  repeat
   turn:=7;
   if lognorm1[turn]<>1000 then begin
     TTAmin:=lognorm1[turn];lognorm1[turn]:=1000;
   end
   else begin
    mean:=8.41; std:=2.1;
    TTAmin:=Lognormal;
   end;
  until ((TTAmin>=5) and (TTAmin<=(TTA-2)));
  p1^.TTA:=TTA;p1^.TTAmin:=TTAmin;
  p1^.back :=nil; p1^.ba_right:=nil; p1^.ba_left:=nil; p1^.takeveh:=nil;
  {--------------}
  p1^.dvr.NULL:=0.2; p1^.dvr.NULL1:=-0.2;
  {--------------}
  p1^.ahead   :=start[p1^.lane];
  p1^.ah_left :=start[p1^.lane-1];
  p1^.ah_right:=start[p1^.lane+1];
  p1^.ABXTimer:=0;
  if p1^.ahead<>nil then begin
   p1^.NO_ahead:=p1^.ahead^.veh.Number;
  end
  else p1^.NO_ahead:=0;
  {P1^.Journey :=0.0;}
  P1^.overtake :=0; P1^.overtake1:=0; P1^.overtake2:=0;
  p1^.OvertakeTime :=0.0;
  P1^.factor    :=0; P1^.factor2  :=0;
  P1^.factor3   :=0; p1^.factorth :=0;
  p1^.F2        :=0; p1^.F3       :=0;
```

```
P1^.Olane:=p1^.lane;
p1^.stimu     :=0.0;
for i:=0 to 2 do begin
 p1^.theta[i] :=0.0; p1^.Dvper[i] :='s';
 p1^.EST[i] :=100;   p1^.odv1[i]  :=0.0;  P1^.PDC[i] :=0.0;
end;
for i:=0 to 12 do  P1^.dc[i] :=0.0;
Space:=25;
P1^.cldv:=CLDVthresh(Space); P1^.opdv:=OPDVthresh(Space);
{randomize;}
p1^.dvr.RAND[1] := random;{for the ABX}
p1^.dvr.RAND[2] := random;{for the SDX}
p1^.dvr.RAND[5] := 100*Random;
turn:=16;
if norm1[turn]<>1000 then begin
  p1^.dvr.RAND[7]:=norm1[turn];norm1[turn]:=1000;
end
else begin
 mean:=2;std:=1;p1^.dvr.RAND[7]:=Normal;
end;
turn:=12;
if norm1[turn]<>1000 then begin
   p1^.LCT:=Round(norm1[turn]);norm1[turn]:=1000;
end
else begin
   mean:=LCTime;std:=LCtimeS/10;p1^.LCT:=round(Normal);
end;

if p1^.veh.kind >2 then p1^.dvr.RAND[5] := 0.15;
repeat
 turn:=3;
 if lognorm1[turn]<>1000 then begin
   p1^.dvr.HBLC :=lognorm1[turn];lognorm1[turn]:=1000;
 end
 else begin
  mean:=0.98157; std:=0.48687; p1^.dvr.HBLC :=Lognormal;
 end;
until ((p1^.dvr.HBLC <2.4)and(p1^.dvr.HBLC >0.3));

turn:=10;
if norm1[turn]<>1000 then begin
   p1^.dvr.RAND[4]:=round(norm1[turn]);norm1[turn]:=1000;
end
else begin
   mean:=0.0;std:=0.1;p1^.dvr.RAND[4]:=round(Normal);
end;
P1^.dvr.Dspeed:= Dspeed;P1^.dvr.Cdspeed:= Dspeed;
P1^.veh.lenV:= L; P1^.veh.WidV := W;P1^.veh.Kind  := K;
case k of
 1,2:begin
      case p1^.lane of
      1:begin
         if Dspeed <=34.8 then
             p1^.Flane:=1
         else begin
            if Dspeed <=37.8 then p1^.Flane:=2 else p1^.Flane:=3;
         end;
```

