A RULE-BASED SYSTEM FOR PRELIMINARY ACCIDENT RECONSTRUCTION

by

KAROLY WILLIAM KRAJCZAR

B.A.Sc., The University of British Columbia, 1986

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

Department of Civil Engineering

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

July 1989

6

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of CIVIL ENGINEERING

The University of British Columbia
Vancouver, Canada

Date October 5 /89

DE-6 (2/88)

ABSTRACT

Expert systems are being used to solve a number of complex problems that conventional programming techniques have difficulty managing. Problems that are conceptual and cannot be reduced to numbers, can often be solved using expert system technology.

This paper describes the development of a prototype rule-based expert system that can be used during the initial stages of a motor vehicle accident investigation. The program is capable of calculating vehicle speeds using any combination of the following techniques: skid, roll, overturn, yaw, vault, momentum and energy. Although the program's analytical capabilities are very comprehensive, the system's most notable feature is it's interpretive abilities. The program is able to determine how to solve the accident, and verifies the evidence before it is used in the calculations. Additionally, an interactive explanation facility allows the user to examine the program's reasoning.

The existing knowledge base requires further development before it can achieve the level of performance expected from trained accident investigators. However, it demonstrates how expert system technology in conjunction with conventional techniques can be used to solve complex engineering problems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank Dr. Francis P.D. Navin for his supervision and guidance throughout the duration of this project, and Constable Rene Bohemier for his helpful insight in the field of accident investigation.

I would also like to thank Mike McNabb for his patience and suggestions, John Nepomuceno for his artistic talents, and Rob Thomson and Peggy Gibbs for their encouragement.

1. INTRODUCTION

Reconstructing a motor vehicle accident is a very complex task. The primary goals are to clearly understand those factors which contribute to the accident and determine the cause of any subsequent injuries. While engineering mechanics is an integral part of this process, a great deal of conceptualization and reasoning is involved before any calculations can be made. Today, given the correct information and with the aid of computers, trained engineers are able to reconstruct an accident with a fairly high degree of certainty. Unfortunately, complete and accurate information about the accident is often difficult to obtain. Engineers are usually called in weeks after the accident takes place and must rely on data collected by the police during the initial investigation.

The objective of this study is to develop a rule-based expert system for use during the initial stages of a motor vehicle accident reconstruction. A commercial expert system shell, VP-Expert, is used to develop a prototype system called KAR (Knowledge-base Accident Reconstruction). Although the program developed in this paper can calculate speed estimates, its primary intent is to provide guidance and insight for non-technical users involved in the initial investigation. By helping the police identify all pertinent evidence and supplying them with a speed estimate, a decision can be made immediately as to whether a more indepth reconstruction is required.

1.1 An Overview of Artificial Intelligence

Artificial Intelligence (AI) is a branch of computer science that is aimed at developing systems that can mimic the human decision-making process. Some of the major topics investigated by AI researchers include: mathematical problem-solving and theorem proving, speech and voice recognition, visual image recognition, natural language understanding, robotics, and expert systems. Expert Systems is one area of AI that has produced a number of successful applications (i.e. PROSPECTOR [5], PUFF [7], DENDRAL [12], R1 [13], MYCIN [18]). Consequently, government and industry are beginning to look at other applications of this new technology.

1.2 Characteristics of Expert System Tasks

Before an expert system strategy is selected, it must be determined whether the problem is suited to this approach. Problems that exhibit the following characteristics are usually good candidates for an expert system: (i) there are many possible solutions; (ii) the problem-solving expertise is conceptual and cannot be reduced to numbers; (iii) the information needed is incomplete, uncertain, subjective, inconsistent and subject to change; (iv) the conclusions reached will often be uncertain; (v) experts may disagree on how to solve the problem; (vi) the task is always changing and evolving; and (vii) the cost of a poor or late decision is very high [20]. Medical Diagnosis, diverse data analysis, production scheduling and equipment layout are examples of duties that can be effectively handled by an expert system.

1.3 Expert System Software vs. Conventional Software

Expert systems differ from conventional programs in several ways. Conventional programs deal with largely numerical data and algorithmic processing. Van Horn [20] describes a conventional program as a structured "set of algorithms that contain precisely defined terms which are represented by numbers or numerical relationships. All the data used for the program must be numerical, and if some of the data is not entered a result cannot be obtained. For each set of circumstances there will be one best answer." Problems that cannot be reduced to numbers present difficulties for conventional techniques.

Expert systems take a different approach to problem solving. These programs handle symbolic information as opposed to numerical information, and use heuristic processing in contrast to algorithmic processing. Heuristic knowledge is based on "trail and error". This knowledge has not been rigorously proven, but through practice has demonstrated its usefulness and reliability. Heuristic statements generally take the form of "if-then rules". It is the heuristic knowledge of a seasoned practitioner that an expert system tries to model. Hunt [9] defines an expert system as " an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require human expertise for their solution."

1.4 Anatomy of an Expert System

Some critics believe that an expert system must incorporate natural language understanding and automatic learning before it can be referred to as an expert system. Others refer to a program comprised of "if-then rules" as an expert system. Due to this inconsistency, more precise terms are used to identify systems that may not exhibit all of the essential expert system characteristics. Knowledge-based systems, rule-based systems, and knowledge-processing systems are often used to refer to "expert systems".

The ideal expert system should comprise six major components: a knowledge base, an inference engine, an explanation facility, a natural language interface, a knowledge

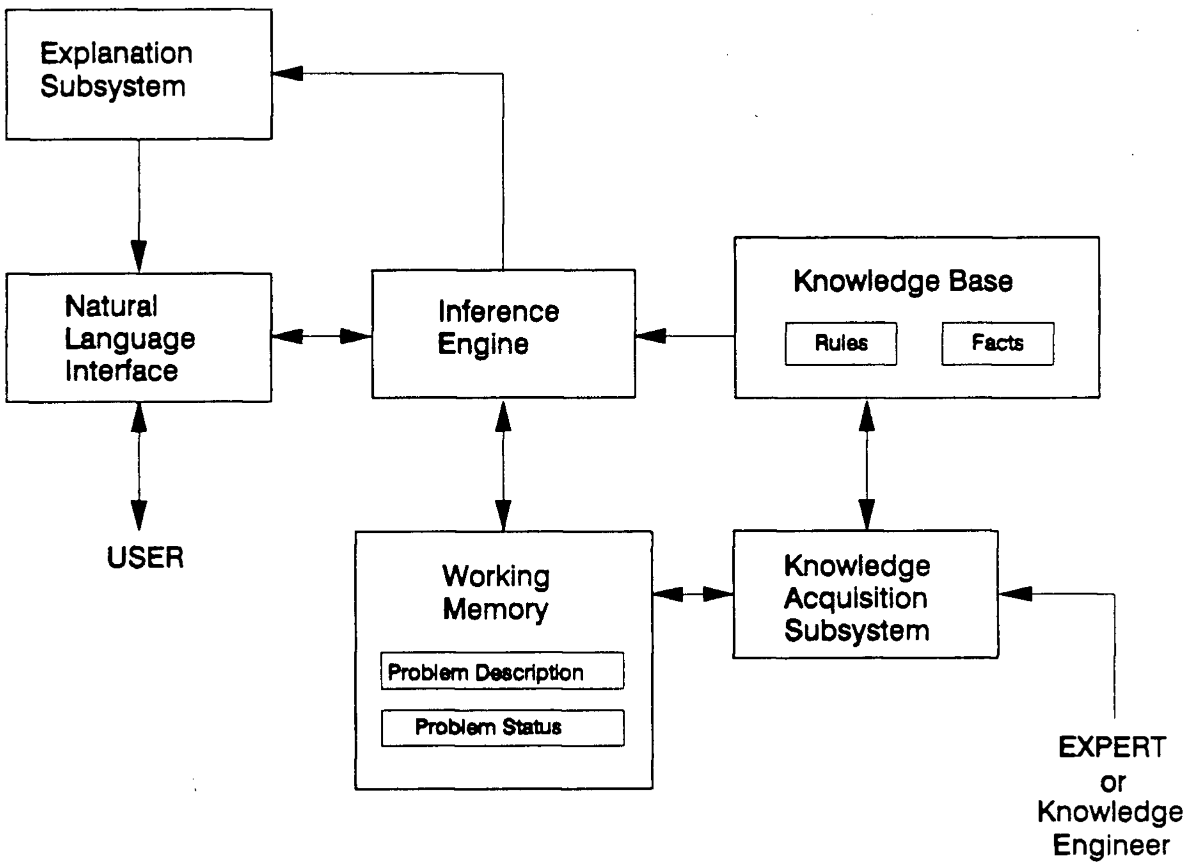


Figure 1: Anatomy of an Ideal Expert System

acquisition subsystem, and working memory (Figure 1). However, no existing expert system contains all of these components, but several of these components occur in every system. Each of these is discussed briefly in the following sections.

1.4.1 The Knowledge Base

Collecting specific knowledge from an expert or a group of experts is referred to as knowledge acquisition. This is the responsibility of a knowledge engineer who obtains this knowledge from a recognized expert and encodes it in the appropriate form for computation. Throughout the knowledge acquisition stage rules and relationships about a specific field are defined and reworked to best represent the expert's heuristic knowledge. This collection of rules and facts is referred to as the knowledge base.

There are currently three principle mechanisms for representing knowledge in an expert system: production rules, frames and predicate logic. An expert system uses one or more of these formalisms as a means of expressing knowledge in the knowledge base.

1.4.1.1 Production Rules

Production rules are conditional statements that take the form:

"If Fact1 and Fact2 then conclude Fact3".

This type of knowledge representation features a rule interpreter that decides how and when to apply certain rules, and a "working memory" that retains data, goals and intermediate results while the program is running.

The *IF* portion of a rule consists of one or more conditions. A condition compares a variable to a value using a relational operator (i.e., =, <, <>, >, >=, <=). The *THEN* portion of the rule can conclude one or more conditions, and is true only when the *IF* conditions are satisfied.

An example of a production rule that might be used to select a suitable cheese to accompany a meal is represented below:

IF
    Complement = bread AND
    Preference = Mild OR
    Preference = Flavorful AND
    Consistency = Firm
THEN
    The_Cheese = Gouda

In this example, each rule condition is tested by questioning the user. The first condition activates the question *"What will you be serving with the cheese?"*. If the response is *"bread"* and the remaining rule conditions are true, Gouda cheese is recommended.

Representing knowledge in this form has certain disadvantages. The rule interpreter is computationally inefficient. In order to obtain an answer it must examine every rule in the knowledge-base. Also, the lack of an explicit framework for representing the knowledge make it very difficult to understand how the rules relate to one another. Additionally, when dealing with very complex domains the syntax becomes cumbersome. Despite these criticisms, rule-based systems are popular. The main reasons being that experts find it relatively easy to express knowledge in this way, and many programmers find it relatively easy to encode in this form.

1.4.1.2 Frames

Marvin Minsky [14], who conceived the idea of frames, believed that the human brain encodes only those salient properties that characterize a specific object, and not by a series of strict, exhaustive definitions of properties.

Frames are complex data structures which represent stereotyped objects, events, or situations. A frame can be thought of as a complex node in a network that is made up of a series of slots. A special slot contains the name of the object, and the other slots contain common attributes and constraints.

For example, a general frame for an automobile could be represented as follows [4]:

Generic AUTOMOBILE Frame
Specialization-Of: VEHICLE
Generalization-Of:
STATION-WAGON,COUPES,SEDANS,HATCHBACKS
Manufacturer:
Range: (FORD,MAZDA,BMW,SAAB,HONDA)
Default: MAZDA
Country-Of-Manufacturer:
Range: (USA,JAPAN,GERMANY,SWEDEN)
Default: JAPAN
Model:
Range: ()
Color:
Range: (BLACK,WHITE,RED)
If-Needed:
(Examine-Title or Consult-Dealer or Look-at-Automobile)
Reliability:
Range: (HIGH,MEDIUM,LOW)

l/100 km:
Range:(1-20)
Year:
Range: (1940-1987)
If-Changed:
(ERROR: Value cannot be modified)
Owner:
Range:
Person-Name
If-Added:
(Apply-For-Title and Obtain-Tag and Pay-Sales Tax)

The slots in this frame identify the manufacturer, country of manufacture, color, model, reliability, l/100 km, year and owner. A number of options can be associated with each slot. These options include a range of possible values, a default value, if-needed (a method for determining the actual value), if-added (the actions to take when a value is given to the slot) and if-changed (the actions to take if the value changes).

Specific automobiles are identified using this generic frame. For example a Honda Civic might be described as follows:

JOE-SMITH'S-AUTOMOBILE Frame
Specialization-Of: HATCHBACK
Manufacturer: HONDA
Country-Of-Manufacture: JAPAN
Model: Civic
Color: Red
Reliability: LOW
l/100 km: 8
Year: 1976
Owner: JOE SMITH
Doors: 2

Note that this frame is a Specialization-Of a HATCHBACK and that the HATCHBACK frame has added a slot called Doors to this structure.

Frame-based systems have gained increasing acceptance since they allow the packaging of declarative knowledge (the basic structure of the frame) with procedural knowledge (the if-needed, if-added and if-changed facets of the slots). The ability to deal with things like exceptions and defaults is something that is very difficult to handle in standard logic. However, many "real-world" situations will not adhere to a generic frame structure. This increases the complexity of the system because each situation has unique features that must be represented.

1.4.1.3 Predicate Logic

Prolog (PROgramming in LOGic) is an example of a high level language that is based on predicate logic. It offers a built-in mechanism for interpreting rules and inferring facts that are not explicitly stated. For example, given the rule "All men are mortal" and the fact "Socrates is a man" it can be deduce that "Socrates is mortal".

Predicate logic requires that relationships be presented as arguments. These arguments consist of a relation which affects one or more objects. Arguments can represent facts or can be combined to produce complex statements. In Prolog, the previous example would be written in the following way [19]:

mortal(X):- man(X)
man(Socrates)

The first statement is a rule and the second statement is a fact. Man and mortal are relations that describe the object (which in this case is a man called Socrates).

While more powerful than production rule interpreters, this representation scheme is costly in terms of memory and CPU time. It is also difficult to control the program flow and manage user interaction.

1.4.2 The Inference Engine

The inference engine is the expert systems reasoning mechanism. It is comprised of non-domain specific rules that interpret the rules in the knowledge base. Currently there are two principle mechanisms used in inference engines - forward chaining and backward chaining.

The forward chaining, or data driven, inference engine attempts to reason forward from a set of facts to an appropriate solution. This type of inference is suited for problems that can produce an infinite number of solutions. Machine configuration, data analysis and design are examples of problems that exhibit this characteristic.

Backward chaining, or goal driven inference starts with a hypothetical conclusion and works backward to seek supporting evidence. MYCIN, an expert system used to diagnose blood infections and prescribe the appropriate treatments, is an example of a backward chaining production system.

It is important to point out the difference between forward and backward chaining, and forward and backward reasoning. Chaining describes the way in which the rules are activated. Reasoning describes the way in which the program as a whole is organized. It is possible to implement a backward reasoning strategy using forward chaining. R1, a program that configures VAX machines, does just this. At a certain level of abstraction, the programs main goal is to configure a system, which can be decomposed into subgoals, such as configuring the Central Processing Unit, and so on. Reasoning back from a main goal via subgoals is referred to as a top-down or backward reasoning strategy. However, during run-time the program actually works through the production rules bottom-up, (i.e., starts with a set of components and tries to achieve a configuration that satisfies each subgoal).

While the inference engine is usually "hard-coded" into the interpreter (and very difficult to change), it is possible to control the reasoning strategy by implementing meta-rules at the program level. Meta-rules differ from ordinary rules in that they direct the reasoning rather than actually perform the reasoning. Meta-rules can either be domain-specific or domain-free.

1.4.3 The Explanation Facility

The ability of an expert system to show the user its "reasoning" is an important feature. The simplest form of an explanation is a trace of the rules and facts that were used to reach the conclusion. Most systems offer a textual interpretation of this information. More sophisticated systems provide "how" and "why" explanations.

When a system is asked "how" did it reach a conclusion, a trace of the logical inference chain followed to achieve the conclusion is provided. "Why" queries are helpful when the user wants clarification as to why a specific question is being asked. In this case, the rule may be revealed to the user for him to examine, or a paraphrasing of the rule's intent may be displayed. These mechanisms depend on the programming environment and the degree of program sophistication.

The ability to explain a line of reasoning may not be necessary in some expert systems. Rigorous scientific systems may reach conclusions that are self-explanatory to those using them. On the other hand, more subjective domains may demand an in-depth explanation of the system's reasoning. Detailed explanations become more critical as the penalty of a wrong decision and/or the frequency of unexpected results increases.

1.4.4 The Natural Language Interface

Communication between the user and the expert system is typically done by entering information in a structured format, or selecting the correct response from a computer menu. However, more sophisticated systems are being developed to utilize natural language understanding. An ideal natural language interface would be able to express itself in standard English, and be capable of comprehension at or above the level of human understanding. Today's best natural language programs are capable of understanding grammatically correct sentences. Ambiguities in context and ungrammatical sentence structure have presented many difficulties for researchers in this field.

1.4.5 The Knowledge Acquisition Subsystem

Throughout the development of an expert system, the knowledge base steadily grows. From inception, through prototype, to maturity, new rules and facts are introduced that improve the systems performance. The mechanism used to alter the knowledge base is referred to as the Knowledge Acquisition subsystem.

In most cases, this is merely an interactive editor that the programmer uses to change or add rules to the existing knowledge base. More sophisticated programs like Meta-DENDRAL [2] interact with an expert to devise and test new rules. At it's most complex, this facility is capable of discovering new concepts and relationships [11]. Programs with these capabilities rely heavily on the principles of "machine learning", another subfield of AI.

1.4.6 Working Memory

This is the area of memory that retains a description of the problem that is being examined. The information the user supplies, plus all that is inferred from the knowledge base is maintained in working memory for the duration of the consultation.

The interpreter uses the data stored in working memory to activate new rules which infer new facts that are subsequently added to the working memory. It is this iterative process that enables the system to produce an answer to the problem.

1.5 Accident Reconstruction

An accident investigation begins by collecting information at the scene of an accident. The success of an investigation depends largely on the thoroughness of this initial task.

Accident reconstruction is one of the final stages of an accident investigation. For most accidents, a formal reconstruction is not required; but for some, a full scale investigation is the only way to determine what caused the accident. The reason for reconstructing an accident can stem from a civil suit, research of injury mechanisms and safety devices, or a government inquiry into safety regulations. Depending upon the query, the reconstructionist may be required to estimate speeds, prove that traffic laws were violated or determine how injuries were sustained.

1.6 How Expert Systems Can Compliment Accident Reconstruction

Reconstructing a motor vehicle accident is an exercise that requires a wide range of expertise. If the purpose of the investigation is to determine injury mechanisms, then engineers and doctors with specialized training in the field of biomechanics are called upon. If a vehicle malfunction contributed to or was responsible for the accident, then mechanics trained in defect investigation are required. If the vehicle's behavior during the collision (speed, direction etc.) must be determined, then engineers with an understanding of collision dynamics are needed. Each of these experts possess extensive knowledge related to their specialization. To immediately bring together all of this expertise, while beneficial, is virtually impossible for all but the most serious of motor vehicle accidents.

The problem solving expertise of these professionals is largely conceptual, although certain aspects of accident reconstruction is quantitative. As mentioned earlier, expert systems lend themselves nicely to conceptual-type problems. However, the difficulty in applying an expert system approach to this problem is accentuated because the field of accident reconstruction is so complex. The problem has to broken down into sub-domains. Within these sub-domains, further stratification may be necessary to reduce the problem to a manageable level.

For this reason, the area investigated by this paper is limited to collision dynamics. Currently, there are standard procedures coupled with many commercial programs which the reconstructionist can use to help determine vehicle speeds, angle of impact, velocity changes etc. An expert system becomes useful at the initial stage of the investigation. The people responsible for this facet of the investigation (usually the police) seldom have any formal training in accident reconstruction. Inevitably, essential information may be overlooked. As a result, the reconstructionist, who is usually called in well after the accident, must rely on secondary information that is often incomplete or ambiguous.

The expert system developed in this paper is for the non-technical user, such as a police officer. By utilizing heuristic and conventional procedures the system is able to point out important factors, help clarify evidence, check the input, and provide the user with preliminary speed estimates.

2. LITERATURE REVIEW

Two topics were examined in the review of the literature: expert systems and accident reconstruction. Expert systems are relatively new, but publications describing the fundamentals of existing systems are abundant. The science of reconstructing a motor vehicle accident is a well established discipline. Literature on reconstruction varies from identification and measurement of scene evidence to complex computer simulations of collision dynamics.

2.1 Expert Systems

In the previous chapter the anatomy of an expert system was described. This section examines the function of many existing systems, and some of the tools and environments suitable for developing an expert system.

2.1.1 Functions of Expert Systems

Although it may be more appropriate to categorize expert systems according to their complexity and problem structure [9], there is a tendency to classify systems according to their function. Hayes-Roth et al. [8] suggest the following functional categories: interpretation, prediction, diagnosis, design, planning, monitoring, debugging, repair, instruction and control.

An interpretation system is basically a data interpreter. It attempts to assign symbolic meaning to observed data. DENDRAL [12], conceived at Stanford University in 1965, is one of the first interpretation systems. DENDRAL is able to determine the molecular structure of an unknown organic compound by interpreting data obtained from a mass spectrometer. HEARSAY- II [6], a speech-understanding system, and PROSPECTOR [5], used by geologists to help identify mineral deposits, are other classic examples of interpretation systems.

Prediction systems generate possible outcomes for given situations. These systems typically employ a parametric model that helps form the basis of a prediction. Weather forecasting, demographic predictions, traffic prediction and military forecasting fit into this category.

Diagnostic systems determine the cause of a problem from a set of observable symptoms. MYCIN [18], another Stanford University product, was developed to help physicians diagnose and treat blood infections. Shortly thereafter PUFF [7], a system for detecting pulmonary disease, was implemented. ONCOCIN, a cancer diagnoser, is another medical aid. Electronics, mechanics and computer software are other disciplines that have given rise to diagnostic expert systems.

Design systems configure objects that must adhere to a set of constraints. Circuit layout, building design and budgeting are examples of this duty. R1 [13] is a classic design system. It was built for Digital Equipment Corporation to configure VAX computer systems, and is one of the most successful commercial expert systems to date.

Planning systems solve problems by assigning functions to objects within the system. Robotics, project planning, communications and military planning are examples of problems in this category. ABSTRIPS [16] and NOAH [17] were among the first planning programs used in robotics. ISIS, developed at Carnegie-Mellon University, is used by Westinghouse to manage and schedule projects in their factories.

Monitoring systems track a system's behavior and identify points of weakness. Several computer-aided monitoring systems exist for nuclear power plants, air traffic control, and medical tasks. None of these expert systems have advanced beyond the laboratory.

Debugging and repair are two problems that expert systems are beginning to address. Computer-aided debugging systems exist for computer programming in the form of intelligent knowledge bases and text editors.

Instruction systems incorporate diagnosis and debugging subsystems which address the student's knowledge in a specific area. These systems begin by constructing a hypothetical description of the students knowledge. The student's weaknesses are then diagnosed and a specialized interactive tutorial is formulated to convey the remedial knowledge to the student. SOPHIE [1], an electronics laboratory instructor, was an early instructive expert system.

The final category is control systems. A control system supervises the overall behavior of a process. By continually monitoring the process, anticipated problems can be identified and remedial plans can be formulated to ensure successful operation. Problems addressed by control systems include air traffic control, business management and mission control.

2.1.2 Implementation Tools

Once a problem has been formalized, suitable development tools or language environments must be selected. In general, the tool/language spectrum can be divided into four categories:

(i) expert system shells (VP-Expert, M.1, Rulemaster)
(ii) special purpose languages (Prolog, OPS5)
(iii) general purpose languages (LISP)
(iv) mixed programming environments (CENTAUR)

Shells have many desirable features. These tools have built-in inference engines for interpreting rules, user interfaces, explanation facilities, knowledge base editors and debuggers, and built-in representation schemes for uncertainty. This eliminates much of the programming task, allowing the developer to concentrate exclusively on the knowledge base. However, shells can be restrictive in several ways. The inference engine's control structure may not match the experts way of solving the problem. The knowledge representation scheme (i.e., rule language) may not have sufficient expressive power. The explanation facilities may be inadequate. Nevertheless, given the number of commercially available shells and the different types of control structures and representation schemes they offer, it is likely that one will be suitable.

Special purpose languages like PROLOG or OPS5 were designed specifically for knowledge engineering. These languages decide how logic is structured and how the program will work. For example, PROLOG is based on a mathematical system known as predicate logic and utilizes a backward chaining mechanism for inferring rules and facts. While these languages provide more flexibility in terms of control structure and representation, the process of applying them is more difficult than with shells, because the user interface and explanation facility must be designed in addition to the knowledge base.

General purpose languages and mixed programming environments are used by experienced AI programmers to help them deal with complex domains. LISP is one of the oldest and most popular programming languages in AI. It is a symbolic manipulation language that is very flexible and can be adapted to a great number of problems. LISP programs are usually implemented on a mainframe or LISP machine which provide enormous amounts of memory and do not inhibit the development of large applications. Mixed programming environments incorporate software modules that allow the programmer to mix programming control structures. For example, CENTAUR mixes rule and frame-based formalisms.

Another consideration that may affect the selection of a tool or language is the hardware and resources available for the project. Many inexpensive shells and language compiler/interpreters are available for personal computers. With the increasing memory capacity of these machines, smaller applications or prototype systems can be designed with these tools. For large, sophisticated applications the advantages of a mainframe and a high-level language cannot be surpassed.

2.2 Accident Reconstruction

The reconstruction of a motor vehicle accident can require the services of people with varied backgrounds. Police officers, doctors, engineers, vehicle mechanics and eye-witnesses all assist in an accident investigation.

The role of the engineer is usually confined the analysis of vehicle behaviour prior to, during, and after a collision through the interpretation of on-scene evidence. Several textbooks and papers have been written that thoroughly discuss the identification and analysis of scene evidence. Additionally, computer modelling is currently at the stage where reliable speed estimates can be obtained for well documented accidents. The computer's ability to handle large amounts of data and perform calculations very quickly has allowed more complete and accurate analysis than hand calculations would permit. The following sections describe some of the accident investigation textbooks as well as the evolution of computer simulation in the field of accident reconstruction.

2.2.1 Accident Investigation Manuals

There are many textbooks available for accident reconstruction [23,26,34,49]. These manuals describe the basic physics that govern vehicle collisions and provide guidelines for the application of certain reconstruction techniques.

Papers by Emori [28,29] and McHenry et al. [39] look at more complex analytical approaches for reconstructing an accident. These techniques are widely used by experts in the field and have subsequently been incorporated into computer programs for accident reconstruction.

2.2.2 Computer Simulation

Computer models to simulate vehicle collisions were initially introduced by McHenry (SMAC - Simulation Model of Automobile Collisions) [39]. This program utilized a "forward calculation" approach to determine vehicle speeds. Initial pre-crash velocities were assumed and run through a 2- dimensional vehicle mathematical model. An iterative routine (START) was added at a later date to generate pre-crash conditions in such a way that the final vehicle positions, the vehicle damage, and the post-crash motion of the mathematical model coincided with those of the actual accident.