```
          end;
        2:begin
           if Dspeed <=25.4 then
              p1^.Flane:=1
           else begin
              if Dspeed >37.8 then p1^.Flane:=3 else p1^.Flane:=2;
           end;
          end;
        3:begin
           if Dspeed >=30.42 then
              p1^.Flane:=3
           else begin
              if Dspeed <25.4 then p1^.Flane:=1 else p1^.Flane:=2;
           end;
          end;
        end;
       end;
    3,4:begin p1^.Flane:=p1^.Lane;end;
   end;
   {-----------------------}
   if p1^.ACC.EQ ='Y' then begin
     p1^.ACC.Thmax_D    := Hmax_D;
     p1^.ACC.Thmax_C    := Hmax_C;
     turn:=20;
     repeat
      if norm1[turn]<1000 then begin
       p1^.ACC.Th_D:=norm1[turn];norm1[turn]:=1000;
      end
      else begin
       mean:=Th_D;std:=0.2;p1^.ACC.Th_D:=Normal;
      end;
     until (p1^.ACC.Th_D>=(Th_D-
0.4))and(p1^.ACC.Th_D<=(Th_D+0.4))and(p1^.ACC.Th_D<=p1^.ACC.Thmax_D);
     turn:=21;
     repeat
      if norm1[turn]<1000 then begin
       p1^.ACC.Tmin_R:=norm1[turn];norm1[turn]:=1000;
      end
      else begin
       mean:= Tmin_R;std:=0.2;p1^.ACC.Tmin_R:=Normal;
      end;
     until (p1^.ACC.Tmin_R>=(Tmin_R-10))and(p1^.ACC.Tmin_R<=(Tmin_R+10));
     p1^.ACC.ACmax      := ACC_ACmax;
     p1^.ACC.DCmax      := ACC_DCmax;
     p1^.ACC.DC_Eng     := -0.5;
     p1^.ACC.Vmin_D     := 13.89;
     p1^.ACC.Vmax_D     := 1.1*p1^.dvr.dspeed;
     p1^.ACC.Vmin_AC    := 13.89;
     p1^.ACC.Timer      := 0.0;
   end
   else begin
     p1^.ACC.Thmax_D    := Hmax_D;
     p1^.ACC.Thmax_C    := Hmax_C;
     p1^.ACC.Th_D       := Th_D;
     p1^.ACC.Tmin_R     := Tmin_R;
     p1^.ACC.ACmax      := ACC_ACmax;
     p1^.ACC.DCmax      := ACC_DCmax;
```

```
      p1^.ACC.DC_Eng      := -0.5;
      p1^.ACC.Vmin_D      := 13.89;
      p1^.ACC.Vmax_D      := 1.1*p1^.dvr.dspeed;
      p1^.ACC.Vmin_AC     := 13.89;
      p1^.ACC.Timer       := 0.0;
    end;
    {----------------------}
    for i:=0 to 12 do begin
     p1^.x[i]:=10.0;p1^.V[i]:=0.0;
    end;
    p1^.x[4]:=10.0;
    if (p1^.ahead<>nil)and(((p1^.ahead^.x[4]- p1^.x[4])/p1^.dvr.dspeed)<4)
then begin
      if p1^.ahead^.v[0] > p1^.dvr.dspeed then begin
       for i:=0 to 4 do p1^.v[i]  := p1^.dvr.dspeed;
       for i:=3 downto 0 do p1^.x[i]:= p1^.x[i+1]+ 0.125 * p1^.v[i+1];
      end
      else begin
       for i:=0 to 4 do p1^.v[i]  := p1^.ahead^.v[i];
       for i:=3 downto 0 do p1^.x[i]:= p1^.x[i+1]+ 0.125 * p1^.v[i+1];
      end;
      if p1^.x[0] > (p1^.ahead^.x[0]- (p1^.ahead^.veh.lenV/100)-2) then
begin
       p1^.x[0]:=p1^.ahead^.x[0]- (p1^.ahead^.veh.lenV/100)-2;
       if p1^.ahead^.v[0] >0.0 then begin
        for i:=0 to 4 do p1^.v[i]  := 0.8*p1^.ahead^.v[0];
        for i:=1 to 4 do p1^.v[i]  := p1^.x[i-1]-p1^.V[i]*0.125;
       end
       else begin
        for i:=0 to 4 do begin
         p1^.v[i]  := 0.0;p1^.x[i]:=p1^.x[1];
        end;
       end;
      end;
    end
    else begin
     for i:=0 to 4 do p1^.v[i]  := p1^.dvr.dspeed;
     for i:=3 downto 0 do p1^.x[i]:= p1^.x[i+1]+ 0.125 * p1^.v[i+1];
    end;
{----------------------}
end;{ procedure
----------------------------------------------------------------}
procedure generator;
var
Lan,Fact     :byte;
begin
  Lan:=1;
  while Lan<=3 do  begin
   if T >= TF[Lan,2] then begin
     turn:=lan+3;
     if lognorm1[turn] <> 1000.0 then begin
       TF[lan,1] :=lognorm1[turn];lognorm1[turn]:=1000.0;
     end
     else begin
       mean :=FMH[lan]; STD:= 0.4;TF [lan,1]:= lognormal;
     end;
     mean := random;
```