During the seventies, SMAC was subject to rigorous testing and underwent a number of revisions to improve it's precision [31,40,50]. The result was a computer program which produced velocity estimates within a claimed accuracy of plus or minus 5 percent. However, to produce these results the user had to provide a very detailed description of the accident.

Realizing the need for a system that could accurately reconstruct accidents that were ill-defined, the Calspan Corporation launched a research project that led to the development of CRASH (Calspan Reconstruction of Accident Speeds on the Highway). The CRASH program was designed to accommodate a range of accident evidence, from CDC's (Collision Deformation Classifications) at one extreme, to complete definitions of damage dimensions and vehicle positions at the other. Unlike SMAC, CRASH worked backwards from specified post-impact conditions to solve for pre-impact conditions. Based on the evidence supplied, the program could do either a full scale trajectory analysis to determine pre-impact speeds, or a simple energy-based calculation to determine the speed change (delta-V) the vehicle underwent during the impact phase. In reasonably well documented cases, CRASH was shown to produce results within plus or minus 12 percent accuracy [40]. In the ensuing years CRASH has been subject to many revisions and refinements [41,42,43,44,47].

The advent of mini and micro computers motivated researchers to develop similar models for use on smaller computers. In 1980, Hess [30] developed two programs written in BASIC for the micro-computer. The first, APPLETRAJ estimated a vehicles linear and angular velocity at the start of a skid based on the field data provided. The second, APPLECRASH was designed to accept vehicle crush data from a pair of damaged vehicles and produce an estimate of each vehicle's change in velocity as a result of the collision.

The increasing memory capacity of the micro computer inevitably led to the conversion of the CRASH program for use on the IBM PC and compatible machines. CRASH3PC [37] is one example of this work.

The trend toward computer simulation of vehicle collisions has led to other developments. VTS [22], IMPAC [51], EDCRASH [27], EES-ARM [52], CAAR_1 and 2 [36] are examples of these accident reconstruction programs. The mathematics and physics underlying each model are similar, and each program has its own advantages and disadvantages. All are equally susceptible to the "garbage in, garbage out" rule, and all require good user understanding of the algorithm to properly interpret the results.

3. SYSTEM METHODOLOGY

Existing computer programs for accident reconstruction contain an inherent weakness; the results they provide can only be as good as the information they process. Incomplete data is something the reconstructionist is often faced with. Usually he is not brought in until civil or criminal action is sought and must rely on evidence gathered by the police. Most attending police officers cannot interpret scene evidence because of their limited training in accident reconstruction. In many cases, information essential for the engineer is overlooked at the scene of the accident. An expert system that could help the police investigator check pertinent data and aid in a preliminary analysis of the accident would be useful.

The goal of the expert system developed in this paper, Knowledge-base Accident Reconstruction (KAR), is to provide a computer program that can guide the user through the reconstruction process. Its primary functions are to check the data collected by the police, suggest methods for solving speeds, and calculate the speeds if possible. By utilizing an expert system approach, heuristics governing the information requirements for each method can be tested and verified before proceeding with the calculations. Explanations and directions on information acquisition can also be incorporated into the system using this strategy.

Because the program is aimed at a non-technical user, the equations of motion programmed into the system are relatively basic and require easily obtainable information. Except for the energy calculations, the complexity of the equations do not exceed those presented in standard reconstruction manuals for police; thereby making the underlying physics accessible to every user.

The following sections describe the methodology and mathematical techniques that KAR uses to determine vehicle speeds.

3.1 Reconstruction Methodology

Speed estimation is usually an important part of the overall reconstruction process. In many cases, this may be the only question under investigation. Skid analysis, yaw mark analysis, vault, conservation of momentum and crush energy are methods available to the engineer for the purpose of determining speed. The method or methods that are selected depend on the evidence that is available (i.e. skid marks, vehicle damage, debris, etc.). At the location of an accident, evidence must be gathered from three primary sources: the scene, the vehicles and witnesses. Each of these categories can be broken down further to represent the different stages of the accident: Pre-Impact, Impact and Post-Impact. Figure 2 is a graphical representation of this breakdown.

The Pre-Impact phase includes all of the events leading up to the collision. The Impact phase involves the collision where some point on both vehicles attain a common velocity. The Post-Impact phase usually involves sliding or rolling to a final rest position. The three stages are represented in Figure 3.

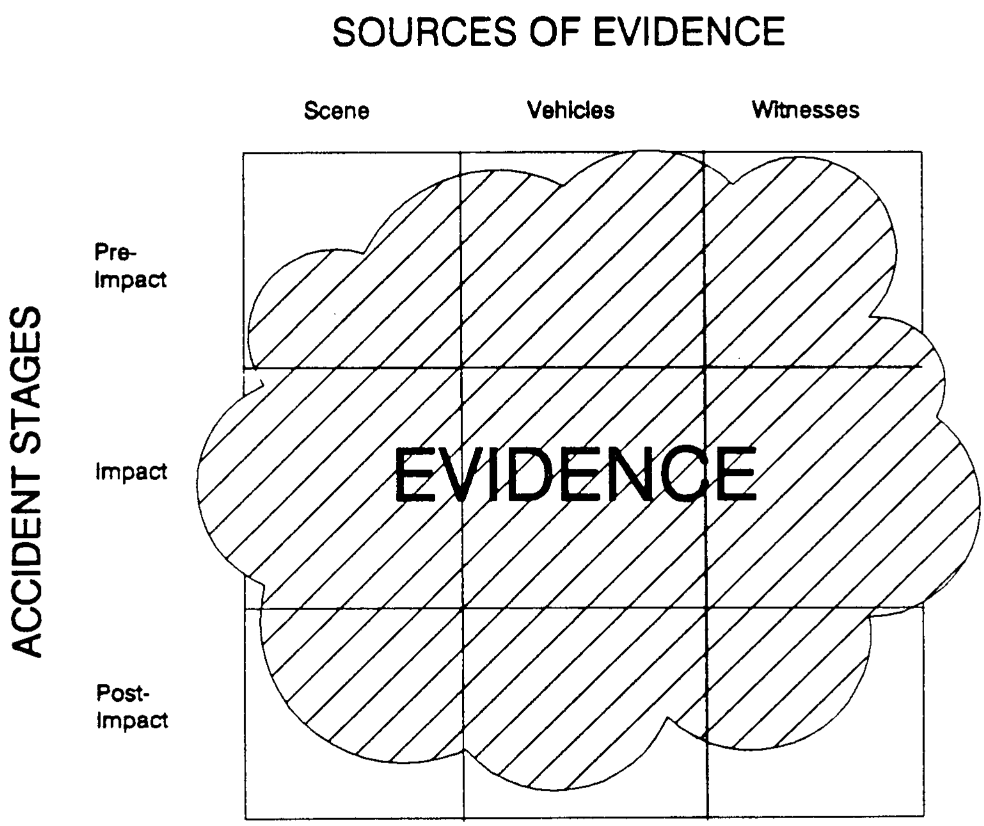


Figure 2: Categorization of Evidence

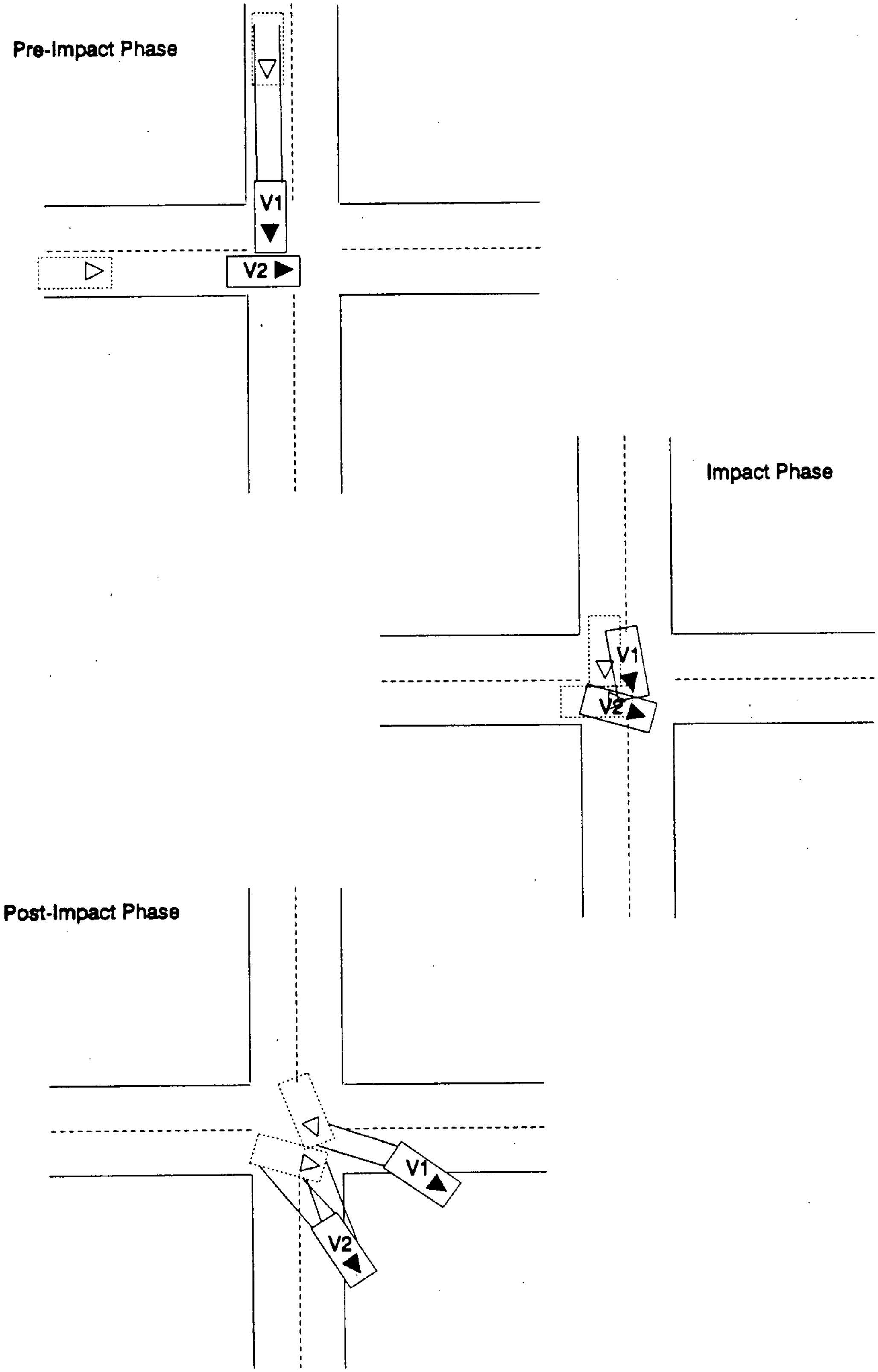


Figure 3: Accident Stages

Arranging evidence into the categories defined in Figure 2 helps the investigator establish the sequence of events and thus provides a clear picture of what took place. This is a mental process and is something the trained investigator can do quite quickly if the accident is straightforward. For more complex accidents where information is missing or ambiguous, grouping the evidence like this and going over possible scenarios is a common practice.

After the evidence has been gathered and the investigator has a good understanding of the accident, methods to determine individual vehicle speeds can be selected. If for example, Vehicle 1 left 5 meters of skid prior to impacting with Vehicle 2 and the drag factor (average coefficient of friction between the tires and the road surface) can be determined, a skid analysis would give the change in velocity over this distance. However, since Vehicle 1 did not skid to a stop (i.e. all of the vehicles energy was not dissipated in skid; some went into deforming vehicle parts, heat, etc.), using this method on its own would underestimate the vehicles initial speed. If the speed at impact, or end of skid, is not zero, some other method such as momentum or energy must be used to determine the impact speed. By combining the speed at impact with the change in speed as a result of skidding, Vehicle 1's initial speed (prior to braking) can be calculated. The situation described above becomes awkward because of the sequence in which the speed is calculated. For hand calculations, the normal reconstruction procedure is to work backwards in time from post-impact conditions to solve for the pre-impact conditions.

3.2 Reconstruction Techniques

The five speed determination techniques programmed into KAR are: skid analysis, yaw analysis, vault, conservation of linear momentum and crush energy. The following sections describe the fundamental physics behind each method, and some of the strategies used to help the investigator identify and correctly interpret the physical evidence.

3.2.1 Skid Analysis

When brakes are applied, the mechanics of the braking system slow the rotation of the wheel. At this point the tire begins to slip relative to the road surface. If the brake is firmly applied, the wheel locks and the tire ceases to rotate leaving a skid mark in most cases.

During the skid, the vehicle's kinetic energy is dissipated through friction on the road surface. Applying the conservation of energy, the speed of the vehicle at the beginning of the skid can be determined by the following equation (see Appendix A for derivation of equation):

$$v_i = \sqrt{2fgd + v_f^2}$$

where

v_i	=	minimum initial speed (m/s)
v_f	=	estimated speed at end of skid mark (m/s)
f	=	coefficient of friction or drag factor
g	=	acceleration of gravity (m/s^2)
d	=	skid distance (m)

Reconstruction manuals refer to this equation as the minimum speed formula because it does not reflect the amount of speed lost before skidding commences (i.e. braking that did not produce skidmarks) and, in the event that the speed lost during the collision cannot be determined (using other methods), v_f can be set to zero or the speed after impact, to give a minimum bound for the initial speed of the vehicle. Variations of this formula were used in the program to account for vehicle braking efficiency, and skid marks over different surfaces. These formulas are listed in Appendix (A).

The value of the drag factor (f) can be determined by conducting skid tests, measuring the value with a drag sled, or using a table of friction coefficients. Test skids provide a good indicator if conditions at the time of the accident can be duplicated. In many instances, however, tests of this nature may be too dangerous or too difficult to conduct. A drag sled measurement is an alternative and is easy to perform. For preliminary analysis a table value is acceptable, but must be adjusted for slope (i.e., compensation for the vehicles weight component parallel to the grade). For convenience, Table 1 was converted to a series of rules and integrated into the system.

Because the program was not designed to handle uncertainty, the ranges presented in Table 1 were converted to average values for use in the rules. Each friction coefficient is represented in the following manner:

```
IF
        Drag_Value = UNKNOWN AND
        Road_Mix = Asphalt_or_Tar AND
        Road_ConditionPC = New AND
        Weather = Dry AND
        Speed_Range = Greater_50kmh
THEN
        Drag_Factor = 0.83
```

For a vehicle travelling in excess of 50 km/h on a dry, new, asphalt roadway, a drag factor of 0.83 would be assigned if the actual value was unknown. If necessary, a slope correction routine adjusts this factor.

Table 1: Coefficients of Friction of Various Roadway Surfaces (Source: Baker [23])

Description of Road Surface	DRY Less than 30 mph		DRY More than 30 mph		WET Less than 30 mph		WET More than 30 mph	
	From	To	From	To	From	To	From	To
PORTLAND CEMENT								
New, Sharp	0.80	1.20	0.70	1.00	0.50	0.80	0.40	0.75
Travelled	0.60	0.80	0.60	0.75	0.45	0.70	0.45	0.65
Polished	0.55	0.75	0.50	0.65	0.45	0.65	0.45	0.60
ASPHALT OR TAR								
New, Sharp	0.80	1.20	0.65	1.00	0.50	0.80	0.45	0.75
Travelled	0.60	0.80	0.55	0.70	0.45	0.70	0.40	0.65
Polished	0.55	0.75	0.45	0.65	0.45	0.65	0.40	0.60
Excess Tar	0.50	0.60	0.35	0.60	0.30	0.60	0.25	0.55
GRAVEL								
Packed, Oiled	0.55	0.85	0.50	0.80	0.40	0.80	0.40	0.60
Loose	0.40	0.70	0.40	0.70	0.45	0.75	0.45	0.75
CINDERS								
Packed	0.50	0.70	0.50	0.70	0.65	0.75	0.65	0.75
ROCK								
Crushed	0.55	0.75	0.55	0.75	0.55	0.75	0.55	0.75
ICE								
Smooth	0.10	0.25	0.07	0.20	0.05	0.10	0.05	0.10
SNOW								
Packed	0.30	0.55	0.35	0.55	0.30	0.60	0.30	0.60
Loose	0.10	0.25	0.10	0.20	0.30	0.60	0.30	0.60

When variable skid lengths are reported for each tire, confusion may arise as to which value should be used in the minimum speed formula. If the brakes are functioning properly on all wheels, the vehicle will track relatively straight. In this case, the longest skid mark will provide the best speed estimate. When it is apparent that braking was not applied uniformly (i.e. tracking to one side or vehicle rotation), an average value should be used. Figure 4 shows the line of reasoning the program follows in order to determine the "correct" distance to be used in the skid equation.

If there is less than 100 percent braking efficiency, the drag factor is multiplied by the braking efficiency ratio (i.e., 0.75 would indicate 3 out of 4 tires braking) to compensate for the rolling tire(s). Therefore, the braking efficiency does not effect the skid distance routine outlined in Figure 4.

Two other methods that utilize the same principles as skid are roll and overturn. If a vehicle rolls for a significant distance after impact the energy dissipated due to rolling resistance can affect the initial speed calculation. By substituting a drag factor of 0.1 and the roll distance into the skid formula, the vehicle's change in speed can be calculated over this distance.

When a vehicle overturns and slids to a stop, the drag factor between the metal and the road surface can be used in the skid formula to determine the vehicle's change in speed.

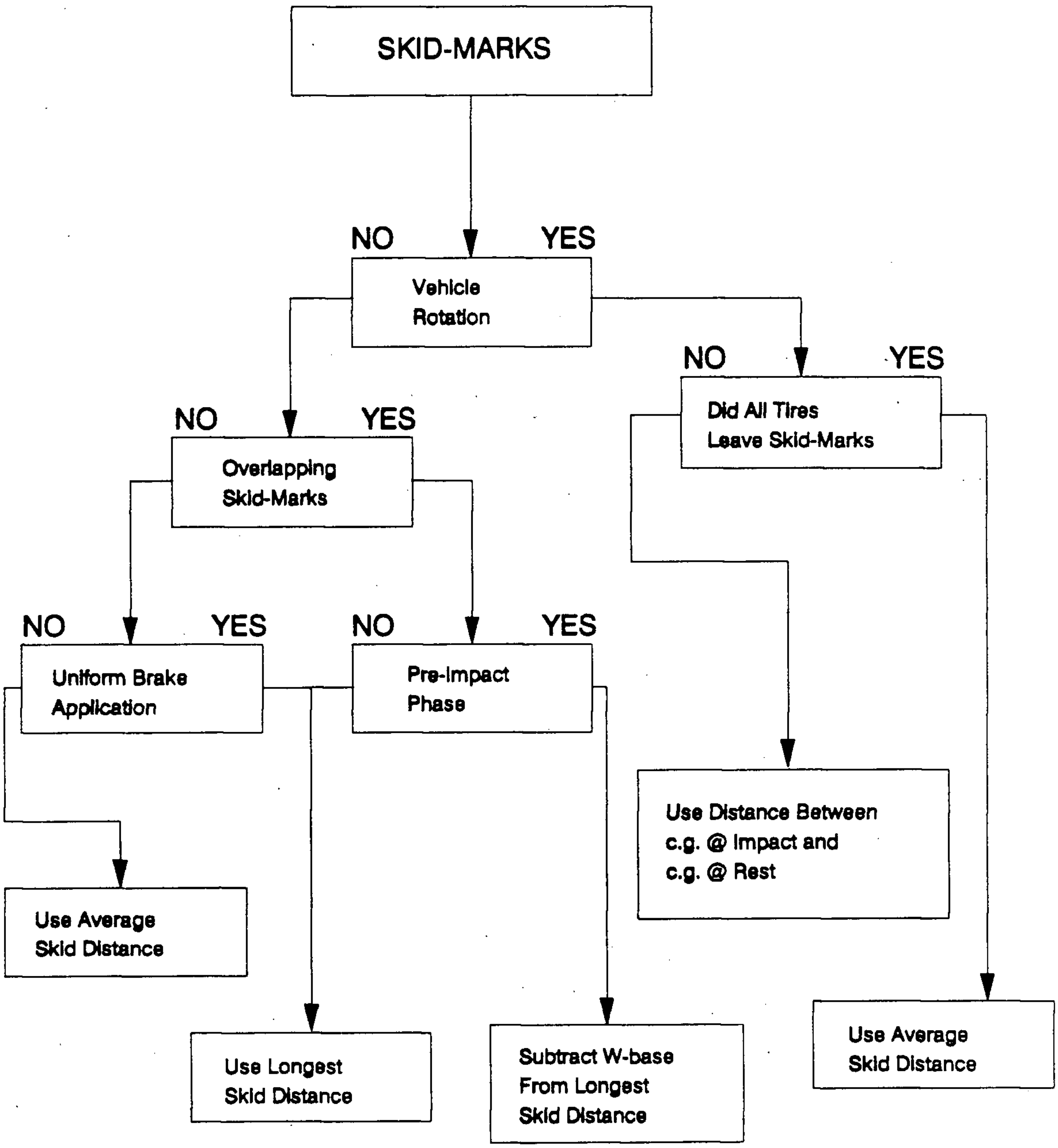


Figure 4: Flow Diagram for Determining Distance to Use in Skid Equation

3.2.2 Speed Calculations From Yaw Marks

When a vehicle negotiates a curve, a balance of centripetal forces and inertia holds the vehicle in the turn. Centripetal forces (frictional forces between the tires and the road) pull the vehicle towards the center of the curve, while the centrifugal force (inertia) works to pull the vehicle in a straight line, or tangent to the curve. For every curve there is a critical speed at which the centrifugal force overcomes the side tire friction, and the vehicle begins to yaw. At this point, the outer edge of the tire leaves a narrow dark mark that appears as a thin line about 5 cms in width and widens to the amount of tire tread in contact with the roadway as the vehicle yaws. Lateral striations in the mark caused by the side of the tire tread are often visible.

The critical curve speed can be calculated using the following equation (see Appendix A for derivation):

$$v_c = \sqrt{Rg\,\frac{(f+e)}{(1-fe)}}$$

where

v_c	=	critical curve speed (m/s)
R	=	radius of the yaw mark (m)
g	=	acceleration of gravity (m/s^2)
f	=	coefficient of friction or drag factor
e	=	superelevation (%/100)

By using the radius of curvature of the yaw mark, the speed of the vehicle as it goes into yaw can be determined.

To ensure a precise speed estimate, the first one-third of the yaw mark should be used to calculate the radius of curvature (R). The drag factor can be determined using the procedures mentioned in the previous section.

3.2.3 Vault

If a vehicle becomes airborne during some phase of the accident, it is possible to calculate the vehicles speed at the point of takeoff using kinematics. Given the horizontal distance the vehicle vaulted, the vertical fall and the takeoff angle, the speed can be calculated as follows:

$$v = \sqrt{\frac{gd^2}{2(h+de)}}$$

where

v	=	speed at takeoff (m/s)
d	=	horizontal distance (m)
h	=	vertical distance (m)
e	=	takeoff angle (%/100)

The point where the vehicle leaves the ground can usually be identified by the displacement of earth or other surface materials. The takeoff angle is measured at this exact location using a slope measurement devise or transit. Since the final resting position of the vehicle is rarely the point where it initially struck the ground, indentations in the earth or scars on hard surfaces can be used to identify the first landing point. When using this equation it is important that all measurements be taken as closely as possible from the center of mass of the vehicle at takeoff to the center of mass at the point where it first strikes the ground.

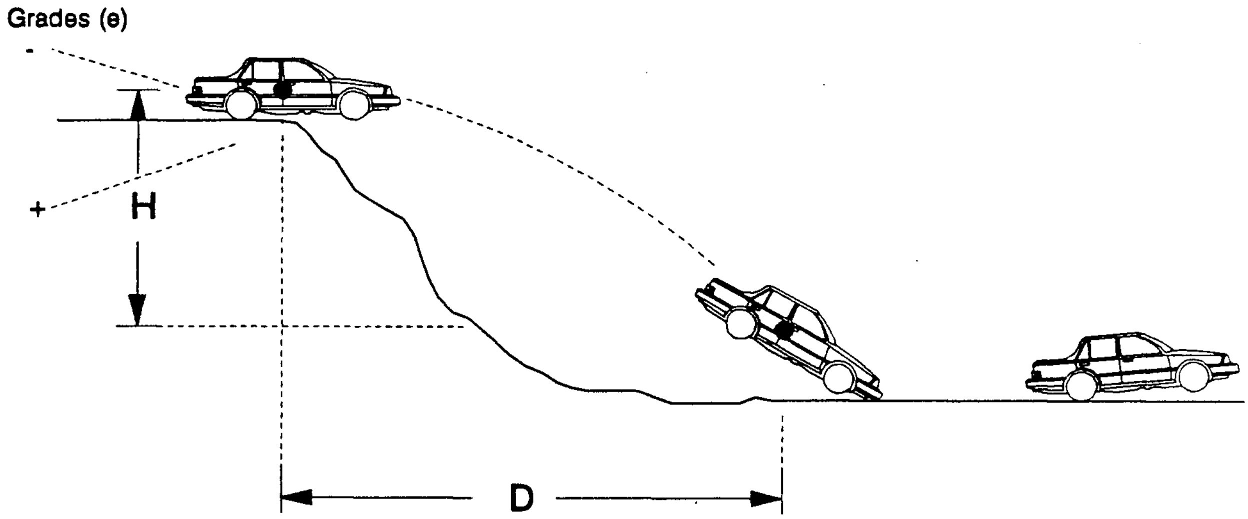


Figure 5: Vault Parameters

There are other situations where the vault formula can provide useful speed estimates. For example, if vehicle debris (windows, lights, hood ornaments etc.) are dislodged during impact, the approximate speed of the vehicle may be estimated given the initial height and distance travelled by the debris. Because the force required to break away the object is not accounted for, speed estimates can be considered conservative.

Like the yaw equation, the vault formula gives the vehicles speed at the beginning of the incident, not the change in speed over the duration of the event. Hence, when yaw or vault are used to calculate speed, any subsequent events (i.e., skid, impact, roll etc.) are ignored. Any events that occur prior to yaw or vault are analyzed and combined to provide an initial speed estimate.

3.2.4 Conservation of Linear Momentum

During the impact phase of an accident each vehicle experiences a change in velocity, however, momentum is conserved. Therefore, if the momentum after impact can be established, the momentum prior to impact can be estimated. Simply stated, the vector sum of the colliding vehicles is equal to the vector sum of the same vehicles after collision. Representing this in x-y coordinates the following equations are obtained:

$$m_1 v_{1xi} + m_2 v_{2xi} = m_1 v_{1xf} + m_2 v_{2xf}$$

$$m_1 v_{1yi} + m_2 v_{2yi} = m_1 v_{1yf} + m_2 v_{2yf}$$

where

m_1	=	mass of vehicle 1 (kg)
m_2	=	mass of vehicle 2 (kg)
v_{nxi}	=	velocity of vehicle n in the x direction prior to impact (m/s)
v_{nyi}	=	velocity of vehicle n in the y direction after impact (m/s)

The logical segment of the program would not permit a momentum calculation if the vehicles approach and departure paths, weights and post-impact velocities were indeterminate. Post-impact velocities calculated by other methods are automatically passed to this routine. Vehicle weights and headings must be supplied by the user at this stage.