```
      if mean > phai[lan] then begin
        mean:=0.0; while mean <=0.0 do begin mean:=random;end;
        TF[lan,1] := TF[lan,1]- (ln(mean)/lampda[lan]);
      end;
      if TF [lan,1]<0.5 then TF[lan,1]:=0.5;
      TF[lan,2] := T + TF[lan,1];
      if not((start[lan]<>nil)and((start[lan]^.x[8]-
(start[lan]^.veh.lenV/100)-2) <=12))then begin
       number := number +1;if number >65000 then number :=1;
       new (p1);
       p1^.lane := lan;p1^.olane := lan;P1^.veh.number :=number;
       generator1;
       if p1^.ahead =nil then lastv[lan]:=p1 else start[lan]^.back:=p1;
       start[lan]:=p1;
       p2 := start[lan+1];
       while (p2<>nil) and (p2^.ba_left = nil) do begin
        p2^.ba_left := start[lan]; p2:=p2^.ahead;
       end;
       p2 :=start[lan-1];
       while (p2<>nil) and (p2^.ba_right = nil) do begin
        p2^.ba_right := start[lan]; p2:=p2^.ahead;
       end;
      end;
    end;
    lan:=lan+1;
  end;
end;{ end of procedure
----------------------------------------------------------------}
procedure final;
var
lan       :byte;
begin
  lan:=1;
  while lan<=3 do  begin
   if (lastv[lan] <> nil) and (lastv[lan]^.x[0]>=10000) then begin
     p2 := lastv[lan];
     lastv[lan]:= lastv[lan]^.back;
     if lastv[lan] <> nil then begin
      lastv[lan]^.ahead := nil;
      p1 := lastv[lan]^.ah_right; p3:= lastv[lan]^.ah_left;
     end
     else begin
      start[lan]:= nil;p1 :=start[lan+1];p3 := start[lan-1];
     end;
     while (p1<>nil) do begin
      p1^.ah_left := nil;
      p1 :=p1^.ahead;
     end;
     while (p3<>nil) do begin
      p3^.ah_right := nil;
      p3 :=p3^.ahead;
     end;
     dispose(p2);
     if start[lan]=nil then begin
      number:=0;TF[lan,2] := T;  posi:=1;
     end;
   end;
```

```
    lan := lan +1;
   end;
end;{ procedure
---------------------------------------------------------------}
procedure update3;
begin
    p1^.overtake := 11;
    p1^.lane := p1^.lane +1;p1^.takeveh:=p1^.ahead;

    if p1^.ahead <> nil then p1^.ahead^.back := p1^.back
    else begin lastv[p1^.lane -1] := p1^.back;  end;

    if p1^.back  <> nil then p1^.back^.ahead := p1^.ahead
    else begin start[p1^.lane -1] := p1^.ahead; end;

    if p1^.ah_right <> nil then p1^.ah_right^.back  := p1
    else begin lastv[p1^.lane] := p1; end;

    if p1^.ba_right <> nil then p1^.ba_right^.ahead := p1
    else begin start[p1^.lane] := p1; end;
{Updating the new lane ........}
    p2 := p1^.ah_right;
    while (p2 <> nil) and (p2^.ba_left = p1) do begin
       p2^.ba_left := p1^.back;
       p2 := p2^.ahead;
    end;
    p2 := p1^.ba_right;
    while (p2 <> nil) and (p2^.ah_left = p1) do begin
       p2^.ah_left := p1^.ahead;
       p2 := p2^.back;
    end;
    p1^.ah_left := p1^.ahead;
    p1^.ba_left := p1^.back;
    p1^.ahead   := p1^.ah_right;
    p1^.back    := p1^.ba_right;
{Updating the olde lane .........}
    p2:= p1^.ah_left;
    while (p2 <> nil) and (p2^.ba_right = p1^.back) do begin
      p2^.ba_right := p1;
      p2 := p2^.ahead;
    end;
    p2:=p1^.ba_left;
    while (p2 <> nil) and (p2^.ah_right = p1^.ahead) do begin
      p2^.ah_right := p1;
      p2 := p2^.back;
    end;
{Updating the right lane of the New lane ........}
    if p1^.lane = 2 then begin
     if p1 <> lastv[p1^.lane] then
     begin p2 := p1^.ahead^.ba_right;p1^.ah_right:=p1^.ahead^.ah_right;end
     else begin p2 := lastv[p1^.lane +1]; p1^.ah_right := nil; end;
     while (p2<> nil) and
           (p2^.x[0] > p1^.x[0]) and (p2^.ba_left = p1^.back) do begin
       p2^.ba_left := p1;
       p1^.ah_right := p2;
       p2 := p2^.back;
     end;
```

```
      if p1 <> start[p1^.lane] then begin
        p2 := p1^.back^.ah_right;
        p1^.ba_right := p1^.back^.ba_right;
      end
      else begin
        p2 := start[p1^.lane +1];
        p1^.ba_right :=nil;
      end;
      while (p2<> nil) and
            (p2^.x[0] <= p1^.x[0]) and (p2^.ah_left = p1^.ahead) do begin
        p2^.ah_left := p1;
        p1^.ba_right :=p2;
        p2:=p2^.ahead;
      end;
     end;
{Updating the left lane of the old lane .......}
     if p1^.lane =3 then begin
      p1^.ah_right:= nil; p1^.ba_right:= nil;
      if p1^.ah_left <> nil then
      p2 := p1^.ah_left^.ba_left
      else begin p2 := lastv[p1^.lane - 2]; end;
      while (p2<> nil) and (p2^.ba_right = p1) do begin
       p2^.ba_right := p1^.ba_left;
       p2 := p2^.back;
      end;
      if p1^.ba_left <>nil then
      p2 := p1^.ba_left^.ah_left
      else begin p2 := start[p1^.lane - 2]; end;
      while (p2<> nil) and (p2^.ah_right = p1) do begin
        p2^.ah_right := p1^.ah_left;
        p2 := p2^.ahead
      end;
     end;
end;{ End of procedure
---------------------------------------------------------------}
procedure update4;
begin
    p1^.overtake := 22;  p1^.lane := p1^.lane - 1;
    p1^.takeveh:=p1^.back;
    if p1^.ahead <> nil then p1^.ahead^.back := p1^.back
    else begin lastv[p1^.lane + 1] := p1^.back;  end;
    if p1^.back  <> nil then p1^.back^.ahead := p1^.ahead
    else begin start[p1^.lane + 1] :=p1^.ahead; end;
    if p1^.ah_left <> nil then  p1^.ah_left^.back  := p1
    else begin lastv[p1^.lane] := p1; end;
    if p1^.ba_left <> nil then  p1^.ba_left^.ahead := p1
    else begin start[p1^.lane] := p1; end;
{Updating the new lane ........}
    p2 := p1^.ah_left;
     while (p2 <> nil) and (p2^.ba_right = p1) do begin
         p2^.ba_right := p1^.back;
         p2 := p2^.ahead;
     end;
    p2 := p1^.ba_left;
     while (p2 <> nil) and (p2^.ah_right = p1) do begin
         p2^.ah_right := p1^.ahead;
         p2 := p2^.back;
```

```
      end;
     p1^.ah_right := p1^.ahead;     p1^.ba_right := p1^.back;
     p1^.ahead    := p1^.ah_left;   p1^.back     := p1^.ba_left;
 {Updating the olde lane .........}
     p2:= p1^.ah_right;
     while (p2 <> nil) and (p2^.ba_left = p1^.back) do begin
       p2^.ba_left := p1; p2 := p2^.ahead;
     end;
     p2:=p1^.ba_right;
     while (p2 <> nil) and (p2^.ah_left = p1^.ahead) do begin
       p2^.ah_left := p1; p2 := p2^.back;
     end;
 {Updating the left lane of the New lane ........}
     If p1^.lane =2 then begin
      if p1 <> lastv[p1^.lane] then
      begin p2 := p1^.ahead^.ba_left; p1^.ah_left := p1^.ahead^.ah_left;
end
      else begin p2 := lastv[p1^.lane -1]; p1^.ah_left := nil; end;
      while (p2<> nil) and
             (p2^.x[0] > p1^.x[0]) and (p2^.ba_right = p1^.back) do begin
        p2^.ba_right := p1;p1^.ah_left := p2; p2:=p2^.back;
      end;
      if p1 <> start[p1^.lane] then
      begin p2 := p1^.back^.ah_left; p1^.ba_left :=p1^.back^.ba_left; end
      else begin p2 := start[p1^.lane -1]; p1^.ba_left := nil; end;
      while (p2<> nil) and
             (p2^.x[0] <= p1^.x[0]) and (p2^.ah_right = p1^.ahead) do begin
        p2^.ah_right := p1;p1^.ba_left := p2; p2:=p2^.ahead;
      end;
     end;
 {Updating the right lane of the old lane ........}
     if p1^.lane =1 then begin
      p1^.ah_left:= nil; p1^.ba_left:= nil;
      if p1^.ah_right <>nil then p2 := p1^.ah_right^.ba_right
      else begin p2 := lastv[p1^.lane +2]; end;
      while (p2<> nil) and (p2^.ba_left = p1) do begin
        p2^.ba_left := p1^.ba_right; p2:=p2^.back;
      end;
      if p1^.ba_right <>nil then p2 := p1^.ba_right^.ah_right
      else begin p2 := start[p1^.lane +2]; end;
      while (p2<> nil) and (p2^.ah_left = p1) do begin
        p2^.ah_left := p1^.ah_right; p2:=p2^.ahead
      end;
     end;
end;{ procedure
---------------------------------------------------------------}
procedure update1;
var i :byte;
begin
   for i:=12 downto 1 do begin
    p1^.x[i] := p1^.x[i-1]; p1^.v[i] := p1^.v[i-1];p1^.dc[i] := p1^.dc[i-
1];
   end;
   p1^.x[0]  := p1^.x[1] +p1^.v[1] * 0.125 + 0.5*sqr(0.125)* p1^.dc[1];
   if p1^.x[0] < p1^.x[1] then p1^.x[0] := p1^.x[1];
   p1^.v[0]  := p1^.v[0] + 0.125 *p1^.dc[1];
   if p1^.v[0] < 0 then begin
```