The NHTSA (National Highway Traffic Safety Administration) trajectory convention is adopted here for representing vehicle headings. This convention is shown in Figure 6.

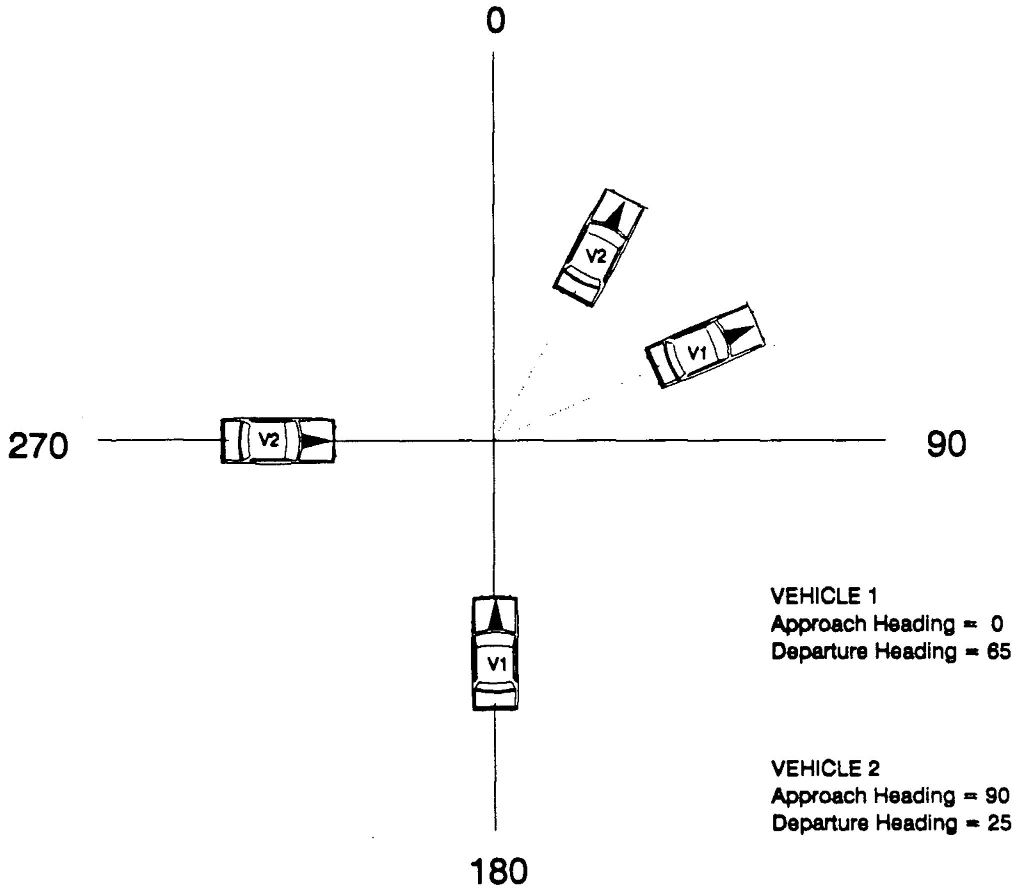


Figure 6: NHTSA Trajectory Convention

To avoid any confusion, vehicle 1's initial heading is set to 0 degrees. All other vehicle headings are input relative to this position. After the headings are input, a simple routine checks if they are plausible. If the departure headings lie outside a specified range prescribed by the approach headings, the user is asked to check the data and re-enter it. If these values fail the test a second time, a warning is issued, but the program proceeds with the calculations. Figure 7 shows the acceptable range of the departure paths as dictated by the approach paths.

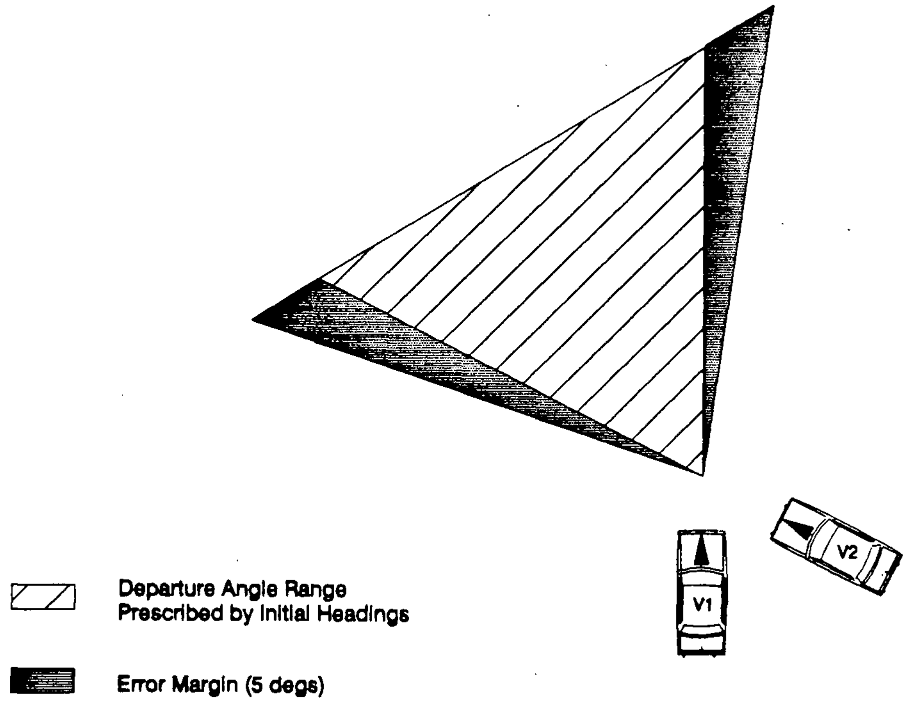


Figure 7: Acceptable Departure Headings

In the event that the approach paths are collinear (i.e. head-on or rear- end accident configurations), an estimate of one of the vehicles pre-impact velocities is necessary in order to calculate the other vehicle's speed.

3.2.5 Crush Energy

An alternative method of solving for impact speeds is to estimate the energy lost during the collision. The energy absorbed by the vehicle during impact is defined by the work done to crush the vehicle to its deformed state. The CRASH program assumes that there is a linear relationship between the force applied during impact and the amount of crush (Figure 8).

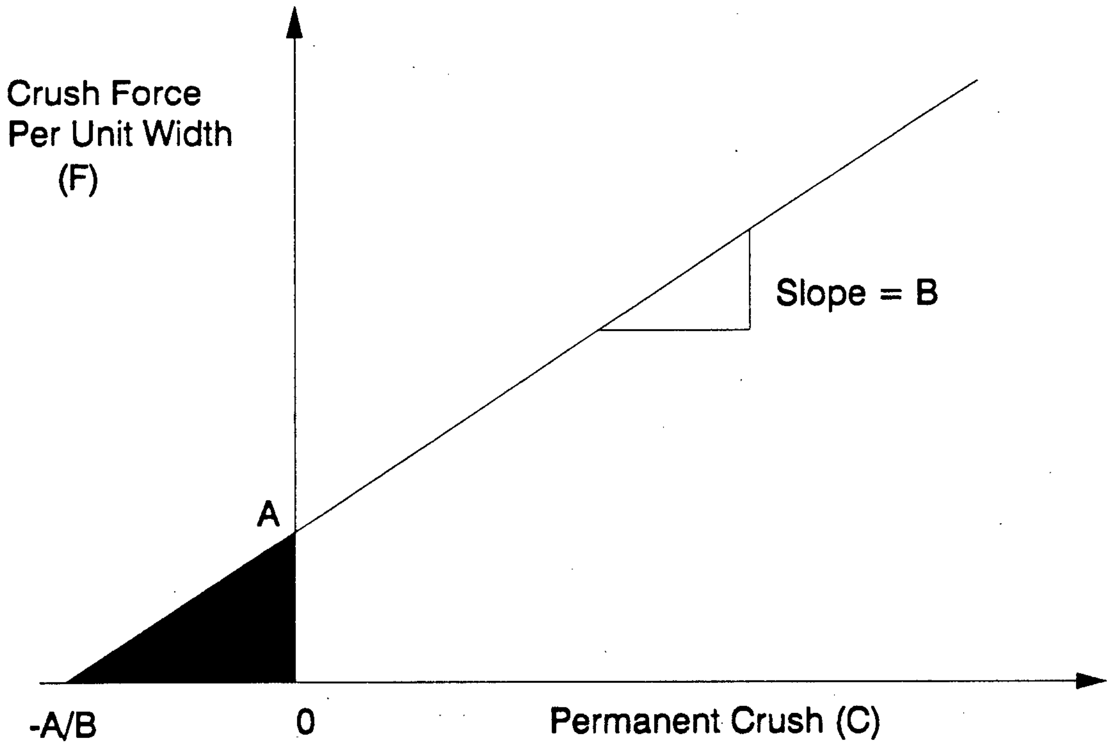


Figure 8: Force vs. Crush

For a force applied normal to the surface, the energy absorbed by the vehicle is the area under this curve or:

$$E = \int (A+Bx)dxdL$$

A numeric integration of this equation for two equally spaced crush dimensions produces the expanded formula:

$$E = L\left(\frac{A}{2}(c_1+c_2)+\frac{B}{6}(c_1{}^2+c_1c_2+c_2{}^2)+\frac{A^2}{2B}\right)$$

where

E	=	energy absorbed by crushed vehicle (N.m)
L	=	width of crush profile (cms)
A,B	=	empirical coefficients obtained from crash test
c_n	=	individual crush measurements (cms) (measured relative to the original surface)

The crush coefficients A and B (Table 2), were converted to a set of rules. If the program is required to calculate the delta-V's, the coefficients are assigned according to the collision surface and the vehicle's size.

Table 2: Vehicle Stiffness Coefficients (Source: McCarthy et. al. [37])

Stiffness Coefficients		Wheelbase (cms)					
		205-240	241-257	258-279	280-297	298-312	>311
Front	A	530.6	455.1	557.0	625.5	571.0	571.0
	B	32.5	29.8	38.8	23.5	25.6	25.6
Rear	A	643.1	687.0	720.4	627.2	521.8	521.8
	B	26.3	28.4	30.4	9.0	48.4	48.4
Side	A	135.3	246.0	304.0	251.3	311.0	311.0
(R,L)	B	25.6	46.4	39.4	34.6	32.5	32.5

Given the value of E, the change in velocity experienced by each vehicle can be determined. By equating the total pre-impact kinetic energy with the total post-impact kinetic energy plus the crush energy absorbed by each vehicle, and then substituting momentum, the following equations can be stated (see Appendix A for derivation):

$$\Delta v_1 = \sqrt{\frac{2\,(E_1+E_2)}{m_1\left(1+m_1/m_2\right)}} \qquad \Delta v_2 = \sqrt{\frac{2\,(E_1+E_2)}{m_2\left(1+m_2/m_1\right)}}$$

where

Δv_n = change in velocity for vehicle n during approach period (m/s)
E_n = energy absorbed by vehicle n (N.m)
m_n = mass of vehicle n (kg)

Hence, the velocity changes can be established directly from the vehicle damage. In order to calculate the initial velocities however, post-impact speeds and the principle direction of the impact force (PDOF) are necessary.

The PDOF can be obtained from the damaged vehicle in the field. Figure 9 illustrates how this angle is defined.

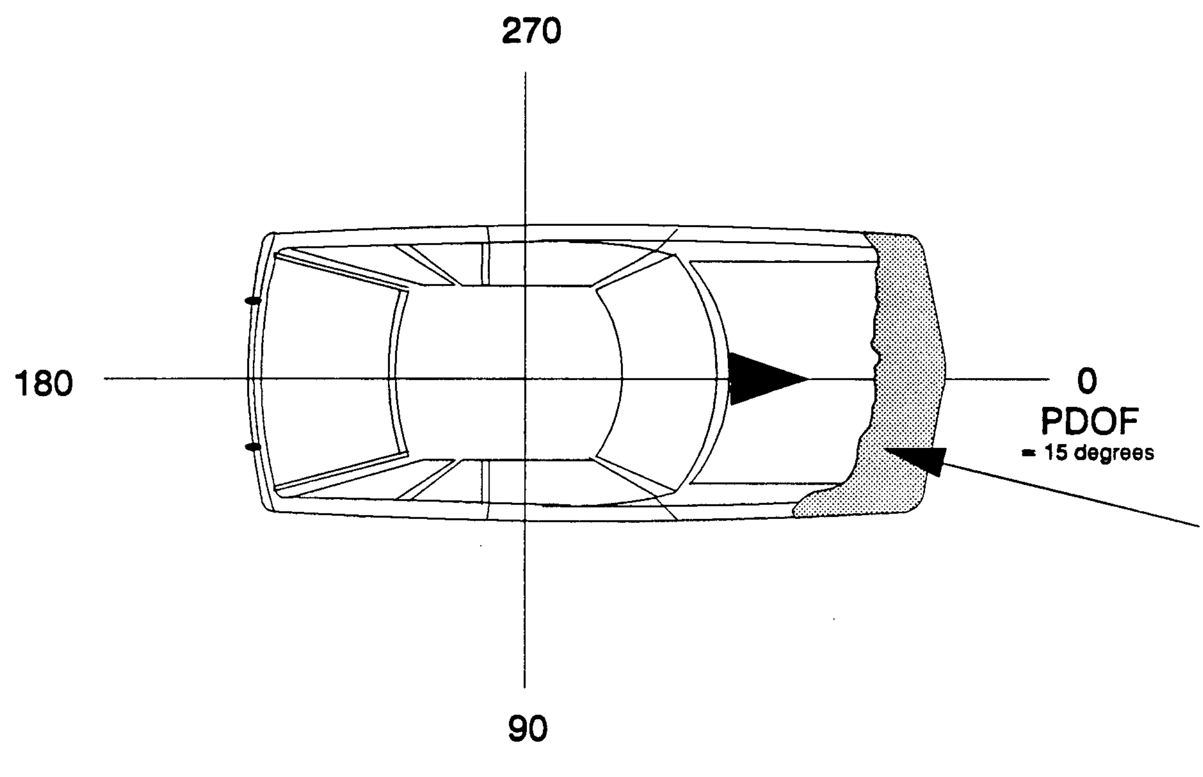


Figure 9: Principle Direction of Force

Since the PDOF is measured with respect to the vehicle's longitudinal axis (initial heading), the vehicle's initial speed can be calculated using a combination of the Cosine Law and the Quadratic Equation (Figure 10).

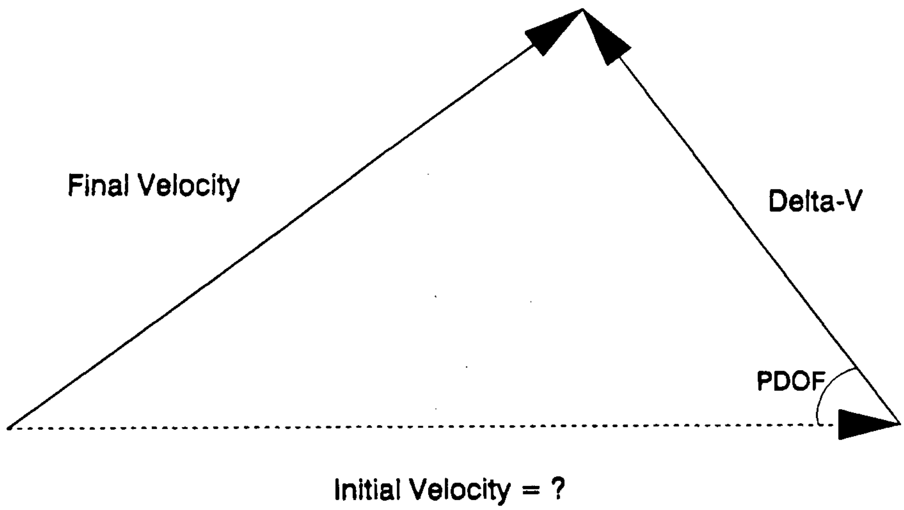


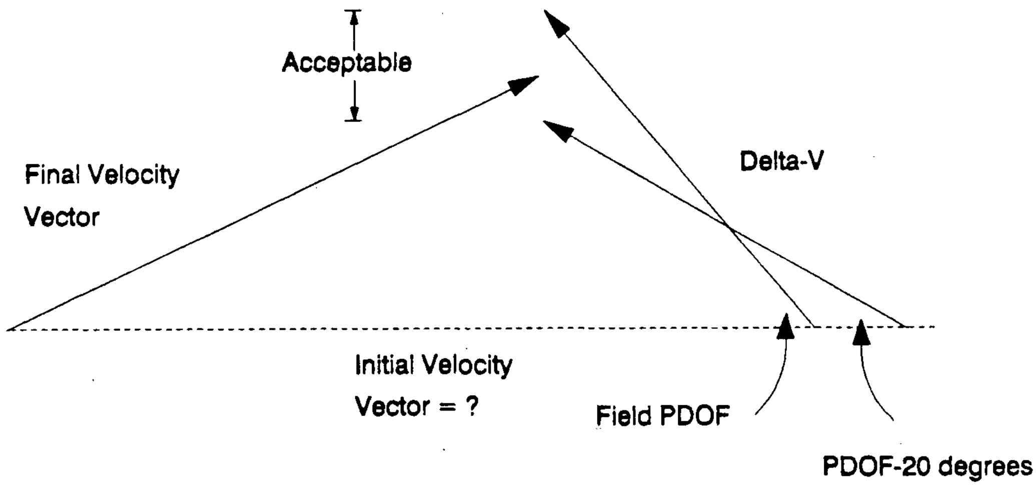
Figure 10: Velocity Vector Diagram

It is important that the PDOF be measured as accurately as possible. Smith and Noga [50] studied the sensitivity of the PDOF and crush measurements on the delta-V. They found that in two-vehicle accidents the estimate of the PDOF contributed almost entirely to the sensitivity of delta-V. Field observations of the PDOF were out by as much as 20 degrees (95% confidence limit). Because this value is subject to considerable error, an optional check procedure was programmed into the system. If the vehicle's approach and departure headings are well established, they can be used to ensure the PDOF is within an acceptable range of error.

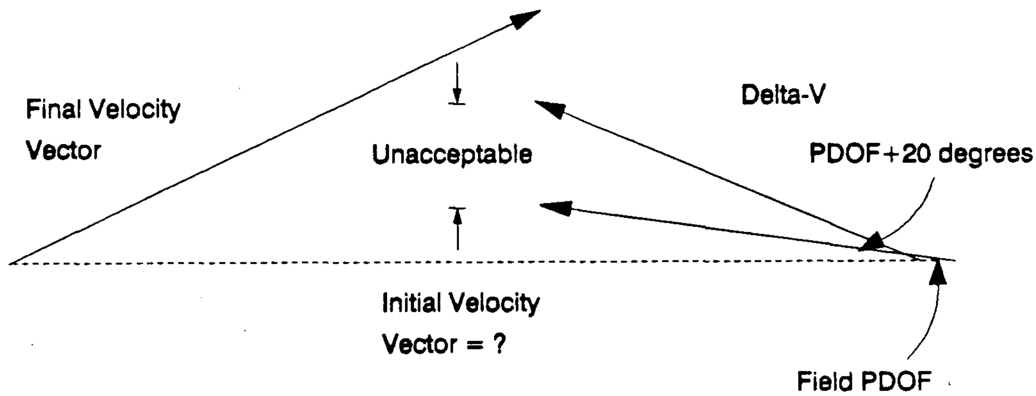
First, the change in velocity (delta-V) is calculated from the crush measurements. This value is rotated ± 20 degrees (the maximum range of error for the PDOF), relative to the field PDOF, to ensure that this vector will close with the final velocity vector (Figure 11).

If the vectors do not close, the initial velocity cannot be calculated because the delta-V is either too small or too large. This would most likely be a result of error in the crush measurements. The user is then warned that the measurements are overestimating or underestimating the value of delta-V.

For non-central and oblique impacts, the calculation of delta-V is dependent on the PDOF and its offset from the vehicle's center of gravity. (At this stage, the energy equations programmed into the system assume a central impact. If the equations were modified to encompass non-central collisions, an iterative calculation would have to be added to this check procedure because of delta-V's dependency on the PDOF.)



* DELTA-V CLOSES WITH FINAL VELOCITY



* DELTA-V DOES NOT CLOSE WITH FINAL VELOCITY

Figure 11: PDOF Check Routine

4. EXPLANATION OF THE SYSTEM

KAR was developed on an IBM PS/2 Model 80 computer using the expert system shell VP-Expert. It can be used on any IBM Compatible machine with a minimum of 640 kilobytes of RAM. The Knowledge base consists of approximately 120 kilobytes of source code and contains over 250 active rules (see Appendices B and C for program listing). The system is presently capable of solving vehicle speeds for accidents involving two automobiles. However, it is possible to access the programs analytical segment directly and analyze a single vehicle accident.

Because VP-Expert cannot access extended RAM, the size of a knowledge base is limited to 100 kilobytes. However, it is possible to link separate knowledge bases together so that programs too large to be contained in memory can be used sequentially. For this reason the program is divided into two logical modules. The first module contains the rules that determine the solution approach, and the second module contains the analytical procedures and decision rules that apply to each technique.

4.1 Development Tool

VP-Expert is a production rule expert system shell. It is equipped with an inference engine for interpreting rules, a user interface, and an explanation facility. KAR was developed by encoding knowledge relating to accident reconstruction into the knowledge base editor in a form the inference engine could interpret (Figure 12).

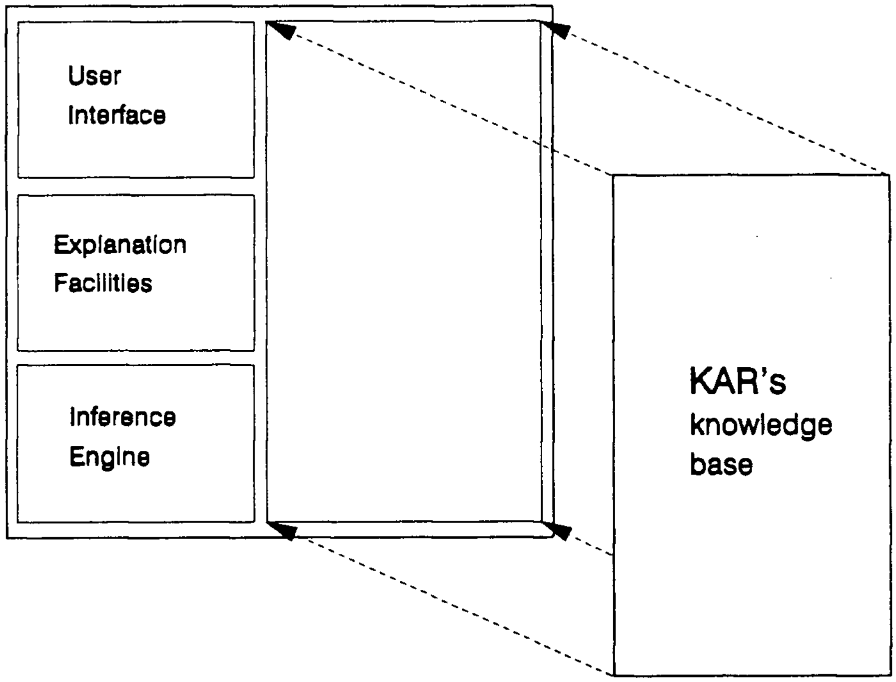


Figure 12: Inserting Knowledge into VP-Expert

The VP-Expert knowledge base consists of 3 basic elements: an ACTIONS block, Rules and Statements. The ACTIONS block directs the program during the consultation. Typically the ACTIONS block will contain at least one FIND clause. The FIND clause identifies the major goals of the consultation and directs the inference engines search of the rules. The primary method of inference employed by VP-Expert is backward chaining. However, it is possible to accomplish a form of forward chaining by using a sequence of FIND clauses within the ACTIONS block to influence the inference engines path through the knowledge base.

Rules are written as if-then structures and are represented in the following manner:

```
RULE <rule_label>
IF
        <condition1> [AND/OR]
        <condition2> [AND/OR]
        [etc...]
THEN
        <conclusion1>
        <conclusion2>
        [etc...]
        <clause1>
        <clause2>
        [etc...]
ELSE
        <conclusion3>
        <conclusion4>
        [etc...]
        <clause3>
        [etc...]
BECAUSE
        <text>;
```

The rule premise can contain up to 10 conditions. Each condition contains a variable, a relational operator and a value (i.e., *Road_Mix = Ice*). Logical operators *AND* and *OR* are used to combine multiple expressions and form complex rule conditions.

The rule conclusion consists of one or more expressions each assigning a specific value to a variable. If the conditions in the rule premise are not satisfied, it is possible to assign alternate conclusions by including an *ELSE* statement at the end of the rule.

Within the rule conclusion it is also possible to include optional clauses. These clauses can be used to display text, open windows, ask questions, or find the value of a particular variable.

The *BECAUSE* keyword is used to provide explanatory text in response to "why" or "how" commands given by the user. If the reason for a condition in the rule premise is unclear the user can issue the "why" command. If the rule contains the *BECAUSE* keyword, this text is displayed. If this keyword is not included in the rule, the rule itself is displayed.

There are two basic types of statements. One group of statements is used to alter the appearance of the screen during the programs runtime. The other set of statements is directly tied to the rules in the knowledge base. The latter, which include the *ASK* and *CHOICE* statements, appear at the end of the knowledge base and are activated when a value cannot be obtained from the knowledge base and must be supplied by the user.

4.2 Selecting the Appropriate Methods of Solution

The first module contains the rules that determine a suitable approach for solving vehicle speeds. This part of the program has been designed to manage any accident configuration involving two vehicles. No numerical data is requested at this stage, only a thorough description of the accident. Based on the qualitative responses of the user, this module assesses the accident and decides which mathematical methods are appropriate. The selected methods are then passed to the second module for analysis.

The ACTIONS block in this module contains only one goal, *FIND Solution*. This clause evaluates 2 rules that control the execution of this module. If the first rule succeeds, the lower level rules are tested to determine the solution approach. If the second rule succeeds, this module is bypassed and the user can select the solution methods directly. This is desirable for experienced users or for accidents that are relatively straight forward.

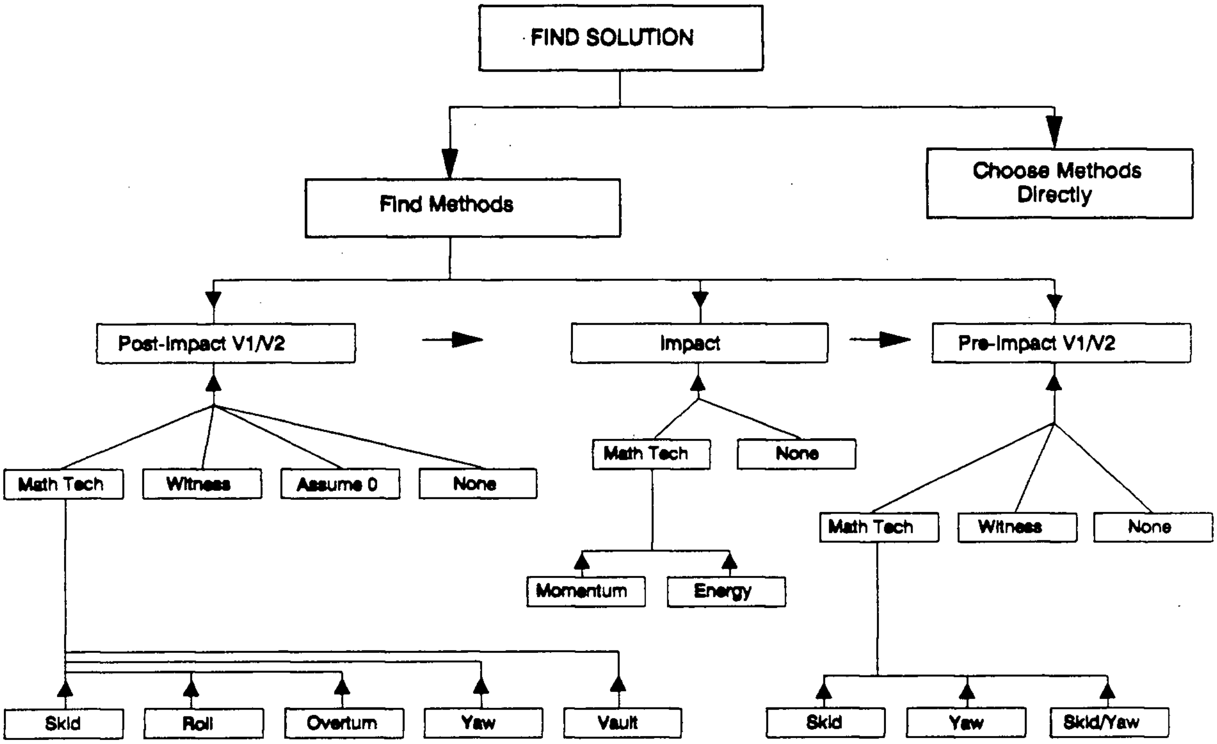


Figure 13: Rule Hierarchy

The first rule contains five major goals in its conclusion. These goals attempt to identify the method(s) of solution for each vehicle during each phase of the accident. Figure 13 demonstrates the rule hierarchy employed by this module. The arrows indicate the flow of the program. Initially, the inference engine is directed to the five major goals (in Figure 13 only 3 boxes are shown because Vehicle 1 and Vehicle 2 are grouped together for the post and pre-impact phases) by including FIND statements in the conclusion of the top-level rules (a form of forward chaining). Each goal is tested in order, starting with the post-impact phase of Vehicle 1, and working through to the pre-impact phase of Vehicle 2. Once a goal has been set, its solution is sought by means of a top-down search (backward chaining). For example, the goal for the post-impact phase of Vehicle 1, *"FIND Method_Postv1"*, evaluates every rule in the knowledge base with *"Method_Postv1"* in its conclusion.

An example of a rule that regulates the use of one of the mathematical techniques is given below. This rule decides whether a yaw analysis can be used to estimate Vehicle 1's post-impact speed.

```
IF
        Post_Actv1 = Yaw AND
        Surf_Postv1 = Yes AND
        Flat_Postv1 = Yes
THEN
        Method_Postv1 = Yaw;
```

Each condition in the rule premise activates a query (in many cases the conditional statements will activate lower level rules) unless it has already been assigned a value. In order for this particular rule to pass the following conditions have to be met: yaw during the post impact phase must be present, all four tires have to be on the same surface for the first one-third of the mark, and the vehicle's weight must be constant for the duration of the mark (i.e., the vehicle could not have been travelling over the crest of a hill). The equation used to determine speed at the onset of yaw is dependent on a constant drag factor and a constant vehicle weight force. The last two conditions prevent the incorrect use of this equation.

Often more than one technique may be required to solve a vehicles speed during a specific phase of the accident. Therefore, each goal can have multiple values assigned to it (i.e. *Method_Postv1 = Skid,Vault*). An important factor that has to be considered when combining post or pre-impact speeds is that skid, roll, and overturn calculations give the vehicle's speed change, while yaw and vault give the vehicle's initial speed. Therefore, if a combination of events occur during the post or pre-impact phases, evaluating any event that follows yaw or vault is unnecessary. To avoid confusion the user is asked to select the appropriate events in the sequence that they occurred. Based on order of events the program determines which methods have to be analyzed.

For the impact phase, it may be possible to calculate speeds using both momentum and energy methods. In this case both methods are sent to the second module for analysis. Based on the accident configuration, the second module will decide which impact speed is the most reliable estimate.

If there is insufficient physical evidence to calculate the speed at a certain stage, other options are available. A witnesses estimate can be used, or if the vehicle travelled only a short distance after impact, the post-impact speed can be set to zero. If at any stage of the accident, no method of estimating speed is available, the goal is set to *"None"*. The second module handles this response by warning the user that this stage was left out of the analysis.

4.3 Organization of the Analytical Module

The second module is designed to accept the methods passed to it from the first module and begin a quantitative analysis of the accident. Logically, it approaches the problem in the same manner as the first module (post-impact through pre-impact). Initially, this module does not possess any detailed information about the accident, only which methods to attempt. First, it must obtain the data from the user and then perform the calculations and appropriate checks.

Like the first module, there are five top level goals corresponding to each stage of the accident for both vehicles. These goals activate the appropriate analytical techniques. The program has seven speed techniques programmed into lower level rules: skid, roll, overturn, yaw, vault, momentum and energy. The basic physics and information requirements for each method have been discussed in Chapter 3. The following sections describe how the program manages the calculations for each stage, and how it deals with the situation when calculations are not possible.

4.3.1 The Post-Impact Phase

The post-impact phase is one of the most variable stages of an accident. For low speed accidents vehicles may stop directly after impact, but accidents where moderate to high speeds are involved can produce a number of complex scenarios. In some cases the vehicles are redirected off the roadway which introduces new surfaces to the analysis, and possibly abrupt drop-offs which result in vaulting or overturn. Although most accidents may involve only one or two events during this stage, the program has to be able to manage any conceivable scenario.

If more than one mathematical technique is analyzed during the post impact phase, the individual speed changes are combined as the square root of the sum of the squares. For example, the post-impact speed of a vehicle that skidded, rolled and then vaulted after impact would be calculated as follows:

$$v_t = \sqrt{v_s^2 + v_r^2 + v_v^2}$$

where

v_t	=	combined post-impact speed (m/s)
v_s	=	speed change calculated from skid distance
v_r	=	speed change calculated from roll distance
v_v	=	speed at the point of takeoff

The order in which the individual speeds are calculated is irrelevant. The first module would have excluded from the analysis any events that might have followed the vault, because the vault equation calculates the vehicles speed prior to takeoff and not the speed change over the event.

If the post-impact speed cannot be calculated, a reliable eye-witnesses estimate of speed can be used for subsequent calculations. If both of these options fail, a post impact speed cannot be obtained for this vehicle. Since the post-impact speed must be known in order to calculate a vehicles impact speed, the speed loss associated with impact will also be ignored. If there is evidence prior to impact, a speed estimate will be calculated based on these measurements, however, this will likely be very conservative. When a stage of the accident is excluded from the analysis a warning is issued that appears in the speed summary table at the end of the program. This advises the user that any speed estimate obtained from the analysis will be conservative.

4.3.2 The Impact Phase

For well documented accidents where the approach and departure headings are known and the vehicle crush profile has been recorded, it may be possible to use both momentum and energy to calculate the impact speeds. This provides a useful comparison and can help reinforce the first calculation.

An important difference between these two techniques is that energy can be used to calculate the impact speed of a vehicle without any information about the other vehicle. If no post-impact information is available for either vehicle, energy can still be used to calculate the change in velocity experienced by each vehicle. Momentum, on the other hand, requires the post-impact speeds of both vehicles. Therefore, in many instances energy may be the only feasible method for solving the impact speeds.

If both techniques are selected, the program will tell the user which impact speed to carry over to the pre-impact calculations based on the accident configuration and the approximate impact speed. Momentum is best suited for intersection type accidents where the vehicles are approaching one another at approximately 90 degrees. In this situation, marginal errors associated with the approach and departure headings do not have a noticeable effect on the speeds. However, when the vehicles approach paths are almost collinear, small deviations in the headings can have a major effect on the calculated speeds.

The accuracy of the speeds generated from an energy analysis is largely dependent on the PDOF. As the line of approach between the two vehicles starts to deviate from 180 degrees, the range over which the PDOF can act increases. Figure 14 illustrates the range over which the PDOF can act as a function of the vehicles approach headings. In most cases, energy would be best suited for the top configuration while momentum would give the best results for the bottom configuration.

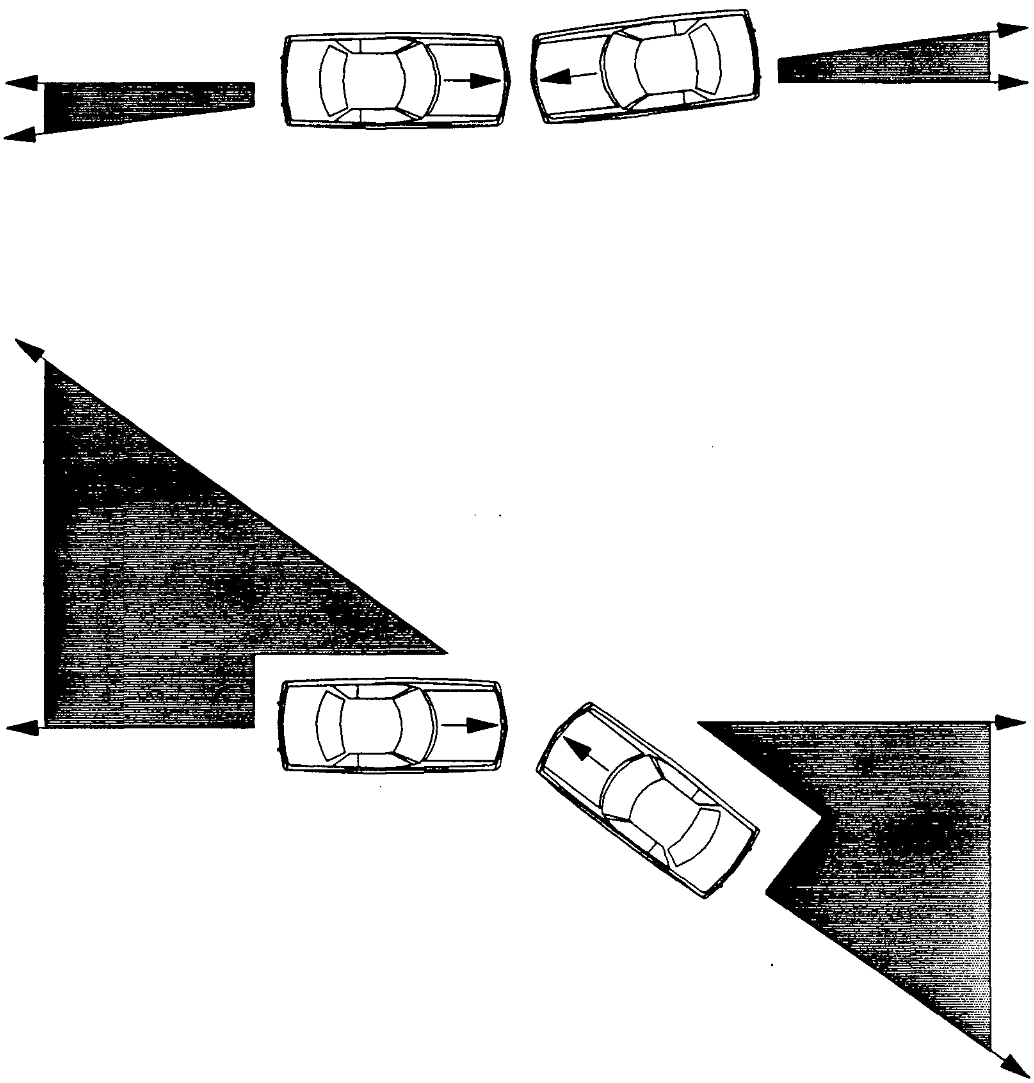


Figure 14: Range of PDOF as a Function of the Approach Headings

Since PDOF is largely a subjective measure (it is usually determined by examining each vehicle's crush profile), the error associated with it increases as the range of the PDOF increases. In addition to this error, the accuracy of the experimentally derived coefficients used to calculate the crush energy is questionable. These coefficients are based on the crash performance of 200 vehicles in staged 20 to 40 mph collisions.

Hence, in the event that both techniques are used to calculate the impact speed, momentum is recommended for any accident where the headings are well defined and the approach paths are not collinear. For head-on and rear-end accidents in the 20 to 40 mph speed range, energy is advised over momentum. The final decision is left to the user.

4.3.3 The Pre-Impact Phase

Analysis of the pre-impact phase usually involves yaw or skid calculations. If a vehicle skids prior to impact the appropriate distance is input and the vehicles initial speed is calculated based on the speed change over this distance and the speed at impact. When a vehicle yaws prior to impact the initial speed can be calculated directly from the yaw mark. Therefore, it is unnecessary to perform calculations for the impact or post-impact phases. However, unless the user specifically requests that these stages be left out of the analysis, the program will go ahead with the appropriate post-impact and impact calculations. The vehicles calculated speed at impact can be used to check that the speed calculated from yaw is reasonable.

In certain situations, skid marks may appear prior to the onset of yaw. The program simply combines the change in velocity over the skid distance with the yaw speed as the square root of the sum of the squares (see Section 4.3.1).

If there is no physical evidence prior to impact, this stage is ignored and the vehicle's speed at impact is considered to be the initial speed.

5. SAMPLE RUN

The following chapter provides an example of how KAR can be used to determine the speeds in a two vehicle accident. It is impossible to demonstrate all of the programs capabilities in a single accident, but some of KAR's significant features are illustrated in this example.

The accident used in this demonstration occurred in the city of Vancouver. The driver of the first vehicle suffered minor injuries, and the driver of the second vehicle was killed. Based on the Vancouver Police Departments preliminary findings, it was believed that the driver of the first vehicle was speeding. A local consulting firm was called in 8 weeks after the accident to conduct a formal reconstruction. All of the data used in their analysis was obtained by the police. Using the same police data, vehicle speeds are calculated using KAR and compared to the results obtained by the consultants. Permission to use this data was granted by the Vancouver City Police Department.

5.1 Summary of the Accident

The accident took place at the intersection of a 4 lane divided arterial roadway (East-West) and a residential side street (North-South). The intersection is controlled by stop signs at both side street approaches. At the time of the accident it was clear and dry and beginning to get dark. Vehicle 1, a Datsun 240-Z, was westbound in the curb lane. Vehicle 2, a Volvo GL, was southbound attempting to cross the arterial roadway. Vehicle 2 had stopped at the sign and was proceeding through the intersection. The driver of Vehicle 2's eastbound line of sight may have been obscured by a large tree and parked cars.

The vehicles collided in the middle of the westbound lane. Vehicle 1 skidded for approximately 20 meters prior to impact, and an additional 8 meters after impact. Vehicle 2 was pushed sideways 15 meters from the point of impact leaving an extensive side scuff mark emanating from the right rear tire. The sudden deviation in Vehicle 1's skid marks made it easy to establish the point of impact. A diagram of the accident is presented in Figure 15.

5.2 Accident Data

Vehicle rest positions, skid marks, and lane locations are all measured with respect to a common reference point using an x-y coordinate system (see appendix D for measurement sheet). From these measurements, skid mark distances, vehicle headings and final resting positions are determined. In addition to this information, each vehicle was photographed and measured by the author to obtain a crush profile for the purpose of a crash analysis. The estimate of the PDOF is based on the damage pattern and knowledge of the vehicle trajectories at impact. Figures 16 and 17 show the crush profile and the PDOF acting on each vehicle. Scale weights plus an additional 90 kilograms (one passenger plus lost fluids) are used as an estimate of the vehicle weights at impact. A complete data sheet is presented in Table 3.

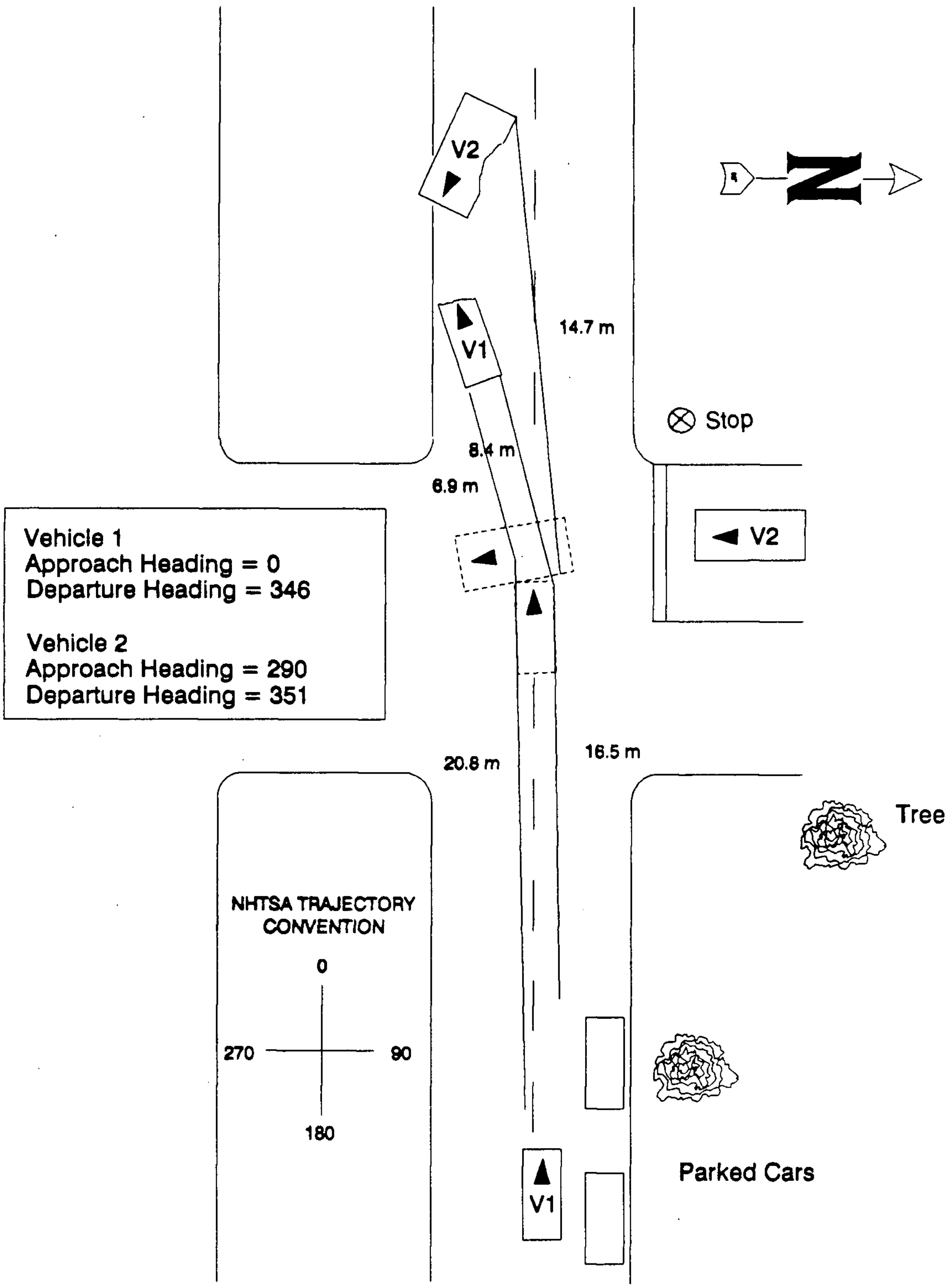


Figure 15: Accident Diagram

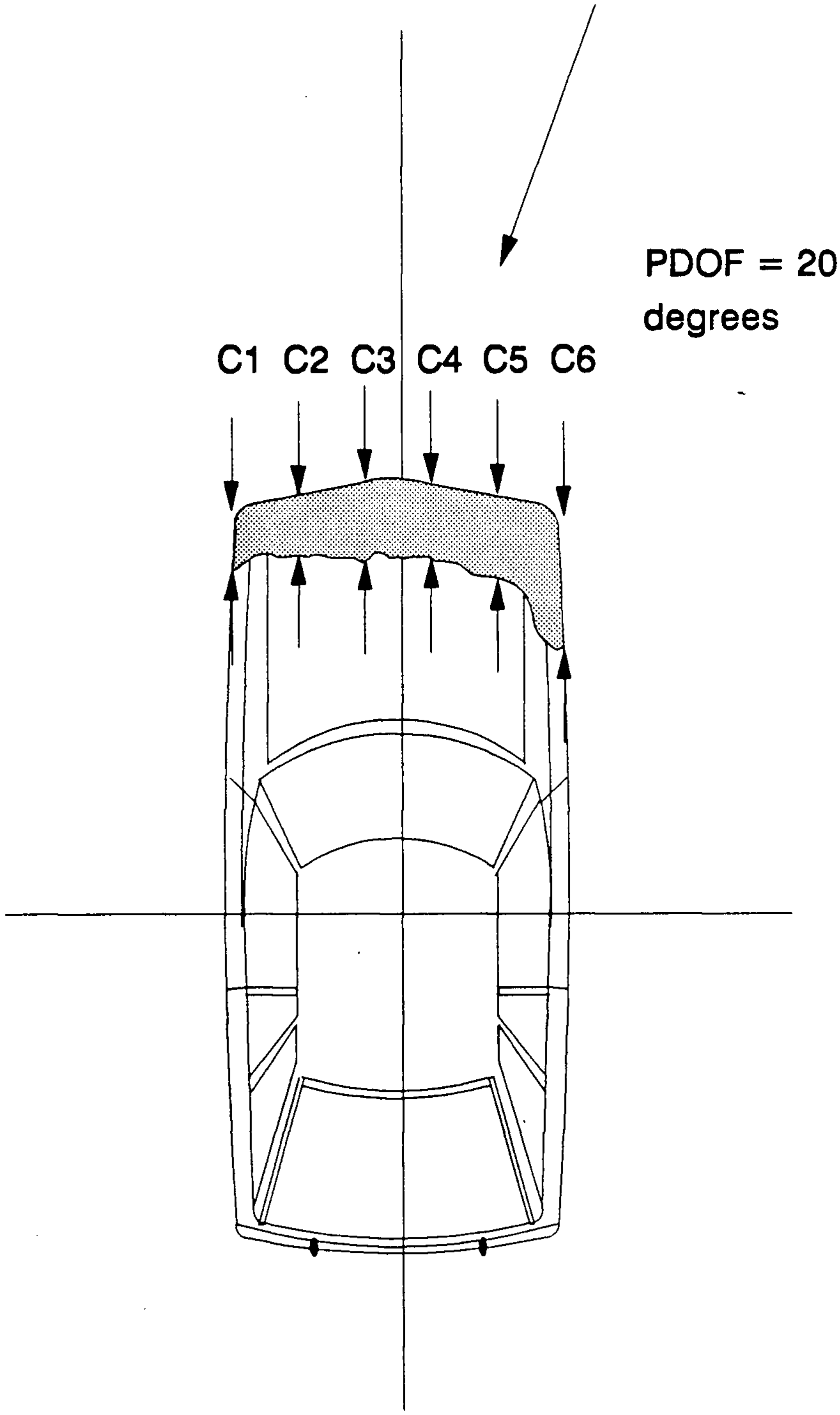


Figure 16: Vehicle 1's Crush Profile

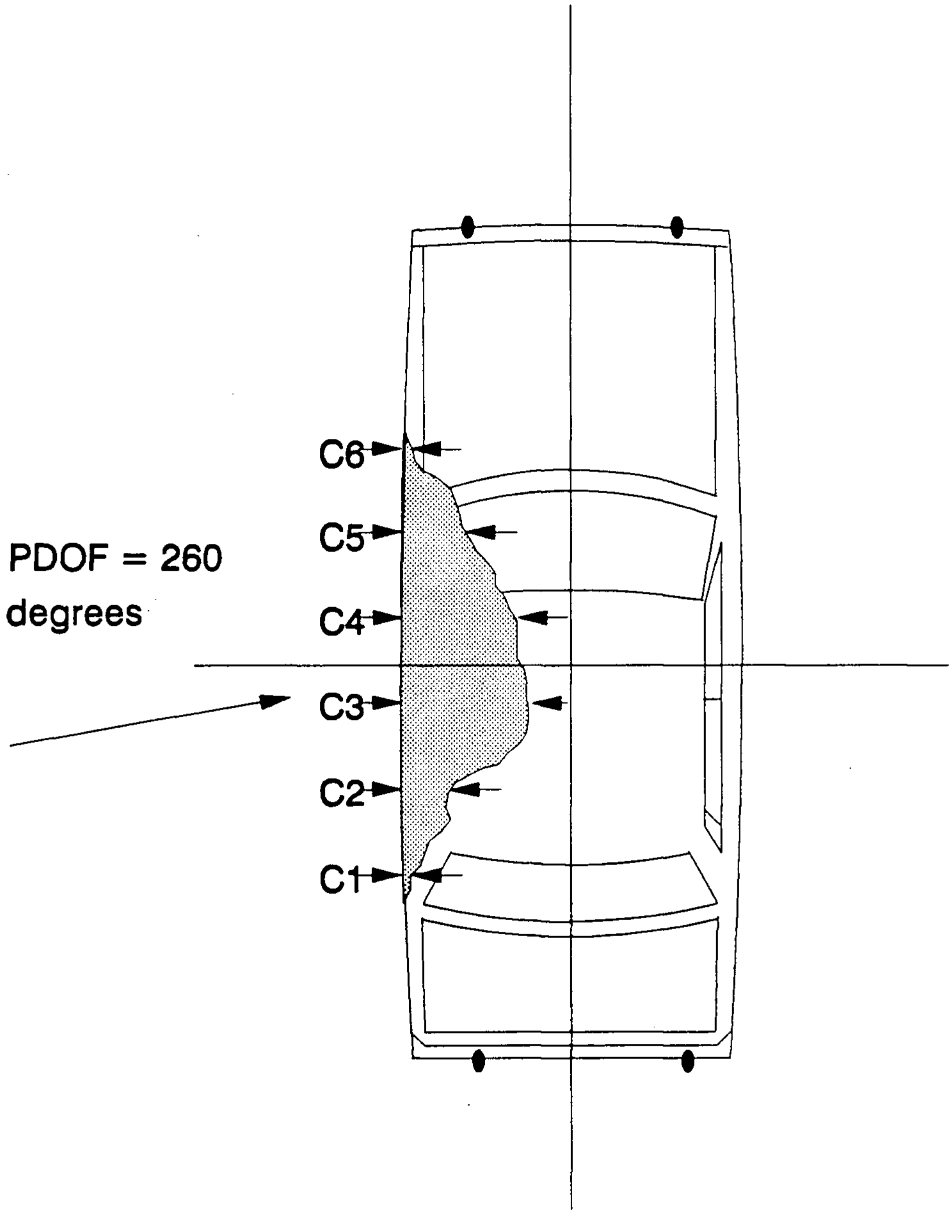


Figure 17: Vehicle 2's Crush Profile

Table 3: Sample Run Data Sheet

	Vehicle 1	Vehicle 2
Skid Marks (m):		
Post-Impact	Left = 6.9 Right = 8.4 cg-cg = 11.0	Left Rear = 2.7 Right Rear = 14.7 cg-cg = 15.5
Comments:	overlap marks	rotation, no front marks
Pre-Impact	Left = 20.8 Right = 16.5	N/A
Comments:	overlap marks	
Headings:		
Approach Departure	0° 346°	290° 351°
Crush Data:		
PDOF	20°	260°
Measurements (cms)		
Width	152	185
c_1	26	10
c_2	26	25
c_3	24	46
c_4	24	56
c_5	29	20
c_6	69	10
Vehicle Specifications:		
Wheelbase (cms)	230	265
Weight	1140	1410
Brake Efficiency	100%	100%

5.3 Analysis

The accident analysis involves two stages corresponding to the separate program modules. The first stage determines which methods are to be used to solve the vehicles speeds. The second stage obtains the "correct data" and makes the appropriate calculations. The first stage involves a short interrogation where the line of questioning is governed by the responses given. Each question represents a condition in a rule. If the condition is not satisfied, the rule fails and another rule is tested. Hence, the line of questioning will vary considerably for each accident.