```
     p1^.v[0] :=0.0; p1^.x[0]  := p1^.x[0]-(1.5*sqr(p1^.v[1])/p1^.dc[1]);
    end;
    p2:=p1^.ahead;
    if(p2<>nil)and((p2^.x[0]-(p2^.veh.lenV/100)-1)<p1^.x[0])then begin
     p1^.x[0]:=p2^.x[0]-(p2^.veh.lenV/100)-1;
     p1^.v[0]:=p2^.v[0]-0.2;
     if p1^.V[0] <0.0 then p1^.V[0]:=0.0;
     p1^.dc[0] := p2^.dc[0]-0.4;
    end;
end;{ procedure
-------------------------------------------------------------}
procedure update2;
begin
   p2 := p1^.ah_right;
   While (p2 <> nil) and (p1^.x[0] > p2^.x[0]) do begin
      p1^.ah_right := p2^.ahead; p1^.ba_right := p2;
      p2^.ba_left  := p1^.back ; p2^.ah_left  := p1;
      p2 := p2^.ahead;
    end;
   p2 := p1^.ba_right;
   while (p2 <> nil) and (p1^.x[0] <= p2^.x[0]) do begin
      p1^.ba_right := p2^.back;  p1^.ah_right := p2;
      p2^.ah_left  := p1^.ahead; p2^.ba_left  := p1;
      p2 :=p2^.back;
   end;
end;{ procedure
-------------------------------------------------------------}
procedure update5;
begin
 p1^.olane := p1^.lane;
 if (p1^.overtake = 22) then begin
  p1^.overtakeTime := p1^.overtakeTime +0.125;
 end;
 if (p1^.overtake = 11)  then begin
  p1^.overtakeTime := p1^.overtakeTime +0.125;
 end;
 p1 := p1^.ahead;
end;{ procedure
-------------------------------------------------------------}
procedure Updatep;
begin
 p1 := start[1];while (p1<> nil) do begin update5;  end;
 p1 := start[2];while (p1<> nil) do begin update5;  end;
 p1 := start[3];while (p1<> nil) do begin update5;  end;
 {---------------}
 p1 := start[1];
 while (p1<> nil) do begin
  p3 := p1^.ahead;
  case p1^.overtake of
    1: begin update3; p1 := p3; end;
  else begin p1 := p3;end;
  end;
 end;
 p1 := start[2];
 while (p1<> nil) do begin
  p3 := p1^.ahead;
  case p1^.overtake of
```

```
    1: begin update3; p1 := p3; end;
    2: begin update4; p1 := p3; end;
  else begin p1 := p3;end;
  end;
 end;
 p1 := start[3];
 while (p1<> nil) do begin
  p3 := p1^.ahead;
  case p1^.overtake of
    2: begin update4; p1 := p3; end;
  else begin p1 := p3;end;
  end;
 end;
 {---------------}
 p1 := lastV[1]; while (p1<> nil) do begin update1; p1 := p1^.back; end;
 p1 := lastV[2]; while (p1<> nil) do begin update1; p1 := p1^.back; end;
 p1 := lastV[3]; while (p1<> nil) do begin update1; p1 := p1^.back; end;
 {---------------}
 p1 := lastV[1]; while (p1<> nil) do begin update2; p1 := p1^.back; end;
 p1 := lastV[2]; while (p1<> nil) do begin update2; p1 := p1^.back; end;
end;{ procedure
---------------------------------------------------------------}
begin
end.
{------*******************************************************-------}
```