Below is an excerpt from this session (answers are selected from a screen menu):

Q.1. Can you identify or reasonably estimate the point of impact?
A.1. Yes
Q.2. Do you know the final resting position of Vehicle 1?
A.2. Yes
Q.3. How did Vehicle 1 get from point of impact to its final resting position?
A.3. Skid/Slid
Q.4. Is there a reliable eye-witness estimate of Vehicle 1's post-impact speed?
A.4. No

Since both Vehicles slid to stop, Q.5-Q.7 (for Vehicle 2) same as Q.2-Q.4.

Q.8. Do you know the path of each vehicle as it approaches the point of impact?
A.8. Yes
Q.9. Do you have the approximate vehicle weights?
A.9. Yes
Q.10. Was either vehicle involved in a secondary impact with some other object as they proceeded from the point of impact to the final resting position?
A.10. No
Q.11. Are the vehicles available for inspection or is there numerical documentation of the vehicle damage?
A.11. Yes
Q.12. Did Vehicle 1 leave tire marks prior to impact?
A.12. Yes

Q.13. Where they typical skid marks or yaw marks, or a combination of skidding followed by yaw?
A.13. Skid
Q.14. Did Vehicle 2 leave tire marks prior to impact?
A.14. No

Unfortunately, because of the nature of this accident, the line of questioning at this stage is not very involved. Skid analysis is selected for the post-impact phase of both vehicles and the pre-impact phase of Vehicle 1. Momentum and energy are chosen to analyze the impact speeds. These methods are then sent to the second module for analysis. The information gathered during this session is strictly qualitative.

The line of questioning employed by the second module is considerably more indepth. Although the techniques have been identified, this module must determine which variation of the formula to use and insure there is no confusion about the data being entered. Steps taken to verify the input data are discussed below.

The post-impact speed is calculated for each vehicle using the skid equation. The most important parameter in this equation is the skid distance. When obtaining the skid distance from the user a series of questions are asked to determine which distance should be used in the formula (See section 3.2.1). Vehicle 1 left two overlapping skid marks after impact. In this case the longest mark (8.4 meters) is selected to estimate speed. Vehicle 2 slid sideways and rotated after impact leaving two visible marks emanating from the rear tires. Because of the considerable difference in the length of the marks and the absence of any front tire marks, the distance between the vehicles center of gravity at impact and the final resting position (15.5 meters) is used in the skid formula.

During the pre-impact phase Vehicle 1 left two overlapping skid marks leading up to the point of impact. Because the skid was relatively straight and mechanical braking efficiency was 100 percent, the program assumes that the tires locked at approximately the same time. In this situation the longest mark gives the best indication of speed, but because the marks were overlapping, the wheelbase is subtracted from the skid distance to obtain the true distance each tire slid (in an overlapping skid mark, the beginning of the mark is caused by the rear tire and the end of the mark is caused by the front tire). A skid distance of 18.5 meters is used in this calculation. The consultant made this calculation using two skid assumptions: (i) that the skid marks were from the front wheels of the vehicle (20.8 meters); (ii) that the measured skid marks were from both the front and rear wheels (18.5 meters).

Both momentum and energy are used to calculate the speeds at impact. Since the vehicle headings are well defined, the program calculates each vehicles PDOF for use in the energy equations (see section 3.2.5 for check routine). The values obtained in the field were out by approximately 10 degrees. Using the suggested values a more reasonable speed estimate is obtained. Due to the high speed of Vehicle 1 and the configuration of the accident, the program considers the impact speeds calculated by momentum to be more accurate than energy. These speeds are used in Vehicle 1's pre-impact skid analysis.

Although a drag sled was used to determine the drag factor (0.71), the system was asked to suggest a value based on Table 1, stored in the knowledge base. For this road surface a factor of 0.70 is recommended for speeds less than 50 km/h. For speeds in excess of 50 km/h the recommended factor drops to 0.63. The consulting engineer made the same assumption. For the post-impact skid calculations he used the sled value of 0.71, but to calculate the pre-impact speed of the Datsun he reduced this factor to 0.65 to compensate for the vehicle's high speed.

In Table 4 the calculated speeds for each phase of the accident are outlined and compared to values obtained by the consulting engineer. These values seem to be in close agreement.

Table 4: Summary of Speed Estimates for Different Stages

Accident Stage	KAR Vehicle 1	KAR Vehicle 2	Consultant Vehicle 1	Consultant Vehicle 2
Post-Impact (km/h)				
Skid Analysis	39	53	39	48
Impact				
Momentum	95	17	90	16
Energy	76	-	-	-
Pre-Impact				
Skid Analysis * using momentum impact speeds	110	17	105	16

The consultant assumed Vehicle 2's heading at impact to be greater than what I assumed. In addition, impact and pre-impact speeds were calculated with the assistance of EDCRASH - an enhanced and improved version of CRASH III - which provides a very sophisticated impact and spin-out analysis. This would explain the slightly lower values obtained by the consultant.

6. FURTHER RESEARCH REQUIREMENTS

A major part of accident reconstruction involves the visual interpretation of physical evidence. To accommodate this, rules have been established that produce a "mental picture" of the accident. These rules are necessary to ensure the correct methods of analysis and the proper data are used. For example, determining the types of surfaces each vehicle slides over and its orientation with respect to each surface, require a series of lengthy queries that must be qualified before the analysis can continue.

While these rules are the "heart" of an interpretive expert system like KAR, it is possible to hide them from the user. Currently, the system works by interpreting the users responses to the queries. With a graphics routine and a fully supportive database, the rules in the knowledge base could obtain the necessary information from the database instead of the user.

Police data is gathered according to an on-scene x-y coordinate system. Vehicle positions, skid marks, debris, point of impact, lane widths and curb locations are defined in terms of a common reference point. Before this data can be analyzed by KAR it must be converted to distances and headings. In some cases, different road surfaces and elevations must also be considered. This conversion process is not only cumbersome, but also a potential source of error.

However, with the appropriate tools, physical evidence and road surface information could be input directly from the police measurement sheet. A graphics routine could produce a scene diagram of the accident for visual verification, and the data could be stored in coordinate form in a database. A separate program could be used to superimpose the physical evidence and the road surface information, and convert it into a form interpreted by the rules in the knowledge base. The user would simply enter the police generated coordinates of each piece of evidence into the system at the appropriate prompt. The program would then convert these coordinates to distances, directions, etc. and apply the appropriate judgement in order to calculate the vehicle speeds.

To date no existing computer program has been able to perform at this level of sophistication because of its inability to "reason". It relies on the user to apply his or her judgement before performing any calculations. When the user is an experienced investigator this expertise is invaluable. However, the judgement exercised by a non-technical user can produce disastrous results.

The knowledge encoded into KAR is the first step in designing a system that would be able to predict vehicle speeds based solely on the physical evidence. The existing knowledge base is only a prototype and would require considerable investment in development time before it could attain the level of performance demanded by an accident investigator. The rules used to interpret evidence would have to be refined and expanded to include many scenarios. Complex equations of motion used to analyze spin-out trajectories and vehicle interactions would also need to be added. A sophisticated explanation facility would be required to produce a summary of the all the assumptions made and the reasons behind these assumptions.

KAR also has potential as a teaching tool for police. At this stage, the first module of the program identifies how to solve the accident speeds based on a qualitative line of questioning. The reasoning involved in this segment of the program can be accessed through the explanation facilities. This provides the user with an understanding of what the investigator is looking for and why. This segment would require more refinement before it could be considered representative of an experts knowledge.

The second module does not contain any rules that help the user identify scene evidence. Some statements have been included in the rules that tell the user what to look for, but more work must be done in this area before the system could be used as an effective training tool.

7. CONCLUSIONS

The system developed in this paper is a good example of an expert system operating in a complex domain. The problem was confined to vehicle speed calculation in accident reconstruction to simplify the development of a realistic prototype. Creating an expert system for this purpose is difficult because the judgement exercised by engineers in this field is ill-defined. A considerable amount of work is still required before the system could operate at the level of an expert. It is expected that a production system would take a minimum of 5 man years to implement.

At this point KAR utilizes a combination of heuristic and conventional procedures. It is capable of predicting speeds using any combination of the following seven mathematical techniques: skid, roll, overturn, yaw, vault, momentum and energy. Although the program's analytical capabilities are extremely comprehensive, the system's most notable feature is it's interpretive abilities. These abilities include being able to determine how the accident should be solved, and verifying the evidence before it is used in the calculations.

KAR is not an instructive expert system in the strict sense of the definition, however, it's ability to explain it's reasoning provide the non-technical user with considerable insight into accident reconstruction.

Each year, millions of research dollars are spent to improve the accuracy and validate conventional programs like SMAC and CRASH. While this research is fruitful and should not cease, attention must be directed to the major sources of error in "real-life" accidents. These error sources include interpretation and identification of scene evidence.

KAR demonstrates a portion of the knowledge used by an engineer to determine vehicle speeds based on physical evidence.

BIBLIOGRAPHY

Artificial Intelligence and Expert Systems

1. Brown, J.S. and Burton, R.R. (1975). "Multiple Representations of Knowledge for Tutorial Reasoning." Representation and Understanding: Studies in Cognitive Science, Academic Press, New York, pp. 311-349.

2. Buchanan, B.G. and Fergenbaum, F.A. (1978). "DENDRAL and Meta-DENDRAL: Their Applications Dimension." Artificial Intelligence 11, pp. 5-24.

3. Citrenbaum, R., Geissman, J.R. and Schultz, R. (1987). "Selecting a Shell." AI Expert, Vol. 2 No. 9.

4. Dankel, D.D. (1987). "Knowledge Representation for Expert Systems: A Survey and Evaluation of Techniques." Artificial Intelligence: Recent Developments, Society of Automotive Engineers, Inc., Warrendale, PA.

5. Duda, R.O., Gaschnig, J.G., and Hart, P.E. (1979). "Model Design in the PROSPECTOR Consultant System for Mineral Exploration." Expert Systems in the Micro-Electronic Age, Edinburgh Press, Edinburgh, pp. 153-167.

6. Erman, L.D., Hayes-Roth, F., Lesser, V., and Reddy, D. (1980). "The HEARSAY-II Speech-Understanding System: Integrated Knowledge to Solve Uncertainty." Computing Surveys 12, No. 2, pp. 213-253.

7. Feigenbaum, E.A. (1977). "The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering." Proceedings of the 5th International Joint Conference on Artificial Intelligence, pp. 1014-1029.

8. Hayes-Roth, F., Waterman, D.A. and Lenat D.B., eds. (1983). Building Expert Systems. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts.

9. Hunt, V.D. (1986). Artificial Intelligence and Expert Systems Sourcebook. Chapman and Hall, New York.

10. Jackson, P. (1986). Introduction to Expert Systems. Addison-Wesley Publishing Company Inc., Reading, Massachusetts.

11. Lenat, D.B., Sutherland, W.R. and Gibbons, J. (1982). "Heuristic Search for New Microcircuit Structures: An Application of Artificial Intelligence." AI Magazine 3, pp. 17-33.

12. Lindsay, R.K., Buchanan, B.G., Feigenbaum, E.A., and Lederberg, J. (1980). Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project. McGraw-Hill, New York.

13. McDermott, J. (1981). "R1's Formative Years." AI Magazine, Vol. 2, No. 2, pp. 21-29.

14. Minsky, M. (1975). "A Framework for Representing Knowledge.", The Psychology of Computer Vision. McGraw-Hill, New York.

15. Prerau, D.S. (1987). "Knowledge Acquisition in the Development of a Large Expert System." AI Magazine, Vol. 8, No. 2, pp. 43-51.

16. Sacerdoti, E.D. (1974). "Planning in a Hierarchy of Abstraction Spaces." Artificial Intelligence 5, No. 2, pp. 115-135.

17. Sacerdoti, E.D. (1977). A Structure for Plans and Behavior. American Elsevier, New York.

18. Shortliffe, E.H. (1976). Computer-Based Medical Consultations: MYCIN. Elsevier, New York.

19. Sterling, L. and Shapiro, E. (1986). The Art of Prolog. MIT Press, Cambridge, Massachusetts.

20. Van Horn M. (1986). Understanding Expert Systems. Bantam Books, New York.

21. Expert Systems: Their Application in the Canadian Transportation Sector. (1986). Transport Canada, TP 7328E.

Accident Reconstruction

22. Anderson, D.O., Woolley, R.L., Tagg, M.D. and Warner, C.Y. (1985). "The VTS Single-Vehicle Trajectory Simulation." International Congress and Exposition, February 1985, SAE Paper No. 850252.

23. Baker, J.S. (1975). Traffic Accident Investigation Manual. The Traffic Institute, Northwestern University, Evanston, Illinois.

24. Benner, L. (1980). "Accident Investigations - A Case for New Perceptions and Methodologies." SAE Paper No. 800387.

25. Byatt, R. and Watts, R. (1980). Manual of Road Accident Investigation, Volume 1. Pitman Publishing Limited, London.

26. Collins, J.C. (1979). Accident Reconstruction. Charles C Thomas Publisher, Springfield, Illinois.

27. Day, T.D. and Hargens, R.L. (1985). "Differences between EDCRASH and CRASH 3." International Congress and Exposition, February 1985, SAE Paper No. 850253.

28. Emori, R.I. (1968). "Analytical Approach to Automobile Collisions." SAE Paper No. 680016.

29. Emori, R.I. (1968). "Mechanics of Automobile Collisions." First International Conference on Vehicle Mechanics, Wayne State University.

30. Hess, R.L. (1980). "Microcomputer Programs Useful for Aspects of Accident Reconstruction." SAE Paper No. 800174.

31. James, M.E. and Ross, H.E. (1976). "Improvement of Accident Simulation Model." Texas A&M Research Foundation Report No. RF3258-1.

32. Jones, I.S. (1974). "Results of Selected Applications to Actual Highway Accidents of the SMAC Reconstruction Program." 18th Stapp Car Crash Conference.

33. Jones, I.S. and Segal, D.J. (1974). "The Application of the SMAC Accident Reconstruction Program to Actual Highway Accidents." Proceeding of the 18th Conference of the American Association for Automotive Medicine.

34. Limpert, R. (1978). Motor Vehicle Accident Reconstruction and Cause Analysis. The Michie Company, Charlottesville, Virginia.

35. Lofgren, M.J. (1977). Handbook for the Accident Reconstructionist. Princeton Publishing, Princeton, Minnesota.

36. Martinez, J.E. (1985). "Computer Aided Automobile Accident Reconstruction-Theory, Graphics, Animation." International Congress and Exposition, SAE Paper No. 850250.

37. McCarthy, J.R. and Dulion, E.S. (1985). CRASH3PC Operations Manual. J.R. McCarthy Company, Simi Valley, California.

38. McGrew, D.R. (1976). Traffic Accident Investigation and Physical Evidence. Charles C Thomas Publisher, Springfield, Illinois.

39. McHenry, R.R. (1973). "A Computer Program for Reconstruction of Highway Accidents." SAE Paper No. 730980.

40. McHenry, R.R. (1975). "A Comparison of Results Obtained With Different Analytical Techniques for Reconstruction of Highway Accidents." SAE Paper No. 750893.

41. McHenry, R.R. (1976). "Extensions and Refinements of the CRASH Computer Program Part 1 - Analytical Reconstruction of Highway Accidents." NTIS PB76-252114.

42. McHenry, R.R. and Lynch, J.P. (1976). "Mathematical Reconstruction of Highway Accidents - Further Extensions and Refinements of the CRASH Computer Program." DOT-HS-802-287, Calspan Report No. ZQ-5708-V-5.

43. McHenry, R.R. and Lynch, J.P. (1979). "Revision of the CRASH2 Computer Program." USDOT Publication No. DOT-HS-805-209.

44. McHenry, R.R. and McHenry, B.G. (1986). "A Revised Damage Analysis Procedure for the CRASH computer program." SAE Paper No. 861894.

45. McHenry, R.R., Segal, D.J., Lynch, J.P. and Henderson, P.M. (1973). "Mathematical Reconstruction of Highway Accidents." Calspan Report No. ZM-5096-V-1.

46. Morton, W.C. ed. (1985). Accident Reconstruction Manual and Workbook. National Police Institute, Central Missouri State University, Warrensburg, Missouri.

47. Noga, T. and Oppenheim, T. (1981). CRASH3 User's Guide and Technical Manual. USDOT.

48. Nowak, R., Nowak, E., Tryphonopoulos, J. and German, A. (1987). "A Fundamental Momentum Energy Model for Central Motor Vehicle Collisions and a Critical Review of the Damage Algorithm in CRASH." Proceedings of the Canadian Multidisciplinary Road Safety Conference V.

49. Rivers, R.W. (1980). Traffic Accident Investigators' Handbook Charles C Thomas Publisher, Springfield, Illinois.

50. Smith, R.A. and Noga, J.T. (1980). "Examples of Staged Collisions in Accident Reconstruction." American Society of Mechanical Engineers.

51. Woolley, R.L. (1985). "The IMPAC Computer Program for Accident Reconstruction." International Congress and Exposition, SAE Paper No. 850254.

52. Zeidler, F., Schreier, H. and Stadelmann, R. (1985). "Accident Research and Reconstruction ... by the EES-Accident Reconstruction Method." International Congress and Exposition, SAE Paper No. 850256.

APPENDIX A. Derivation of Equations

Skid Equation

$$E_1 = E_2 + Fd$$

where

E_1	=	kinetic energy of vehicle prior to skid	$= \frac{1}{2} m v_i^2$
E_2	=	kinetic energy of vehicle after skid	$= \frac{1}{2} m v_f^2$
F	=	friction force (drag factor times vehicle wt.)	$= fmg$
d	=	skid distance	

$$\frac{1}{2} m v_i^2 = \frac{1}{2} m v_f^2 + fmgd$$

solving for v_i

$$v_i = \sqrt{2fgd + v_f^2}$$

Variations of the formula used in the program are of the form:

$$v_i = \sqrt{2g(f_1 d_1 + f_2 d_2 + \ldots f_n d_n) + v_f^2}$$

Yaw Equation

This equation is derived for a vehicle cornering on an inclined slope. The parameters used in this derivation are described below.

ϕ	=	angle of the slope
N	=	normal force acting perpendicular to the slope
a_n	=	normal acceleration $= \frac{v^2}{R}$
R	=	radius of the yaw mark

summing the forces parallel to the slope

$$ma_n\cos\phi = fN + mg\sin\phi$$

summing the forces normal to the slope

$$ma_n\sin\phi = N - mg\cos\phi$$

solving for N and substituting into the first equation

$$ma_n\cos\phi = f(ma_n\sin\phi + mg\cos\phi) + mg\sin\phi$$

replacing a_n with $\frac{v^2}{R}$ and rearranging

$$v^2 = Rg\frac{f\cos\phi + \sin\phi}{\cos\phi - f\sin\phi}$$

substituting $\tan\phi = e$ (percent elevation of the slope) and solving for v

$$v = \sqrt{Rg\frac{(f+e)}{(1-fe)}}$$

The radius of the yaw mark R can be determined by measuring a chord distance C and the middle ordinate M (distance between the chord and the yaw mark at midpoint of the chord). R can then be stated as follows:

$$R = \frac{C^2}{8M} + \frac{M}{2}$$

Vault Equation

The vault equation can be derived using simple kinematic relationships. The vertical distance the vehicle travelled, h, is equal to:

$$h = -v\sin\phi T + \frac{1}{2}gT^2$$

the horizontal distance, d, is equal to:

$$d = v\cos\phi T$$

solving for T and substituting into the first equation

$$h = -d\tan\phi + \frac{gd^2}{2v^2\cos^2\phi}$$

substituting $\tan\phi = e$ and setting $\cos\phi = 1$ (small angle approximation)

$$v = \sqrt{\frac{gd^2}{2(h+de)}}$$

Energy Equations

Energy is based on the principle that the total pre-impact kinetic energy is equal to the total post-impact kinetic energy plus the energy absorbed in crushing the vehicles to their deformed state.

$$\frac{1}{2}m_1v_1^2 + \frac{1}{2}m_2v_2^2 = \frac{1}{2}(m_1 + m_2)v_c^2 + E_1 + E_2$$

For an inelastic central collision it is assumed that both vehicles attain a common velocity v_c such that

$$m_1 v_1 + m_2 v_2 = (m_1 + m_2) v_c$$

solving for Vehicle 1's pre-impact velocity and substituting into the energy equation results in the following quadratic equation

$$v_2^2 - 2 v_c v_2 + v_c^2 - \frac{2(E_1 + E_2)}{m_2(1 + m_2/m_1)} = 0$$

solving for v_2

$$v_2 = v_c \pm \sqrt{\frac{2\,(E_1+E_2)}{m_2\left(1+m_2/m_1\right)}}$$

the change in velocity experienced by Vehicle 2 ($\Delta v_2 = v_2 - v_c$) is thus equal to

$$\Delta v_2 = \sqrt{\frac{2\,(E_1+E_2)}{m_2\left(1+m_2/m_1\right)}}$$

by substituting Vehicle 2's pre-impact speed into the energy equation a similar expression for Vehicle 1 can be written

$$\Delta v_1 = \sqrt{\frac{2\,(E_1+E_2)}{m_1\left(1+m_1/m_2\right)}}$$

APPENDIX B. Module 1 of the Program

```
!=================================================================
!                          KAR1 Module
!=================================================================

EXECUTE;
BKCOLOR = 3;
RUNTIME;
ENDOFF;

ACTIONS
      WOPEN 3,0,0,23,79,2
      ACTIVE 3
      LOCATE 5,8
      DISPLAY "
                    WELCOME TO

                      KAR

           (Knowledge-based Accident Reconstruction).


           (Press any key to begin consultation)~"
      CLS
      WCLOSE 3
      WOPEN 9,0,0,11,79,3
      ACTIVE 9
                                        ! Find out if user wants to select
      FIND Solution                     ! method or have program determine an
                                        ! appropriate approach
      SAVEFACTS karvalue
      CHAIN kar2;


!=================================================================
!           General Rules to Determine if User Wants to Select
!      Solution Methods or if The Machine Selects Solution Methods
!=================================================================


RULE SOLN1                              ! If the user wants the system to
IF    Soln_Choice = Find_Methods        ! select the methods of solution
THEN                                    ! this rule is executed.
      CLS
      WOPEN 3,0,0,23,79,7
      ACTIVE 3
      DISPLAY "
```

```
        YOUR RESPONSES TO THE FOLLOWING QUESTIONS HELP KAR
        DETERMINE THE

        MOST APPROPRIATE METHOD(S) FOR SOLVING INITIAL VEHICLE
        SPEEDS.





                (Press any key to continue ....)~"
        CLS
        WCLOSE 3
        ACTIVE 9
        FIND Method_Postv1
        CLS
        FIND Method_Postv2
        CLS
        FIND Method_Impact
        CLS
        FIND Method_Prev1
        CLS
        FIND Method_Prev2
        CLS
        Solution = Found;

RULE    SOLN2                                      ! If user wants to choose his
IF      Soln_Choice = Choose_Methods AND           ! own methods of solution,
        Select_Postv1 <> UNKNOWN AND               ! this rule is executed.
        Select_Postv2 <> UNKNOWN AND
        Select_Impact <> UNKNOWN AND
        Select_Prev1 <> UNKNOWN AND
        Select_Prev2 <> UNKNOWN
THEN
        CLS
        Y=0
        WHILEKNOWN Y                               ! This is necessary because
                Y=(Y+1)                            ! VP-Expert will not pass the
                POP Select_Postv1,J1               ! entire contents of a plural
                Method_Postv1 = (J1)               ! variable to another plural
                POP Select_Postv2,J2               ! variable. It will only pass
                Method_Postv2 = (J2)               ! stack. With a POP enclosed
                Method_Impact = (J3)               ! in a WHILEKNOWN, each value
                POP Select_Prev1,J4                ! is removed from the stack
                Method_Prev1 = (J4)                ! and assigned to the other
                POP Select_Prev2,J5                ! plural variable.
                Method_Prev2 = (J5)
                RESET Garbage
                FIND Garbage
        END
```

```
        RESET Y
        Solution = Found;

RULE PLURAL                             ! Since there is no way of
IF      Y>3                             ! determining the number of
THEN                                    ! values in a plural variable
        RESET Y                         ! (VP-Expert strikes again), I
        Garbage = Found                 ! assume that only 3 calculative
ELSE                                    ! methods will be necessary for
        Garbage = Found;                ! each stage of the accident.


!==================================================================!
             General Rules to Determine Post-Impact Method
!==================================================================


RULE GEN1A                              ! Can skid analysis be used to
IF      Pt_Impact = Yes AND             ! determine post-impact speed.
        Rest_Posv1 = Yes AND
        Post_Actv1 = Skid
THEN
        Method_Postv1 = Skid_Analysis;

RULE GEN1AA
IF      Pt_Impact = No AND
        Approx_Imp = Yes AND
        Rest_Posv1 = Yes AND
        Post_Actv1 = Skid
THEN
        Method_Postv1 = Skid_Analysis;

RULE GEN1B
IF      Post_Actv1 = Yaw AND
        Surf_Postv1 = Yes AND           ! Add rule that ask if vehicle
        Flat_Postv1 = Yes               ! steered away from curved path
THEN
        Method_Postv1 = Yaw
BECAUSE "The equation used to calculate yaw speed is dependent on a constant
drag factor acting on each tire and that gravity remain constant for the
duration of the event.";

RULE GEN1C
IF      Post_Actv1 = Roll
THEN
        Method_Postv1 = Roll;

RULE GEN1D
IF      Post_Actv1 = Vault
THEN
        Method_Postv1 = Vault;
```

```
RULE GEN1E
IF      Post_Actv1 = Overturn
THEN
        Method_Postv1 = Overturn;

RULE GEN1F
IF      Post_Witv1 = Yes AND
        Post_Spv1 <> UNKNOWN                        ! May affect results for plural
THEN                                                ! analysis.
        Method_Postv1 = Witness;

RULE GEN1G
IF      Post_Witv1 = No AND
        Post_Actv1 = Dont_Know AND
        Post_Zerov1 = Yes
THEN
        Post_Speedv1 = 0
        Method_Postv1 = Assume;

RULE GEN1H
IF      Post_Witv1 = No AND
        Post_Actv1 = Dont_Know AND
        Post_Zerov1 = No
THEN
        Method_Postv1 = None;


RULE GEN2A                                          ! Can skid analysis be used to
IF      Pt_Impact = Yes AND                         ! determine post-impact speed.
        Rest_Posv2 = Yes AND
        Post_Actv2 = Skid
THEN
        Method_Postv2 = Skid_Analysis;

RULE GEN2AA
IF      Pt_Impact = No AND
        Approx_Imp = Yes AND
        Rest_Posv2 = Yes AND
        Post_Actv2 = Skid
THEN
        Method_Postv1 = Skid_Analysis;

RULE GEN2B
IF      Post_Actv2 = Yaw AND
        Surf_Postv2 = Yes AND
        Flat_Postv2 = Yes
THEN
        Method_Postv2 = Yaw
BECAUSE "The equation used to calculate yaw speed is dependent on a constant
drag factor acting on each tire and that gravity remain constant for the
duration of the event.";
```

```
RULE GEN2C
IF      Post_Actv2 = Roll
THEN
        Method_Postv2 = Roll;

RULE GEN2D
IF      Post_Actv2 = Vault
THEN
        Method_Postv2 = Vault;

RULE GEN2E
IF      Post_Actv2 = Overturn
THEN
        Method_Postv2 = Overturn;

RULE GEN2F
IF      Post_Witv2 = Yes AND
        Post_Spv2 <> UNKNOWN                    ! May affect results for plural
THEN                                            ! analysis.
        Method_Postv2 = Witness;

RULE GEN2G
IF      Post_Witv2 = No AND
        Post_Actv2 = Dont_Know AND
        Post_Zerov2 = Yes
THEN
        Post_Speedv2 = 0
        Method_Postv2 = Assume;

RULE GEN2H
IF      Post_Witv2 = No AND
        Post_Actv2 = Dont_Know AND
        Post_Zerov2 = No
THEN
        Method_Postv2 = None;


!==============================================================!
                General Rules to Determine Impact Method
!==============================================================


RULE GEN3A                              ! Checks if Momentum can be used.
IF      Method_Postv1 <> None AND       ! If post-impact speeds are unknown
        Method_Postv2 <> None AND       ! cannot use momentum
        Approach_Paths = Yes AND
        Departure_Paths = Yes AND
        Veh_Wts = Yes AND
        Sec_Impact = No
THEN
        Method_Impact = Momentum;
```

```
RULE GEN3B                                  ! Checks if departure paths can
IF      Pt_Impact = Yes AND                 ! be determined.
        Rest_Posv1 = Yes AND
        Rest_Posv2 = Yes
THEN
        Departure_Paths = Yes;

RULE GEN3C                                  ! Queries if there was a secondary
IF      Secondary = No                      ! impact.
THEN
        Sec_Impact = No;

RULE GEN3D                                  ! Can secondary impact be
IF      Secondary = Yes AND                 ! compensated for.
        Sec_Compensate = Yes
THEN
        Sec_Impact = No;

RULE GEN3E                                  ! Checks if energy can be used.
IF      Method_Postv1 <> None AND           ! If post-impact speeds are
unknown
        Method_Postv2 <> None AND           ! cannot determine impact speed,
        Available = Yes AND                 ! only change in speed.
        Veh_Wts = Yes
THEN
        Method_Impact = Energy;

RULE GEN3F
IF      Method_Impact <> Momentum AND
        Method_Impact <> Energy
THEN
        Method_Impact = None;


!=============================================================!
        General Rules to Determine Pre-Impact Method
!=============================================================


RULE GEN4A                                  ! Pre-impact skid analysis V1
IF      Pre_Skidv1 = Yes AND
        Type_Prev1 = Skid
THEN
        Method_Prev1 = Skid_Analysis;
```

```
RULE GEN4B
IF      Pre_Skidv1 = Yes AND
        Type_Prev1 = Yaw AND
        Surf_Prev1 = Yes AND
        Flat_Prev1 = Yes
THEN
        Method_Prev1 = Yaw
BECAUSE "The equation used to calculate yaw speed is dependent on a constant
drag factor acting on each tire and that gravity remain constant for the
duration of the event.";

RULE GEN4C
IF      Pre_Skidv1 = Yes AND
        Type_Prev1 = Skid_Yaw AND
        Surf_Prev1 = Yes AND
        Flat_Prev1 = Yes
THEN
        Method_Prev1 = Skid_Analysis
        Method_Prev1 = Yaw
BECAUSE "The equation used to calculate yaw speed is dependent on a constant
drag factor acting on each tire and that gravity remain constant for the
duration of the event.";

RULE GEN4D
IF      Pre_Witv1 = Yes AND
        Pre_Spv1 <> UNKNOWN
THEN
        Method_Prev1 = Witness;

RULE GEN4E
IF      Pre_Skidv1 = No AND
        Pre_Witv1 = No
THEN
        Method_Prev1 = None;

RULE GEN5A                                    ! Pre-impact skid analysis V2
IF      Pre_Skidv2 = Yes AND
        Type_Prev2 = Skid
THEN
        Method_Prev2 = Skid_Analysis;

RULE GEN5B
IF      Pre_Skidv2 = Yes AND
        Type_Prev2 = Yaw AND
        Surf_Prev2 = Yes AND
        Flat_Prev2 = Yes
THEN
        Method_Prev2 = Yaw
BECAUSE "The equation used to calculate yaw speed is dependent on a constant
drag factor acting on each tire and that gravity remain constant for the
duration of the event.";
```

```
RULE GEN5C
IF      Pre_Skidv2 = Yes AND
        Type_Prev2 = Skid_Yaw AND
        Surf_Prev2 = Yes AND
        Flat_Postv1 = Yes
THEN
        Method_Prev2 = Skid_Analysis
        Method_Prev2 = Yaw
BECAUSE "The equation used to calculate yaw speed is dependent on a constant
drag factor acting on each tire and that gravity remain constant for the
duration of the event.";

RULE GEN5D
IF      Pre_Witv2 = Yes AND
        Pre_Spv2 <> UNKNOWN
THEN
        Method_Prev2 = Witness;

RULE GEN5E
IF      Pre_Skidv2 = No AND
        Pre_Witv2 = No
THEN
        Method_Prev2 = None;


!=============================================================!
                        Queries To User
!=============================================================

!------------------- General Queries -------------------------------


ASK Soln_Choice: "Would you like KAR to determine the appropriate methods for
determining speed, or would you like to choose the methods directly?";
CHOICES Soln_Choice: Find_Methods, Choose_Methods;


!---------------- Find_Methods Queries -----------------------------


ASK Pt_Impact: "Can you identify the point of impact?";
CHOICES Pt_Impact: Yes,No;

ASK Approx_Imp: "Can you identify a set of boundaries within which the
point of impact must have occurred?";
CHOICES Approx_Imp: Yes,No;


ASK Rest_Posv1: "Do you know the final resting position of Vehicle 1?";
CHOICES Rest_Posv1: Yes,No;
```

ASK Rest_Posv2: "Do you know the final resting position of Vehicle 2?";
CHOICES Rest_Posv2: Yes,No;

ASK Post_Actv1: "How did Vehicle 1 get from point of impact to its final resting position?
(You can select more than one action, press the END key to continue)
NOTE: If the vehicle yawed and/or vaulted during this sequence, disregard any subsequent events. Yaw and vault give an estimate of speed at the beginning of the event whereas the other techniques give the change in speed over the duration of the event.";
CHOICES Post_Actv1: Skid,Yaw,Roll,Overturn,Vault,Dont_Know;

ASK Post_Actv2: "How did Vehicle 2 get from point of impact to its final resting position?
(You can select more than one action, press the END key to continue)
NOTE: If the vehicle yawed and/or vaulted during this sequence, disregard any subsequent events. Yaw and vault give an estimate of speed at the beginning of the event whereas the other techniques give the change in speed over the duration of the event.";
CHOICES Post_Actv2: Skid,Yaw,Roll,Overturn,Vault,Dont_Know;

ASK Post_Witv1: "Is there a reliable eye-witness estimate of Vehicle 1's post-impact speed?";
CHOICES Post_Witv1: Yes,No;

ASK Post_Witv2: "Is there a reliable eye-witness estimate of Vehicle 2's post-impact speed?";
CHOICES Post_Witv2: Yes,No;

ASK Post_Spv1: "Please enter post-impact speed estimate of Vehicle 1 (km/h).";

ASK Post_Spv2: "Please enter post-impact speed estimate of Vehicle 2 (km/h).";

ASK Pre_Witv1: "Is there a reliable eye-witness estimate of Vehicle 1's pre-impact speed?";
CHOICES Pre_Witv1: Yes,No;

ASK Pre_Witv2: "Is there a reliable eye-witness estimate of Vehicle 2's pre-impact speed?";
CHOICES Pre_Witv2: Yes,No;

ASK Pre_Spv1: "Please enter pre-impact speed estimate of Vehicle 1 (km/h).";

ASK Pre_Spv2: "Please enter pre-impact speed estimate of Vehicle 2 (km/h).";

ASK Approach_Paths: "Do you know the path of each vehicle as it approaches the point of impact?";
CHOICES Approach_Paths: Yes,No,Not_Sure;

ASK Veh_Wts: "Do you have the approximate vehicle weights?";
CHOICES Veh_Wts: Yes,No;

ASK Separate: "Did the vehicles separate after impact?";
CHOICES Separate: Yes,No;

ASK Secondary: "Is either vehicle involved in a secondary impact with some other object as they proceed from point of impact to final resting position?";
CHOICES Secondary: Yes,No;

ASK Sec_Compensate: "Can you compensate for the speed loss associated with the secondary impact?";
CHOICES Sec_Compensate: Yes,No;

ASK Available: "Are the vehicles available for inspection or is there numerical documentation of the vehicle damage?";
CHOICES Available: Yes,No;

ASK Pre_Skidv1: "Did Vehicle 1 leave tire marks prior to impact?";
CHOICES Pre_Skidv1: Yes,No;

ASK Pre_Skidv2: "Did Vehicle 2 leave tire marks prior to impact?";
CHOICES Pre_Skidv2: Yes,No;

ASK Type_Prev1: "Were they typical skid marks or yaw marks, OR a combination of skidding followed by yaw?
NOTE: Yaw marks are easy to differentiate from skid marks.
At the onset of yaw, the outer edge of the tire leaves a narrow dark mark that appears as a thin curving line about 5 cms in width and widens to that amount of tire tread in contact with the roadway as the vehicle goes into yaw. Lateral striation marks caused by the side of the tire tread are often visible.";
CHOICES Type_Prev1: Skid,Yaw,Skid_Yaw;

ASK Type_Prev2: "Were they typical skid marks or yaw marks, OR a combination of skidding followed by yaw?
NOTE: Yaw marks are easy to differentiate from skid marks.
At the onset of yaw, the outer edge of the tire leaves a narrow dark mark that appears as a thin curving line about 5 cms in width and widens to that amount of tire tread in contact with the roadway as the vehicle goes into yaw. Lateral striation marks caused by the side of the tire tread are often visible.";
CHOICES Type_Prev2: Skid,Yaw,Skid_Yaw;

ASK Surf_Postv1: "Are all 4 tires on the same surface for the first one-third of the yaw mark?";
CHOICES Surf_Postv1: Yes,No;

ASK Surf_Postv2: "Are all 4 tires on the same surface for the first one-third of the yaw mark?";
CHOICES Surf_Postv2: Yes,No;

ASK Surf_Prev1: "Are all 4 tires on the same surface for the first one-third of the yaw mark?";
CHOICES Surf_Prev1: Yes,No;

ASK Surf_Prev2: "Are all 4 tires on the same surface for the first one-third of the yaw mark?";
CHOICES Surf_Prev2: Yes,No;

ASK Flat_Postv1: "Was gravity constant for the duration of the yaw mark? (i.e., if the vehicle travelled up and over the crest of a hill while in yaw the vehicles weight (gravity) would not be constant for the entire yaw mark)";
CHOICES Flat_Postv1: Yes,No;

ASK Flat_Postv2: "Was gravity constant for the duration of the yaw mark? (i.e., if the vehicle travelled up and over the crest of a hill while in yaw the vehicles weight (gravity) would not be constant for the entire yaw mark)";
CHOICES Flat_Postv2: Yes,No;

ASK Flat_Prev1: "Was gravity constant for the duration of the yaw mark? (i.e., if the vehicle travelled up and over the crest of a hill while in yaw the vehicles weight (gravity) would not be constant for the entire yaw mark)";
CHOICES Flat_Prev1: Yes,No;

ASK Flat_Prev2: "Was gravity constant for the duration of the yaw mark? (i.e., if the vehicle travelled up and over the crest of a hill while in yaw the vehicles weight (gravity) would not be constant for the entire yaw mark)";
CHOICES Flat_Prev2: Yes,No;

ASK Post_Zerov1: "Can you assume the post-impact speed of Vehicle 1 approximates 0 km/h?";
CHOICES Post_Zerov1: Yes,No;

ASK Post_Zerov2: "Can you assume the post-impact speed of Vehicle 2 approximates 0 km/h?";
CHOICES Post_Zerov2: Yes,No;

!------------------- Choose_Method Queries ------------------------

ASK Select_Postv1: "Which method(s) would you like to use to determine the POST-IMPACT speed of VEHICLE 1?
NOTE: If the vehicle yawed and/or vaulted during this sequence, disregard any subsequent events. Yaw and vault give an estimate of speed at the beginning of the event whereas the other techniques give the change in speed over the duration of the event.
(You can select more than one action, press the END key to continue)";

CHOICES Select_Postv1: Skid_Analysis,Yaw,Roll,Overturn,Vault,Witness,None;

ASK Select_Postv2: "Which method(s) would you like to use to determine the POST-IMPACT speed of VEHICLE 2?
NOTE: If the vehicle yawed and/or vaulted during this sequence, disregard any subsequent events. Yaw and vault give an estimate of speed at the beginning of the event whereas the other techniques give the change in speed over the duration of the event.
(You can select more than one action, press the END key to continue)";

CHOICES Select_Postv2: Skid_Analysis,Yaw,Roll,Overturn,Vault,Witness,None;

ASK Select_Impact: "Which method would you like to use to determine the IMPACT speeds of both vehicles?
(You can select more than one action, press the END key to continue)";
CHOICES Select_Impact: Momentum,Energy,None;

ASK Select_Prev1: "Which method would you like to use to determine the PRE-IMPACT speed of VEHICLE 1?
NOTE: If skidding preceeded yaw then select both of these options.
(Press the END key to continue)
";
CHOICES Select_Prev1: Skid_Analysis,Yaw,Witness,None;

ASK Select_Prev2: "Which method would you like to use to determine the PRE-IMPACT speed of VEHICLE 2?
NOTE: If skidding preceeded yaw then select both of these options.
(Press the END key to continue)
";
CHOICES Select_Prev2: Skid_Analysis,Yaw,Witness,None;

!---------------------- Plural Variables ----------------------------

PLURAL:
Method_Postv1,Method_Postv2,Method_Impact,Method_Prev1,Method_Prev2, Post_Actv1,Post_Actv2;

PLURAL: Select_Postv1,Select_Postv2,Select_Impact,Select_Prev1,Select_Prev2;

APPENDIX C. Partial Listing of Module 2

```
!==========================================================
!                    KAR2 Module
!==========================================================

EXECUTE;
BKCOLOR = 3;
RUNTIME;
ENDOFF;

ACTIONS
        LOADFACTS karvalue
        WOPEN 9,0,0,11,79,3
        ACTIVE 9
        FIND Post_Analysis1
        FIND Post_Analysis2
        FIND Imp_Analysis
        FIND Pre_Analysis1
        FIND Pre_Analysis2
        WOPEN 1,0,0,23,79,4
        WOPEN 6,1,1,21,77,3
        FIND Summary1
        DISPLAY "    Press any key to Continue.~"
        FIND Summary2
        DISPLAY "    Press any key to Continue.~"
        Warning = None                      ! Need this in case no warnings are
        FIND Warnings                       ! registered during consultation.
        DISPLAY "    Press any key to Quit.~"
        WCLOSE 1
        WCLOSE 6
        WCLOSE 2
        WCLOSE 4
        WCLOSE 9
        CLS;

!==========================================================
!                    Display Final Results
!==========================================================

RULE SUM1
IF      Method_Postv1 <> None OR
        Method_Impact <> None OR
        Method_Prev1 <> None
THEN
        ACTIVE 6
```

```
        DISPLAY "        VEHICLE 1
"
        Summary1 = Found;

RULE SUM2
IF      Postv1 <> UNKNOWN
THEN
        ACTIVE 6
        FORMAT Postv1,3.0
        DISPLAY "   CALCULATED: Post-Impact Speed = {Postv1} km/h"
        Summary1 = Found;

RULE SUM3
IF      Method_Postv1 = Witness
THEN
        ACTIVE 6
        DISPLAY "   WITNESS:   Post-Impact Speed =  {Post_Spv1} km/h"
        Summary1 = Found;

RULE SUM4
IF      Method_Postv1 = Assume
THEN
        ACTIVE 6
        DISPLAY "   ASSUMED:   Post-Impact Speed =  0 km/h"
        Summary1 = Found;

RULE SUM5
IF      Impv1 <> UNKNOWN
THEN
        ACTIVE 6
        FORMAT Impv1, 3.0
        FORMAT MomD_v1, 3.0
        DISPLAY
"       MOMENTUM: Impact speed     = {Impv1} km/h   DeltaV = {MomD_v1}
        km/h"
        Summary1 = Found;

RULE SUM6
IF      ImpEv1 <> UNKNOWN
THEN
        ACTIVE 6
        FORMAT ImpEv1, 3.0
        FORMAT Delta_v1, 3.0
        DISPLAY
"       ENERGY:   Impact speed     = {ImpEv1} km/h   DeltaV = {Delta_v1} km/h"
        Summary1 = Found;
```

```
RULE SUM7
IF      Prev1 <> UNKNOWN
THEN
        ACTIVE 6
        FORMAT Prev1, 3.0
        DISPLAY "    CALCULATED: Initial speed     = {Prev1} km/h"
        Summary1 = Found;

RULE SUM8
IF      PreYawv1 <> UNKNOWN
THEN
        ACTIVE 6
        FORMAT PreYawv1, 3.0
        DISPLAY "    * Yaw is not dependant on impact or post-impact calculations *
        YAW: Initial speed     = {PreYawv1} km/h"
        Summary1 = Found;

RULE SUM9
IF      Method_Prev1 = Witness
THEN
        ACTIVE 6
        DISPLAY "    WITNESS:    Initial speed     = {Pre_Spv1} km/h"
        Summary1 = Found;

RULE SUM10
IF      Method_Postv2 <> None OR
        Method_Impact <> None OR
        Method_Prev2 <> None
THEN
        ACTIVE 6
        DISPLAY "
        VEHICLE 2
"
        Summary2 = Found;

RULE SUM11
IF      Postv2 <> UNKNOWN
THEN
        ACTIVE 6
        FORMAT Postv2,3.0
        DISPLAY "    CALCULATED: Post-Impact Speed = {Postv2} km/h"
        Summary2 = Found;

RULE SUM12
IF      Method_Postv2 = Witness
THEN
        ACTIVE 6
        DISPLAY "    WITNESS:    Post-Impact Speed =  {Post_Spv2} km/h"
        Summary2 = Found;
```

```
RULE SUM13
IF      Method_Postv2 = Assume
THEN
        ACTIVE 6
        DISPLAY "    ASSUMED:   Post-Impact Speed =   0 km/h"
        Summary2 = Found;

RULE SUM14
IF      Impv2 <> UNKNOWN
THEN
        ACTIVE 6
        FORMAT Impv2, 3.0
        FORMAT MomD_v2, 3.0
        DISPLAY
"       MOMENTUM:  Impact speed      = {Impv2} km/h    DeltaV = {MomD_v2}
        km/h"
        Summary2 = Found;

RULE SUM15
IF      ImpEv2 <> UNKNOWN
THEN
        ACTIVE 6
        FORMAT ImpEv2, 3.0
        FORMAT Delta_v2, 3.0
        DISPLAY
"       ENERGY:    Impact speed      = {ImpEv2} km/h    DeltaV = {Delta_v2} km/h"
        Summary2 = Found;

RULE SUM16
IF      Prev2 <> UNKNOWN
THEN
        ACTIVE 6
        FORMAT Prev2, 3.0
        DISPLAY "    CALCULATED: Initial speed     = {Prev2} km/h"
        Summary2 = Found;

RULE SUM17
IF      PreYawv2 <> UNKNOWN
THEN
        ACTIVE 6
        FORMAT PreYawv2, 3.0
        DISPLAY "     * Yaw is not dependant on impact or post-impact calculations *
        YAW: Initial speed     = {PreYawv2} km/h"
        Summary2 = Found;

RULE SUM18
IF      Method_Prev2 = Witness
THEN
        ACTIVE 6
        DISPLAY "    WITNESS:   Initial speed     = {Pre_Spv2} km/h"
        Summary2 = Found;
```

```
RULE SUM20
IF      Warning = None                              ! Displays Warning title.
THEN
        ACTIVE 6
        DISPLAY "
        WARNINGS:"
        Warnings = Found;


RULE SUM21
IF      Warning = No_Impact
THEN
        ACTIVE 6
        DISPLAY "
* The speed losses during impact were not calculated. Initial speeds
* may be overly conservative."
        Warnings = Found;

RULE SUM22
IF      Warning = Momentum_Angles
THEN
        ACTIVE 6
        DISPLAY "
* Momentum Angles appear to be incorrect.  Check your convention."
        Warnings = Found;

RULE SUM23
IF      Warning = Crush_Measv1 AND
        Tran1 > (Transp1)
THEN
        ACTIVE 6
        DISPLAY "
* Vehicle 1's Delta_V calculated from the crush measurements does not
* coincide with Vehicle 1's pre and post-impact headings.  This is not
* due to an error in the PDOF.  Check crush measurements, these values are
* OVERESTIMATING Delta_V."
        Warnings = Found;

RULE SUM24
IF      Warning = Crush_Measv1 AND
        Tran1 < (Transp1)
THEN
        ACTIVE 6
        DISPLAY "
* Vehicle 1's Delta_V calculated from the crush measurements does not
* coincide with Vehicle 1's pre and post-impact headings.  This is not
* due to an error in the PDOF.  Check crush measurements, these values are
* UNDERESTIMATING Delta_V."
        Warnings = Found;
```

```
RULE SUM25
IF      Warning = Crush_Measv2 AND
        Tran2 > (Transp2)
THEN
        ACTIVE 6
        DISPLAY "
* Vehicle 2's Delta_V calculated from the crush measurements does not
* coincide with Vehicle 2's pre and post-impact headings. This is not
* due to an error in the PDOF. Check crush measurements, these values are
* OVERESTIMATING Delta_V."
        Warnings = Found;

RULE SUM26
IF      Warning = Crush_Measv2 AND
        Tran2 < (Transp2)
THEN
        ACTIVE 6
        DISPLAY "
* Vehicle 2's Delta_V calculated from the crush measurements does not
* coincide with Vehicle 2's pre and post-impact headings. This is not
* due to an error in the PDOF. Check crush measurements, these values are
* UNDERESTIMATING Delta_V."
        Warnings = Found;


!=============================================================
!               Rules to Print Titles and Call Speed Routines
!=============================================================


RULE TITLE1A                                    ! Checks if there are any
IF      Method_Postv1 = Skid_Analysis OR        ! post-impact speeds to
        Method_Postv1 = Roll OR                 ! calculate and dislays
        Method_Postv1 = Vault OR                ! them if necessary
        Method_Postv1 = Overturn OR
        Method_Postv1 = Yaw
THEN
        WOPEN 4,1,5,12,70,2
        ACTIVE 9
        FIND The_Post_Spds1
        ACTIVE 2
        CLS
        Pt1 = (Roll_Spdv1*Roll_Spdv1+O_Spdv1*O_Spdv1+Vlt_Spdv1*Vlt_Spdv1)
        Postv1=((@SQRT(Post_Sk_Spdv1*Post_Sk_Spdv1+Post_Yawv1
        *Post_Yawv1+Pt1))*3.6)
        FORMAT Postv1, 3.0
        DISPLAY"
                POST-IMPACT ANALYSIS
"
        DISPLAY "The post-impact speed of Vehicle 1 = {Postv1} km/h.
"
```

```
        DISPLAY "(Press any key to continue ....)~"
        CLS
        WCLOSE 2
        WCLOSE 4
        FIND Skid_Re                        ! Reset skid variables
        FIND Yaw_Re                         ! Reset yaw variables
        FIND Roll_Re                        ! Reset roll variables
        FIND Vlt_Re                         ! Reset vault variables
        FIND O_Re                           ! Reset overturn variables
        RESET Skid_Assgnr
        RESET Yaw_Assgnr
        RESET Roll_Assgnr
        RESET Vlt_Assgnr
        RESET O_Assgnr
        RESET Skid_Re
        RESET Yaw_Re
        RESET Roll_Re
        RESET Vlt_Re
        RESET O_Re
        RESET Pt1
        ACTIVE 9
        Post_Analysis1 = Found;

RULE TITLE1B
IF      Method_Postv1 = Witness AND
        Post_Spv1 <> UNKNOWN
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY"
                WITNESS STATEMENT
"
        DISPLAY "The post-impact speed of Vehicle 1 = {Post_Spv1} km/h.
"
        DISPLAY "(Press any key to continue ....)~"
        CLS
        WCLOSE 2
        ACTIVE 9
        CLS
        Post_Analysis1 = Found;


RULE TITLE2A                                    ! Checks if there are any
IF      Method_Postv2 = Skid_Analysis OR        ! post-impact speeds to
        Method_Postv2 = Roll OR                 ! calculate and dislays
        Method_Postv2 = Vault OR                ! them if necessary
        Method_Postv2 = Overturn OR
        Method_Postv2 = Yaw
THEN
        WOPEN 4,1,5,12,70,2
        ACTIVE 9
        FIND The_Post_Spds2
```

```
        ACTIVE 2
        CLS
        Pt2 = (Roll_Spdv2*Roll_Spdv2+O_Spdv2*O_Spdv2+Vlt_Spdv2*Vlt_Spdv2)
        Postv2=((@SQRT(Post_Sk_Spdv2*Post_Sk_Spdv2+Post_Yawv2
        *Post_Yawv2 +Pt2))*3.6)
        FORMAT Postv2, 3.0
        DISPLAY"
               POST-IMPACT ANALYSIS
"
        DISPLAY "The post-impact speed of Vehicle 2 = {Postv2} km/h.
"
        DISPLAY "(Press any key to continue ....)~"
        CLS
        WCLOSE 2
        WCLOSE 4
        FIND Skid_Re                    ! Reset skid variables
        FIND Yaw_Re                     ! Reset yaw variables
        FIND Roll_Re                    ! Reset roll variables
        FIND Vlt_Re                     ! Reset vault variables
        FIND O_Re                       ! Reset overturn variables
        RESET Skid_Assgnr
        RESET Yaw_Assgnr
        RESET Roll_Assgnr
        RESET Vlt_Assgnr
        RESET O_Assgnr
        RESET Skid_Re
        RESET Yaw_Re
        RESET Roll_Re
        RESET Vlt_Re
        RESET O_Re
        RESET Pt2
        ACTIVE 9
        Post_Analysis2 = Found;


RULE TITLE2B
IF      Method_Postv2 = Witness AND
        Post_Spv2 <> UNKNOWN
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY"
               WITNESS STATEMENT
"
        DISPLAY "The post-impact speed of Vehicle 2 = {Post_Spv2} km/h.
"
        DISPLAY "(Press any key to continue ....)~"
        CLS
        WCLOSE 2
        ACTIVE 9
        CLS
        Post_Analysis2 = Found;
```

```
                                              ! If Momentum is selected this
RULE TITLE3A                                  ! rule executes.
IF      Method_Impact = Momentum AND
        Method_Postv1 <> None AND             ! Fails if user selects Momentum
        Method_Postv2 <> None                 !but doesn't calc. post-impact
THEN                                          ! speeds.
        WOPEN 4,1,5,12,70,2
        ACTIVE 9
        FIND Corr_Pst_Speed1
        FIND Corr_Pst_Speed2
        FIND The_Imp_Spds1
        ACTIVE 2
        CLS
        Impv1 = (Imp_Speedv1*3.6)
        Impv2 = (Imp_Speedv2*3.6)
        FORMAT Impv1, 3.0
        FORMAT Impv2, 3.0
        DISPLAY "
                MOMENTUM  ANALYSIS
"
        DISPLAY "The impact speed of Vehicle 1 = {Impv1} km/h
The impact speed of Vehicle 2 = {Impv2} km/h"
        DISPLAY "(Press any key to continue ....)~"
        CLS
        WCLOSE 2
        WCLOSE 4
        RESET Mom_Assgnr
        ACTIVE 9
        Imp_Analysis = Found;

                                              ! If Energy is selected this rule
RULE TITLE3B                                  ! is executed.
IF      Method_Impact = Energy AND
        Method_Postv1 <> None AND             ! Fails if user selects energy
        Method_Postv2 <> None                 ! but doesn't calc post-impact
THEN                                          ! speeds.
        WOPEN 4,1,5,12,70,2
        ACTIVE 9
        FIND Corr_Pst_Speed1
        FIND Corr_Pst_Speed2
        FIND The_Imp_Spds2
        ACTIVE 2
        CLS
        ImpEv1 = (Imp_E_Speedv1*3.6)
        ImpEv2 = (Imp_E_Speedv2*3.6)
        FORMAT ImpEv1, 3.0
        FORMAT ImpEv2, 3.0
        DISPLAY "
                ENERGY  ANALYSIS
"
        DISPLAY "The impact speed of Vehicle 1 = {ImpEv1} km/h
```

```
The impact speed of Vehicle 2 = {ImpEv2} km/h"
        DISPLAY "(Press any key to continue ....)~"
        CLS
        WCLOSE 2
        WCLOSE 4
        RESET Ene_Assgnr
        ACTIVE 9
        Imp_Analysis = Found;


RULE TITLE4A
IF      Method_Prev1 = Skid_Analysis AND
        Method_Prev1 <> Yaw
THEN
        WOPEN 4,1,5,12,70,2
        ACTIVE 9
        FIND Corr_Pst_Speed1                    ! Check that these have been
        FIND Corr_Pst_Speed2                    ! initialized in case the impact
        FIND Corr_Imp_Speed                     ! routines were not called, hence
        CLS
        FIND The_Pre_Spds1                      ! a post speed could not be passed
        ACTIVE 2                                ! to this stage for an estimate
        CLS
        Prev1 = ((@SQRT(Pre_Speedv1*Pre_Speedv1+Imp_Speedv1*Imp_Speedv1))
        *3.6)
        FORMAT Prev1, 3.0
        DISPLAY "
                PRE-IMPACT  ANALYSIS
"
        DISPLAY "The initial speed of Vehicle 1 = {Prev1} km/h.
"
        DISPLAY "(Press any key to continue ....)~"
        CLS
        WCLOSE 2
        WCLOSE 4
        FIND Skid_Re
        RESET Skid_Assgnr
        RESET Skid_Re
        ACTIVE 9
        Pre_Analysis1 = Found;


RULE TITLE4B                                    ! Initial speed estimate
IF      Method_Prev1 = Yaw AND                  ! from yaw directly.
        Method_Prev1 <> Skid_Analysis           ! In case veh skidded prior to
THEN                                            ! yaw, sent to next rule to
        WOPEN 4,1,5,12,70,2                     ! combine speeds
        ACTIVE 9
        FIND The_Pre_Spds1
        ACTIVE 2
        CLS
        PreYawv1 = (Pre_Yawv1*3.6)
```

```
        FORMAT PreYawv1, 3.0
        DISPLAY"
                METHOD - YAW ANALYSIS
"
        DISPLAY "The initial speed of Vehicle 1 = {PreYawv1} km/h.
"
        DISPLAY "(Press any key to continue ....)~"
        CLS
        WCLOSE 2
        WCLOSE 4
        FIND Yaw_Re
        RESET Yaw_Assgnr
        RESET Yaw_Re
        ACTIVE 9
        Pre_Analysis1 = Found;


RULE TITLE4C                                    ! If veh skidded and then
IF      Method_Prev1 = Yaw AND                  ! went into yaw. Gives
        Method_Prev1 = Skid_Analysis            ! initial speed directly
THEN
        WOPEN 4,1,5,12,70,2
        ACTIVE 9
        FIND The_Pre_Spds1
        ACTIVE 2
        CLS
        PreYawv1 = ((@SQRT(Pre_Speedv1*Pre_Speedv1+Pre_Yawv1*Pre_Yawv1))
        *3.6)
        FORMAT PreYawv1, 3.0
        DISPLAY"
                METHOD - SKID/YAW ANALYSIS
"
        DISPLAY "The initial speed of Vehicle 1 = {PreYawv1} km/h.
"
        DISPLAY "(Press any key to continue ....)~"
        CLS
        WCLOSE 2
        WCLOSE 4
        FIND Yaw_Re
        FIND Skid_Re
        RESET Yaw_Assgnr
        RESET Skid_Assgnr
        RESET Yaw_Re
        RESET Skid_Re
        ACTIVE 9
        Pre_Analysis1 = Found;
```

```
RULE TITLE4D                                   ! Displays witness estimate of
IF      Method_Prev1 = Witness AND             ! V1's pre-impact speed.
        Pre_Spv1 <> UNKNOWN
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY"
                METHOD - WITNESS STATEMENT
"
        DISPLAY "The initial speed of Vehicle 1 = {Pre_Spv1} km/h.
"
        DISPLAY "(Press any key to continue ....)~"
        CLS
        WCLOSE 2
        ACTIVE 9
        CLS
        Pre_Analysis1 = Found;


RULE TITLE5A
IF      Method_Prev2 = Skid_Analysis AND
        Method_Prev2 <> Yaw
THEN
        WOPEN 4,1,5,12,70,2
        ACTIVE 9
        FIND Corr_Pst_Speed1
        FIND Corr_Pst_Speed2
        FIND Corr_Imp_Speed
        CLS
        FIND The_Pre_Spds2
        ACTIVE 2
        CLS
        Prev2 = ((@SQRT(Pre_Speedv2*Pre_Speedv2+Imp_Speedv2*Imp_Speedv2))
        *3.6)
        FORMAT Prev2, 3.0
        DISPLAY "
                PRE-IMPACT ANALYSIS
"
        DISPLAY "The initial speed of Vehicle 2 = {Prev2} km/h.
"
        DISPLAY "(Press any key to continue ....)~"
        CLS
        WCLOSE 2
        WCLOSE 4
        FIND Skid_Re
        RESET Skid_Assgnr
        RESET Skid_Re
        ACTIVE 9
        Pre_Analysis2 = Found;
```

```
RULE TITLE5B
IF      Method_Prev2 = Yaw AND
        Method_Prev2 <> Skid_Analysis
THEN
        WOPEN 4,1,5,12,70,2
        ACTIVE 9
        FIND The_Pre_Spds2
        ACTIVE 2
        CLS
        PreYawv2 = (Pre_Yawv2*3.6)
        FORMAT PreYawv2, 3.0
        DISPLAY"
               METHOD - YAW ANALYSIS
"
        DISPLAY "The initial speed of Vehicle 2 = {PreYawv2} km/h.
"
        DISPLAY "(Press any key to continue ....)~"
        CLS
        WCLOSE 2
        WCLOSE 4
        FIND Yaw_Re
        RESET Yaw_Assgnr
        RESET Yaw_Re
        ACTIVE 9
        Pre_Analysis2 = Found;


RULE TITLE5C                                   ! If veh skidded and then
IF      Method_Prev2 = Yaw AND                 ! went into yaw. Gives
        Method_Prev2 = Skid_Analysis           ! initial speed directly
THEN
        WOPEN 4,1,5,12,70,2
        ACTIVE 9
        FIND The_Pre_Spds2
        ACTIVE 2
        CLS
        PreYawv2 = ((@SQRT(Pre_Speedv2*Pre_Speedv2+Pre_Yawv2*Pre_Yawv2))
        *3.6)
        FORMAT PreYawv2, 3.0
        DISPLAY"
               METHOD - SKID/YAW ANALYSIS
"
        DISPLAY "The initial speed of Vehicle 2 = {PreYawv2} km/h.
"
        DISPLAY "(Press any key to continue ....)~"
        CLS
        WCLOSE 2
        WCLOSE 4
        FIND Yaw_Re
        FIND Skid_Re
        RESET Yaw_Assgnr
        RESET Skid_Assgnr
```

```
        RESET Yaw_Re
        RESET Skid_Re
        ACTIVE 9
        Pre_Analysis2 = Found;

RULE TITLE5D                                    ! Displays witness estimate of
IF      Method_Prev2 = Witness AND              !V2's pre-impact speed.
        Pre_Spv2 <> UNKNOWN
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY"
              METHOD - WITNESS STATEMENT
"
        DISPLAY "The initial speed of Vehicle 2 = {Pre_Spv2} km/h.
"
        DISPLAY "(Press any key to continue ....)~"
        CLS
        WCLOSE 2
        ACTIVE 9
        CLS
        Pre_Analysis2 = Found;


!==============================================================
!             Rules to Select the Post-Impact Speed That Will be Sent
!             To the Impact Speed Routines
!==============================================================


RULE SELECT1                                    ! A calculated speed is
IF      Method_Postv1 = Skid_Analysis OR        ! preferred over a
        Method_Postv1 = Roll OR                 ! witness estimate, hence
        Method_Postv1 = Vault OR                ! it is used for impact
        Method_Postv1 = Overturn OR             ! calculations if available
        Method_Postv1 = Yaw
THEN
        Post_Speedv1 = (Postv1/3.6)
        Corr_Pst_Speed1 = Found;

RULE SELECT3
IF      Method_Postv1 = Witness                 ! Witness estimate used only
THEN                                            ! if post-impact speed calc's
        Post_Speedv1 = (Post_Spv1/3.6)          ! are impossible.
        Corr_Pst_Speed1 = Found;

RULE SELECT4
IF      Method_Postv2 = Skid_Analysis OR
        Method_Postv2 = Roll OR
        Method_Postv2 = Vault OR
        Method_Postv2 = Overturn OR
        Method_Postv2 = Yaw
```

```
THEN
        Post_Speedv2 = (Postv2/3.6)
        Corr_Pst_Speed2 = Found;

RULE SELECT6
IF      Method_Postv2 = Witness
THEN
        Post_Speedv2 = (Post_Spv2/3.6)
        Corr_Pst_Speed2 = Found;


!=========================================================
!               Rules to Select the Impact Speed That Will be Sent
!               To the Pre-Impact Speed Routines
!=========================================================


RULE SELECT7
IF      Method_Impact = Momentum AND
        Method_Impact = Energy
THEN
        DISPLAY "Both Momentum and Energy methods were used to calculate
the impact speeds:
MOMENTUM - Impact Speed V1 = {Impv1} km/h  Impact Speed V2 = {Impv2} km/h
ENERGY   - Impact Speed V1 = {ImpEv1} km/h  Impact Speed V2 = {ImpEv2} km/h"
        FIND Mom_Ene
        Corr_Imp_Speed = Found;

RULE CHOOSE7
IF      Choose_ME = Momentum
THEN
        Imp_Speedv1 = (Imp_M_Speedv1)
        Imp_Speedv2 = (Imp_M_Speedv2)
        Mom_Ene = Found
ELSE
        Imp_Speedv1 = (Imp_E_Speedv1)
        Imp_Speedv2 = (Imp_E_Speedv2)
        Mom_Ene = Found;

RULE SELECT8
IF      Method_Impact = Momentum
THEN
        Imp_Speedv1 = (Imp_M_Speedv1)
        Imp_Speedv2 = (Imp_M_Speedv2)
        Corr_Imp_Speed = Found;

RULE SELECT9
IF      Method_Impact = Energy
THEN
        Imp_Speedv1 = (Imp_E_Speedv1)
        Imp_Speedv2 = (Imp_E_Speedv2)
        Corr_Imp_Speed = Found;
```

```
RULE SELECT10                                ! This rule will pass the
IF      Method_Impact = None                 ! post-impact speed to the
                                             ! pre-impact routines if the
THEN                                         ! speed loss during impact
        Imp_Speedv1 = (Post_Speedv1)         ! cannot be determined
        Imp_Speedv2 = (Post_Speedv2)         ! Gives a conservative result
        Warning = No_Impact                  ! and a warning is issued
        Corr_Imp_Speed = Found;


!==============================================================
!       Rules That Direct Program to Find Speeds
!==============================================================


RULE 1A                                      ! Finds V1 post-impact speed from a
IF      Method_Postv1 = Skid_Analysis        ! skid analysis
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "The questions being asked are related
to the POST_IMPACT phase of VEHICLE 1.
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND Post_Sk_Spdv1
        CLS
        FIND Skid_Assgnr
        The_Post_Spds1 = Found;

RULE 1B                                      ! Finds V1 post-impact speed from a
IF      Method_Postv1 = Yaw                  ! yaw analysis
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "The questions being asked are related
to the POST_IMPACT phase of VEHICLE 1.
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND Post_Yawv1
        CLS
        FIND Yaw_Assgnr
        The_Post_Spds1 = Found;

RULE 1C
IF      Method_Postv1 = Roll
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "The questions being asked are related
to the POST_IMPACT phase of VEHICLE 1.
```

```
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND Roll_Spdv1
        CLS
        FIND Roll_Assgnr
        The_Post_Spds1 = Found;

RULE 1D
IF      Method_Postv1 = Vault
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "The questions being asked are related
to the POST_IMPACT phase of VEHICLE 1.
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND Vlt_Spdv1
        CLS
        FIND Vlt_Assgnr
        The_Post_Spds1 = Found;

RULE 1E
IF      Method_Postv1 = Overturn
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "The questions being asked are related
to the POST_IMPACT phase of VEHICLE 1.
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND O_Spdv1
        CLS
        FIND O_Assgnr
        The_Post_Spds1 = Found;

RULE 2A                                  ! Finds V2 post-impact speed from a
IF      Method_Postv2 = Skid_Analysis    ! skid analysis
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "The questions being asked are related
to the POST_IMPACT phase of VEHICLE 2.
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND Post_Sk_Spdv2
        CLS
        FIND Skid_Assgnr
        The_Post_Spds2 = Found;
```

```
RULE 2B                                  ! Finds V2 post-impact speed from a
IF      Method_Postv2 = Yaw              ! yaw analysis
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "The questions being asked are related
to the POST_IMPACT phase of VEHICLE 2.
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND Post_Yawv2
        CLS
        FIND Yaw_Assgnr
        The_Post_Spds2 = Found;

RULE 2C
IF      Method_Postv2 = Roll
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "The questions being asked are related
to the POST_IMPACT phase of VEHICLE 2.
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND Roll_Spdv2
        CLS
        FIND Roll_Assgnr
        The_Post_Spds2 = Found;

RULE 2D
IF      Method_Postv2 = Vault
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "The questions being asked are related
to the POST_IMPACT phase of VEHICLE 2.
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND Vlt_Spdv2
        CLS
        FIND Vlt_Assgnr
        The_Post_Spds2 = Found;

RULE 2E
IF      Method_Postv2 = Overturn
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "The questions being asked are related
to the POST_IMPACT phase of VEHICLE 2.
        (Press any key to continue ....)~"
        ACTIVE 9
```

```
        FIND O_Spdv2
        CLS
        FIND O_Assgnr
        The_Post_Spds2 = Found;

RULE 3A
IF      Method_Impact = Momentum
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "MOMENTUM CALCULATIONS - The questions being asked
are related to the impact between both vehicles.
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND Mom_Speed
        CLS
        FIND Mom_Assgnr
        The_Imp_Spds1 = Found;

RULE 3B
IF      Method_Impact = Energy
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "ENERGY CALCULATIONS - The questions being asked
are related to the impact between both vehicles.
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND Ene_Speed
        CLS
        FIND Ene_Assgnr
        The_Imp_Spds2 = Found;

RULE 4A
IF      Method_Prev1 = Skid_Analysis
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "The questions being asked are related
to the PRE-IMPACT phase of VEHICLE 1.
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND Pre_Speedv1
        CLS
        FIND Skid_Assgnr
        The_Pre_Spds1 = Found;

RULE 4B                             ! Finds V1 pre-impact speed from a
IF      Method_Prev1 = Yaw          ! yaw analysis
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
```

```
        DISPLAY "The questions being asked are related
to the PRE_IMPACT phase of VEHICLE 1.
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND Pre_Yawv1
        CLS
        FIND Yaw_Assgnr
        The_Pre_Spds1 = Found;

RULE 5A
IF      Method_Prev2 = Skid_Analysis
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "The questions being asked are related
to the PRE_IMPACT phase of VEHICLE 2.
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND Pre_Speedv2
        CLS
        FIND Skid_Assgnr
        The_Pre_Spds2 = Found;

RULE 5B                                 ! Finds V2 pre-impact speed from a
IF      Method_Prev2 = Yaw              ! yaw analysis and displays speed.
THEN
        WOPEN 2,14,10,6,60,4
        ACTIVE 2
        DISPLAY "The questions being asked are related
to the PRE_IMPACT phase of VEHICLE 2.
        (Press any key to continue ....)~"
        ACTIVE 9
        FIND Pre_Yawv2
        CLS
        FIND Yaw_Assgnr
        The_Pre_Spds2 = Found;


!==============================================================
!                  Rules to Display Parameters Used in
!                  Calculations
!==============================================================


RULE PARAM1
IF      Mult_Surface = No
THEN
        ACTIVE 4
        DISPLAY"                    SKID PARAMETERS
  Braking Efficiency = {Brake_Eff}  Skid Distance = {Skid_Dist} (meters)
                Drag Factor = {Total_Drag}"
        Skid_Assgnr = Found;
```

```
RULE PARAM2
IF      Mult_Surface = Yes AND
        Num_Surfaces = Two AND
        Diff_Surfaces2 = Continuous
THEN
        ACTIVE 4
        DISPLAY"                SKID PARAMETERS
                Braking Efficiency = {Brake_Eff}
Skid Distance 1 = {Skid_Dist1} (meters), Skid Distance 2 = {Skid_Dist2} (meters)
        Drag Factor(1) = {Tot_Drag1}  Drag Factor(2) = {Tot_Drag2}"
        Skid_Assgnr = Found;

RULE PARAM3
IF      Mult_Surface = Yes AND
        Num_Surfaces = Two AND
        Diff_Surfaces2 = Split_Sides
THEN
        ACTIVE 4
        DISPLAY"                SKID PARAMETERS
  Braking Efficiency = {Brake_Eff}   Skid Distance = {Skid_Dist} (meters)
        Drag Factor(1) = {Tot_Drag1}  Drag Factor(2) = {Tot_Drag2}"
        Skid_Assgnr = Found;

RULE PARAM4
IF      Mult_Surface = Yes AND
        Num_Surfaces = Two AND
        Diff_Surfaces2 = Combination
THEN
        ACTIVE 4
        DISPLAY"                SKID PARAMETERS
                Braking Efficiency = {Brake_Eff}
Skid Distance 1 = {Skid_DistA} (meters), Skid Distance 2 = {Skid_DistB} (meters)
        Drag Factor(1) = {Tot_Drag1}  Drag Factor(2) = {Tot_Drag2}"
        Skid_Assgnr = Found;

RULE PARAM5
IF      Mult_Surface = Yes AND
        Num_Surfaces = Three AND
        Diff_Surfaces3 = Continuous
THEN
        ACTIVE 4
        DISPLAY"                SKID PARAMETERS
 Braking Efficiency = {Brake_Eff}   Skid Distance 1 = {Skid_Dist1} (meters)
Skid Distance 2 = {Skid_Dist2} (meters), Skid Distance 3 = {Skid_Dist3} (meters)
        Drag Factor(1) = {Tot_Drag1}  Drag Factor(2) = {Tot_Drag2}
                Drag Factor(3) = {Tot_Drag3}"
        Skid_Assgnr = Found;
```

```
RULE PARAM6
IF      Mult_Surface = Yes AND
        Num_Surfaces = Three AND
        Diff_Surfaces3 = Combination
THEN
        ACTIVE 4
        DISPLAY"                    SKID PARAMETERS
                Braking Efficiency = {Brake_Eff}
 Skid Distance 1 = {Skid_DistA} (meters), Skid Distance 2 = {Skid_DistB} (meters)
         Drag Factor(1) = {Tot_Drag1}  Drag Factor(2) = {Tot_Drag2}
                Drag Factor(3) = {Tot_Drag3}"
        Skid_Assgnr = Found;

RULE PARAM7
IF      Radius = Value AND
        Super = No
THEN
        ACTIVE 4
        DISPLAY"                    YAW PARAMETERS
        Radius = {Rad} (meters)     Drag Factor = {Total_Drag}"
        Yaw_Assgnr = Found;

RULE PARAM8
IF      Radius = Value AND
        Super = Yes
THEN
        ACTIVE 4
        DISPLAY"                    YAW PARAMETERS
        Radius = {Rad} (meters)    Superelevation = {Sup_Elev} (%)
                Drag Factor = {Total_Drag}"
        Yaw_Assgnr = Found;

RULE PARAM9
IF      Radius = Calculate AND
        Super = No
THEN
        ACTIVE 4
        DISPLAY"                    YAW PARAMETERS
   Chord Length = {Chord} (meters)  Middle Ordinate = {Middle_Ord} (meters)
                Drag Factor = {Total_Drag}"
        Yaw_Assgnr = Found;

RULE PARAM10
IF      Radius = Calculate AND
        Super = Yes
THEN
        ACTIVE 4
        DISPLAY"                    YAW PARAMETERS
   Chord Length = {Chord} (meters)  Middle Ordinate = {Middle_Ord} (meters)
        Superelevation = {Sup_Elev} (%)   Drag Factor = {Total_Drag}"
        Yaw_Assgnr = Found;
```

```
RULE PARAM11
IF      Roll_Dist <> UNKNOWN
THEN
        ACTIVE 4
        DISPLAY"                 ROLL PARAMETERS
                Roll Distance = {Roll_Dist} (meters)"
        Roll_Assgnr = Found;

RULE PARAM12
IF      Take_Off = No
THEN
        ACTIVE 4
        DISPLAY"                VAULT PARAMETERS
 Horizontal Distance = {Horz_Dist} (meters) Vertical Fall = {Vert_Fall} (meters)"
        Vlt_Assgnr = Found;

RULE PARAM13
IF      Take_Off = Yes
THEN
        ACTIVE 4
        DISPLAY"                VAULT PARAMETERS
 Horizontal Distance = {Horz_Dist} (meters) Vertical Fall = {Vert_Fall} (meters)
                 Take-off Slope = {Take_Elev} (%)"
        Vlt_Assgnr = Found;

RULE PARAM14
IF      Over_Dist <> UNKNOWN
THEN
        ACTIVE 4
        DISPLAY"               OVERTURN PARAMETERS
  Overturn Distance = {Over_Dist} (meters)   Drag Factor = {Over_Drag}"
        O_Assgnr = Found;

RULE PARAM15
IF      Mom_Speed <> UNKNOWN
THEN
        ACTIVE 4
        DISPLAY"               MOMENTUM PARAMETERS
        Vehicle 1               Vehicle 2
 Approach Angle V1 = {Theta1} (degs)   Approach Angle V2 = {Phi1} (degs)
 Departure Angle V1 = {Theta2} (degs)  Departure Angle V2 = {Phi2} (degs)
 Mass V1 = {M1} (kg)           Mass V2 = {M2} (kg)"
        Mom_Assgnr = Found;

RULE PARAM16
IF      Energy_Meth = CRASH_Values
THEN
        WOPEN 4,4,5,8,70,2
        ACTIVE 4
        DISPLAY"               ENERGY PARAMETERS
        Vehicle 1               Vehicle 2
```

```
Velocity Change V1 = {Delta_v1} (km/h) Velocity Change V2 = {Delta_v2} (km/h)
 PDOF V1 = {Gamma_v1} (degs)      PDOF V2 = {Gamma_v2} (degs)
 Approach Angle V1 = {Theta1} (degs)    Approach Angle V2 = {Phi1} (degs)
 Departure Angle V1 = {Theta2} (degs)  Departure Angle V2 = {Phi2} (degs)"
        Ene_Assgnr = Found;

RULE PARAM17
IF      Energy_Meth = KAR_Estimate AND
        Num_Measv1 = Six OR
        Num_Measv2 = Six
THEN
        ACTIVE 4
        FORMAT Gamma_v1, 3.0
        FORMAT Gamma_v2, 3.0
        DISPLAY"                 ENERGY PARAMETERS
         Vehicle 1               Vehicle 2
 PDOF V1 = {Gamma_v1} (degs)      PDOF V2 = {Gamma_v2} (degs)
 Approach Angle V1 = {Theta1} (degs)    Approach Angle V2 = {Phi1} (degs)
 Departure Angle V1 = {Theta2} (degs)  Departure Angle V2 = {Phi2} (degs)
 Mass V1 = {M1} (kg)             Mass V2 = {M2} (kg)
 Wheelbase V1 = {Wbase_v1} (cms)          Wheelbase V2 = {Wbase_v2} (cms)
 Collision Surface = {Coll_Surfv1}  Collision Surface = {Coll_Surfv2}
 Crush Length V1 = {Len_v1} (cms)     Crush Length V2 = {Len_v2} (cms)
 C1 = {C1V1}, C2 = {C2V1}, C3 = {C3V1}          C1 = {C1V2}, C2 = {C2V2}, C3
= {C3V2}
 C4 = {C4V1}, C5 = {C5V1}, C6 = {C6V1}          C4 = {C4V2}, C5 = {C5V2}, C6
= {C6V2}"
        Ene_Assgnr = Found;

RULE PARAM18
IF      Energy_Meth = KAR_Estimate AND
        Num_Measv1 = Four OR
        Num_Measv2 = Four
THEN
        ACTIVE 4
        FORMAT Gamma_v1, 3.0
        FORMAT Gamma_v2, 3.0
        DISPLAY"                 ENERGY PARAMETERS
        Vehicle 1                Vehicle 2
 PDOF V1 = {Gamma_v1} (degs)      PDOF V2 = {Gamma_v2} (degs)
 Approach Angle V1 = {Theta1} (degs)    Approach Angle V2 = {Phi1} (degs)
 Departure Angle V1 = {Theta2} (degs)  Departure Angle V2 = {Phi2} (degs)
 Mass V1 = {M1} (kg)             Mass V2 = {M2} (kg)
 Wheelbase V1 = {Wbase_v1} (cms)          Wheelbase V2 = {Wbase_v2} (cms)
 Collision Surface = {Coll_Surfv1}  Collision Surface = {Coll_Surfv2}
 Crush Length V1 = {Len_v1} (cms)     Crush Length V2 = {Len_v2} (cms)
 C1 = {C1V1}, C2 = {C2V1}                  C1 = {C1V2}, C2 = {C2V2}
 C3 = {C3V1}, C4 = {C4V1}                  C3 = {C3V2}, C4 = {C4V2}"
        Ene_Assgnr = Found;
```

```
RULE PARAM19
IF      Energy_Meth = KAR_Estimate AND
        Num_Measv1 = Two OR
        Num_Measv2 = Two
THEN
        ACTIVE 4
        FORMAT Gamma_v1, 3.0
        FORMAT Gamma_v2, 3.0
        DISPLAY"                ENERGY PARAMETERS
        Vehicle 1               Vehicle 2

 PDOF V1 = {Gamma_v1} (degs)       PDOF V2 = {Gamma_v2} (degs)
 Approach Angle V1 = {Theta1} (degs)    Approach Angle V2 = {Phi1} (degs)
 Departure Angle V1 = {Theta2} (degs)  Departure Angle V2 = {Phi2} (degs)
 Mass V1 = {M1} (kg)              Mass V2 = {M2} (kg)
 Wheelbase V1 = {Wbase_v1} (cms)          Wheelbase V2 = {Wbase_v2} (cms)
 Collision Surface = {Coll_Surfv1}  Collision Surface = {Coll_Surfv2}
 Crush Length V1 = {Len_v1} (cms)      Crush Length V2 = {Len_v2} (cms)
 C1 = {C1V1}, C2 = {C2V1}               C1 = {C1V2}, C2 = {C2V2}"
        Ene_Assgnr = Found;


!=============================================================
!               Rules to Reset Variables Used in
!               Skid and Yaw Calculations
!=============================================================


RULE SKIDRESET                          ! Resets skid parameters
IF      Solution <> UNKNOWN             ! Dummy statement (always true)
THEN
        RESET Mult_Surface
        RESET Num_Surfaces
        RESET Diff_Surfaces2
        RESET Diff_Surfaces3
        RESET Brake_Eff
        RESET Skid_Dist
        RESET Skid_Dist1
        RESET Skid_Dist2
        RESET Skid_Dist3
        RESET Skid_DistA
        RESET Skid_DistB
        RESET Total_Drag
        RESET Tot_Drag1
        RESET Tot_Drag2
        RESET Tot_Drag3
        Skid_Re = Found;
```

```
RULE YAWRESET
IF      Solution <> UNKNOWN
THEN
        RESET Radius
        RESET Rad
        RESET Super
        RESET Total_Drag
        RESET Sup_Elev
        RESET Chord
        RESET Middle_Ord
        Yaw_Re = Found;

RULE ROLLRESET
IF      Solution <> UNKNOWN
THEN
        RESET Roll_Dist
        Roll_Re = Found;

RULE VLTRESET
IF      Solution <> UNKNOWN
THEN
        RESET Horz_Dist
        RESET Vert_Fall
        RESET Take_Off
        RESET Take_Elev
        Vlt_Re = Found;

RULE ORESET
IF      Solution <> UNKNOWN
THEN
        RESET Over_Dist
        RESET Over_Drag
        O_Re = Found;


!=============================================================
!             Rules to determine Post-Impact Speed
!             From Skid Marks
!=============================================================


RULE POSTSKID1                              ! Rule calls general rule post-speed
IF      Brake_Eff <> UNKNOWN AND            ! and assigns speed to V1
        Post_Speed <> UNKNOWN
THEN
        Post_Sk_Spdv1 = (Post_Speed)
        RESET Post_Speed;
```

```
RULE POSTSKID2                                  ! Rule calls general rule post-speed
IF     Brake_Eff <> UNKNOWN AND                 ! and assigns speed to V2
       Post_Speed <> UNKNOWN
THEN
       Post_Sk_Spdv2 = (Post_Speed)
       RESET Post_Speed;

RULE POSTSKID3A
IF     Mult_Surface = No AND                    ! Check if surface is continuous.
       Skid_Dist <> UNKNOWN AND
       Total_Drag <> UNKNOWN
THEN
       Post_Speed = (@SQRT(19.6 * Total_Drag * Skid_Dist *
       Brake_Eff));

RULE POSTSKID3B                                 ! Two surfaces
IF     Mult_Surface = Yes AND
       Num_Surfaces = Two AND
       Diff_Surfaces2 = Continuous AND          ! All four tires on each surface
       Skid_Dist1 <> UNKNOWN AND                ! at a time (except during transition
       Skid_Dist2 <> UNKNOWN                    ! between surfaces of course).
THEN
       CLS
       DISPLAY "The following questions are related to the first
surface the vehicle slid over during the post-impact skid.
       (Press any key to continue ....)~"
       CLS
       FIND Tot_Drag1
       CLS
       DISPLAY "The following questions are related to the second
surface the vehicle slid over during the post-impact skid.
       (Press any key to continue ....)~"
       CLS
       FIND Tot_Drag2
       Post_Speed = (@SQRT(19.6*(Tot_Drag1*Skid_Dist1 + Tot_Drag2*
       Skid_Dist2)*Brake_Eff));

RULE POSTSKID3C                                 ! Two surfaces
IF     Mult_Surface = Yes AND
       Num_Surfaces = Two AND
       Diff_Surfaces2 = Split_Sides AND         ! Each side on a different
       Skid_Dist <> UNKNOWN                     ! surface.
THEN
       CLS
       DISPLAY "The following questions are related to the surface
on the vehicles left side during the post-impact skid.
       (Press any key to continue ....)~"
       CLS
       FIND Tot_Drag1
       CLS
       DISPLAY "The following questions are related to the surface
on the vehicles right side during the post-impact skid.
```

```
        (Press any key to continue ....)~"
        CLS
        FIND Tot_Drag2
        Post_Speed = (@SQRT(9.8*(Tot_Drag1 + Tot_Drag2)*Skid_Dist*Brake_Eff));

RULE POSTSKID3D
IF      Mult_Surface = Yes AND              ! Combination of continuous surface
        Num_Surfaces = Two AND
        Diff_Surfaces2 = Combination AND    ! skid followed by a split surface
        Skid_DistA <> UNKNOWN AND           ! skid.
        Skid_DistB <> UNKNOWN
THEN
        CLS
        DISPLAY "The following questions are related to the homogeneous
surface the vehicle slid over during the post-impact skid.
        (Press any key to continue ....)~"
        CLS
        FIND Tot_Drag1
        CLS
        DISPLAY "The following questions are related to the shoulder
surface the vehicle slid over during the post-impact skid.
        (Press any key to continue ....)~"
        CLS
        FIND Tot_Drag2
        Post_Speed = (@SQRT(19.6*(Tot_Drag1*Skid_DistA +
        (Tot_Drag1 + Tot_Drag2)*Skid_DistB/2)*Brake_Eff));

RULE POSTSKID3E                                 ! Three surfaces
IF      Mult_Surface = Yes AND
        Num_Surfaces = Three AND
        Diff_Surfaces3 = Continuous AND         ! Four tires each surface
        Skid_Dist1 <> UNKNOWN AND
        Skid_Dist2 <> UNKNOWN AND
        Skid_Dist3 <> UNKNOWN
THEN
        CLS
        DISPLAY "The following questions are related to the FIRST
surface the vehicle slid over during the post-impact skid.
        (Press any key to continue ....)~"
        CLS
        FIND Tot_Drag1
        CLS
        DISPLAY "The following questions are related to the SECOND
surface the vehicle slid over during the post-impact skid.
        (Press any key to continue ....)~"
        CLS
        FIND Tot_Drag2
        CLS
        DISPLAY "The following questions are related to the THIRD
surface the vehicle slid over during the post-impact skid.
        (Press any key to continue ....)~"
        CLS
```

```
        FIND Tot_Drag3
        Post_Speed = (@SQRT(19.6*(Tot_Drag1*Skid_Dist1+Tot_Drag2*Skid_Dist2
        +Tot_Drag3*Skid_Dist3)*Brake_Eff));

RULE POSTSKID3F
IF      Mult_Surface = Yes AND              ! Combination of continuous surface
        Num_Surfaces = Three AND
        Diff_Surfaces3 = Combination AND    ! skid followed by a split surface
        Skid_DistA <> UNKNOWN AND           ! skid.
        Skid_DistB <> UNKNOWN
THEN
        CLS
        DISPLAY "The following questions are related to the homogenous
surface the vehicle slid over during the post-impact skid.
        (Press any key to continue ....)~"
        CLS
        FIND Tot_Drag1
        CLS
        DISPLAY "The following questions are related to the road surface
(for the pavement/shoulder portion of the skid) the vehicle slid over during
the post-impact skid.
        (Press any key to continue ....)~"
        CLS
        FIND Tot_Drag2
        CLS
        DISPLAY "The following questions are related to the shoulder
surface the vehicle slid over during the post-impact skid.
        (Press any key to continue ....)~"
        CLS
        FIND Tot_Drag3
        Post_Speed = (@SQRT(19.6*(Tot_Drag1*Skid_DistA +
        (Tot_Drag2 + Tot_Drag3)*Skid_DistB/2)*Brake_Eff));


!==============================================================
!               Rules to determine Post-Impact Speed
!               From Roll Distances
!==============================================================


RULE ROLL1
IF      Roll_Spd <> UNKNOWN
THEN
        Roll_Spdv1 = (Roll_Spd)
        RESET Roll_Spd;

RULE ROLL2
IF      Roll_Spd <> UNKNOWN
THEN
        Roll_Spdv2 = (Roll_Spd)
        RESET Roll_Spd;
```

```
RULE ROLL3
IF      Roll_Dist <> UNKNOWN
THEN
        Roll_Spd = (@SQRT(1.96*Roll_Dist));


!=============================================================
!               Rules to determine Post-Impact Speed
!               From Overturn Distances
!=============================================================


RULE VAULT1
IF      Vlt_Spd <> UNKNOWN
THEN
        Vlt_Spdv1 = (Vlt_Spd)
        RESET Vlt_Spd;

RULE VAULT2
IF      Vlt_Spd <> UNKNOWN
THEN
        Vlt_Spdv2 = (Vlt_Spd)
        RESET Vlt_Spd;

RULE VAULT3
IF      Horz_Dist <> UNKNOWN AND
        Vert_Fall <> UNKNOWN AND
        Take_Off = No
THEN
        Vlt_Spd = (2.21*Horz_Dist/(@SQRT(Vert_Fall)));

RULE VAULT4
IF      Horz_Dist <> UNKNOWN AND
        Vert_Fall <> UNKNOWN AND
        Take_Off = Yes AND
        Take_Elev <> UNKNOWN
THEN
        Vlt_Spd = (2.21*Horz_Dist/(@SQRT(Vert_Fall+Horz_Dist
        *Take_Elev/100)));


!=============================================================
!               Rules to determine Post-Impact Speed
!               From Overturn Distances
!=============================================================


RULE OVER1
IF      O_Spd <> UNKNOWN
THEN
        O_Spdv1 = (O_Spd)
        RESET O_Spd;
```

```
RULE OVER2
IF      O_Spd <> UNKNOWN
THEN
        O_Spdv2 = (O_Spd)
        RESET O_Spd;

RULE OVER3
IF      Over_Dist <> UNKNOWN AND
        Over_Drag <> UNKNOWN
THEN
        O_Spd = (@SQRT(19.6*Over_Dist*Over_Drag));


!=================================================================
!               Rules to determine Impact Speed
!               Using Conservation of Linear Momentum
!=================================================================


RULE MOM1                                       ! Must estimate a speed if
IF      Good_Angles = known AND                 ! vehs approaching head-on or
        Theta1 = (Phi1-180) OR                  ! rear-end
        Theta1 = (Phi1)
THEN
        DISPLAY "Since the approach paths of Vehicle 1 and
Vehicle 2 are along the same line it is necessary to
estimate a pre-impact speed for one of the vehicles.
"
        FIND Est_Speed
        AbsA1 = (@ABS(Theta2-Theta1))
        AbsA2 = (@ABS(Phi2-Phi1))
        FIND A1
        FIND A2
        MM1 = (Post_Speedv1*Post_Speedv1+Imp_Speedv1*Imp_Speedv1)
        MomD_v1 = (3.6*(@SQRT(MM1-2*Post_Speedv1*Imp_Speedv1
        *(@COS(A1)))))
        MM2 = (Post_Speedv2*Post_Speedv2+Imp_Speedv2*Imp_Speedv2)
        MomD_v2 = (3.6*(@SQRT(MM2-2*Post_Speedv2*Imp_Speedv2
        *(@COS(A2)))))
        Mom_Speed = found;

RULE MOM2                                       ! V2 approaching at any other
IF      Good_Angles = known AND                 ! angle prior to impact, use
        Theta1 = 0 AND                          ! these formula
        Phi1 <> 180
THEN
        Imp_Speedv2 = ((M1*Post_Speedv1*(@SIN(T2))
        +M2*Post_Speedv2*(@SIN(P2)))/(M2*(@SIN(P1))))
        Imp_Speedv1 = ((M1*Post_Speedv1*(@COS(T2))
        +M2*Post_Speedv2*(@COS(P2))-M2*Imp_Speedv2*(@COS(P1)))/M1)
        AbsA1 = (@ABS(Theta2-Theta1))
```

```
        AbsA2 = (@ABS(Phi2-Phi1))
        FIND A1
        FIND A2
        MM1 = (Post_Speedv1*Post_Speedv1+Imp_Speedv1*Imp_Speedv1)
        MomD_v1 = (3.6*(@SQRT(MM1-2*Post_Speedv1*Imp_Speedv1
        *(@COS(A1)))))
        MM2 = (Post_Speedv2*Post_Speedv2+Imp_Speedv2*Imp_Speedv2)
        MomD_v2 = (3.6*(@SQRT(MM2-2*Post_Speedv2*Imp_Speedv2
        *(@COS(A2)))))
        Mom_Speed = found;


RULE EST1
IF      Pre_Spv1 <> UNKNOWN                 ! Need an estimate of V1's speed
THEN                                        ! prior to impact
        Imp_Speedv1 = (Pre_Spv1/3.6)
        Imp_Speedv2 = ((M1*Post_Speedv1*(@COS(T2))
        +M2*Post_Speedv2*(@COS(P2))-M1*Imp_Speedv1*(@COS(T1)))/M2)
        Est_Speed = found;

RULE EST2
IF      Pre_Spv2 <> UNKNOWN                 !Need an estimate of V2's speed
THEN                                        ! prior to impact
        Imp_Speedv2 = (Pre_Spv2/3.6)
        Imp_Speedv1 = ((M1*Post_Speedv1*(@COS(T2))
        +M2*Post_Speedv2*(@COS(P2))-M2*Imp_Speedv2*(@COS(P1)))/M1)
        Est_Speed = found;


!===========================================================
!                       Momentum: Angle Check
!===========================================================


RULE TOPCHECK
IF      M1 <> UNKNOWN AND                   ! Query for vehicle masses
        M2 <> UNKNOWN                       ! Always passes if known.
THEN
        Theta1 = 0
        DISPLAY "KAR sets Vehicle 1's pre-impact direction (heading)
at 0 degrees. Remaining directions of travel (headings) will be measured
clockwise relative to Vehicle 1's heading of 0 degrees.
*Example* If Vehicle 1 is Northbound (0 degrees) and Vehicle 2 is westbound,
        Vehicle 2's heading prior to impact would be 270 degrees.
"

        X=0
        Var1 = Known
        WHILEKNOWN Var1
                X=(X+1)
                RESET Theta2
                RESET Phi1
                RESET Phi2
```

```
            RESET Angles
            RESET Rng
            RESET Angle_Range
            RESET Low_Range
            RESET High_Range
            RESET Check
            RESET Re_enter
            FIND Angles
            FIND Angle_Range
            FIND Check
            FIND Re_enter
      END
      RESET X
      T1 = (Theta1)
      P1 = (0.01745*Phi1)
      T2 = (0.01745*Theta2)
      P2 = (0.01745*Phi2)
      Good_Angles = Known;

RULE  ANGLE
IF    Theta2 <> UNKNOWN AND                ! Queries user for angles
      Phi1 <> UNKNOWN AND
      Phi2 <> UNKNOWN
THEN
      Rng = (Phi1-Theta1)                  ! Used in rule RANGE
      Angles = Known;

RULE  RANGE
IF    Rng >= 0 AND
      Rng <= 180
THEN
      Low_Range = (Theta1)
      High_Range = (Phi1)
      Angle_Range = Found
ELSE
      Low_Range = (Phi1)
      High_Range = (Theta1)
      Angle_Range = Found;

RULE  CHECK1
IF    Low_Range <= 180 AND
      Theta2 >= (Low_Range) AND
      Phi2 >= (Low_Range) AND
      Theta2 <= (High_Range) AND
      Phi2 <= (High_Range)
THEN
      Check = Good;
```

```
RULE CHECK2
IF      Low_Range > 180 AND
        High_Range > 180 AND
        High_Range < 360 AND
        Theta2 >= (Low_Range) AND
        Phi2 >= (Low_Range) AND
        Theta2 <= (High_Range) AND
        Phi2 <= (High_Range)
THEN
        Check = Good;

RULE CHECK3
IF      Low_Range > 180 AND
        High_Range < 180 AND
        Theta2 >= (Low_Range) OR
        Theta2 <= (High_Range) AND
        Phi2 >= (Low_Range) OR
        Phi2 <= (High_Range)
THEN
        Check = Good
ELSE
        Check = Bad;

RULE ENTER1
IF      Check = Good
THEN
        RESET Var1
        Re_enter = Known;

RULE ENTER2
IF      Check = Bad AND
        X > 1
THEN
        DISPLAY "I am still not satisfied with the angles you have entered,
but I will continue with the calculations."
        RESET Var1
        Warning = Momentum_Angles
        Re_enter = Known
ELSE
        DISPLAY "Your approach and departure angles are questionable.

        Theta1 = {Theta1}       Phi1 = {Phi1}
        Theta2 = {Theta2}       Phi2 = {Phi2}

Please check them over and re-enter the values."
        Re_enter = Known;
```

```
!=============================================================
!                    Queries To User
!=============================================================

!-------------------- Witness Queries ------------------------------

ASK Post_Spv1: "Please enter post-impact speed estimate of Vehicle 1 (km/h).";

ASK Post_Spv2: "Please enter post-impact speed estimate of Vehicle 2 (km/h).";

ASK Pre_Spv1: "Please enter pre-impact speed estimate of Vehicle 1 (km/h).";

ASK Pre_Spv2: "Please enter pre-impact speed estimate of Vehicle 2 (km/h).";

!-------------------- Skid Queries ---------------------------------

ASK Brake_Eff: "What is the braking efficiency
(the number of functional brakes)/(the number of tires)?";
CHOICES Brake_Eff: 1.0,0.75,0.5,0.25;

ASK Mult_Surface: "Is there more than one road surface (drag factor)
involved in this skid?";
CHOICES Mult_Surface: Yes,No;

ASK Num_Surfaces: "How many surfaces did the vehicle slid over during this
phase of the accident?";
CHOICES Num_Surfaces: Two,Three;

ASK Diff_Surfaces2: "Two Surfaces -
(i) Was the skid CONTINUOUS from one surface to another, or
(ii) Was one side of the vehicle skidding on a different surface than
the other side (SPLIT_SIDES), or
(iii) Did it involve a COMBINATION of the above?";
CHOICES Diff_Surfaces2: Continuous,Split_Sides,Combination;

ASK Diff_Surfaces3: "Three Surfaces -
(i) Was the skid CONTINUOUS from one surface to the next, or
(ii) Did it involve one road surface followed/preceded by shoulder-pavement
skid where the two road surfaces were different?";
CHOICES Diff_Surfaces3: Continuous,Combination;

ASK Skid_Dist: "What is the skid distance (in meters)?";

ASK Skid_Dist1: "What is the skid distance over the first surface
(in meters)?";
```

ASK Skid_Dist2: "What is the skid distance over the second surface (in meters)?";

ASK Skid_Dist3: "What is the skid distance over the third surface (in meters)?";

ASK Skid_DistA: "What is the skid distance over the homogeneous surface (in meters)?";

ASK Skid_DistB: "What is the distance of the shoulder/pavement skid (in meters)?";

ASK Drag_Value: "If tests were conducted to determine the drag factor please enter this value, otherwise, enter a '?' and a drag factor will be selected based on your response to a series of questions.";

ASK Incline: "Is this section of roadway/terrain sloped?";
CHOICES Incline: Yes,No;

ASK Deg_Incline: "What is the percent slope (+/-)?";

ASK Road_Mix: "What is the road surface type?";
CHOICES Road_Mix: Portland_Cement,Asphalt_or_Tar,Gravel,Ice;

ASK Road_ConditionPC: "What is the condition of the road?";
CHOICES Road_ConditionPC: New,Travelled,Polished;

ASK Road_ConditionAT: "What is the condition of the road?";
CHOICES Road_ConditionAT: New,Travelled,Polished,Excess_Tar;

ASK Road_ConditionGR: "What is the condition of the road?";
CHOICES Road_ConditionGR: Packed_Oiled,Loose;

ASK Weather: "Was the road surface wet or dry?";
CHOICES Weather: Wet,Dry;

ASK Speed_Range: "What was the approximate speed of the vehicle?";
CHOICES Speed_Range: Less_50kmh,Greater_50kmh;

!---------------- Momentum and Energy Queries -----------------------

ASK Choose_ME: "Which speeds do you want to use for the pre-impact calculations?";
CHOICES Choose_ME: Momentum,Energy;

ASK Theta2: "What is Vehicle 1's heading after impact (in degrees)?";

ASK Phi1: "What is Vehicle 2's heading prior to impact (in degrees)?";

ASK Phi2: "What is Vehicle 2's heading after impact (in degrees)?";

ASK Energy_Meth: "(i) Do you have the velocity changes for both vehicles from a CRASH analysis, or
(ii) Would you like KAR to estimate the velocity changes using energy calculations?";
CHOICES Energy_Meth: CRASH_Values,KAR_Estimate;

ASK Veh_Headings: "Are both vehicles approach and departure headings well defined?";
CHOICES Veh_Headings: Yes,No;

ASK Num_measv1: "How many crush dimensions were taken for Vehicle 1?";
CHOICES Num_Measv1: Two,Four,Six;

ASK Num_measv2: "How many crush dimensions were taken for Vehicle 2?";
CHOICES Num_Measv2: Two,Four,Six;

ASK Len_v1: "What is the width of Vehicle 1's crush profile (cms)?";

ASK Len_v2: "What is the width of Vehicle 2's crush profile (cms)?";

ASK Delta_v1: "Please enter the change in velocity for Vehicle 1 (km/h).";

ASK Delta_v2: "Please enter the change in velocity for Vehicle 2 (km/h).";

ASK Gamma_v1: "Please enter the principle direction of the impact force acting on Vehicle 1 (degrees).";

ASK Gamma_v2: "Please enter the principle direction of the impact force acting on Vehicle 2 (degrees).";

ASK AGamma_v1: "Please enter an estimate of the principle direction of the impact force acting on Vehicle 1 (degrees)
** The exact PDOF will be calculated based on the vehicle headings and*
** will appear in the Parameter Table.";*

ASK AGamma_v2: "Please enter an estimate of the principle direction of the impact force acting on Vehicle 2 (degrees)
** The exact PDOF will be calculated based on the vehicle headings and*
** will appear in the Parameter Table.";*

ASK M1: "What is the weight of Vehicle 1 (in kilograms)?";

ASK M2: "What is the weight of Vehicle 2 (in kilograms)?";

ASK Wbase_v1: "What is Vehicle 1's wheelbase (cms)?";

ASK Wbase_v2: "What is Vehicle 2's wheelbase (cms)?";

ASK Coll_Surfv1: "Where is the collision surface Vehicle 1?";
CHOICES Coll_Surfv1: Front,Rear,Side;

ASK Coll_Surfv2: "Where is the collision surface on Vehicle 2?";
CHOICES Coll_Surfv2: Front,Rear,Side;

ASK C1V1: "Please enter the Crush Measurements.
(For Front and Rear damage C1 is the leftmost measurement,
for Side damage C1 is the rearmost measurement).

Vehicle 1 - C1?";

ASK C2V1: "Vehicle 1 - C2?";

ASK C3V1: "Vehicle 1 - C3?";

ASK C4V1: "Vehicle 1 - C4?";

ASK C5V1: "Vehicle 1 - C5?";

ASK C6V1: "Vehicle 1 - C6?";

ASK C1V2: "Please enter the Crush Measurements.
(For Front and Rear damage C1 is the leftmost measurement,
for Side damage C1 is the rearmost measurement).

Vehicle 2 - C1?";

ASK C2V2: "Vehicle 2 - C2?";

ASK C3V2: "Vehicle 2 - C3?";

ASK C4V2: "Vehicle 2 - C4?";

ASK C5V2: "Vehicle 2 - C5?";

ASK C6V2: "Vehicle 2 - C6?";

!--------------------- Yaw Queries ----------------------------------

ASK Radius: "Do you have a VALUE for the curve radius of the yaw mark, or would you like to CALCULATE it based on a chord length and middle ordinate?";
CHOICES Radius: Value,Calculate;

ASK Chord: "What is the chord length (meters)?";

ASK Middle_Ord: "What is the length of the middle ordinate (meters)?";

ASK Rad: "What is the radius of curvature for the first one_third of the yaw mark (in meters)?";

ASK Super: "Is the roadway superelevated?";
CHOICES Super: Yes,No;

ASK Sup_Elev: "What is the percent superelevation (+/-)?";

!--------------------- Roll Queries --------------------------------

ASK Roll_Dist: "What distance did the vehicle roll (in meters)?";

!--------------------- Vault Queries --------------------------------

ASK Horz_Dist: "What is the HORIZONTAL distance the vehicle travelled from the point of takeoff to the point where it first hit the ground (in meters)? Measurements should be taken from the vehicles center of mass.";

ASK Vert_Fall: "What is the VERTICAL distance the vehicle fell (in meters)? Measure from center of mass to center of mass.";

ASK Take_Off: "Was the vehicle path leading to the point of takeoff sloped?";
CHOICES Take_Off: Yes,No;

ASK Take_Elev: "What is the percent grade of this slope (+/-)?";

!--------------------- Overturn Queries ----------------------------

ASK Over_Dist: "What distance did the vehicle slid while overturned (in meters)?";

ASK Over_Drag: "What is the drag factor between the road surface and the overturned surface?";

!---------------------- Plural Variables ----------------------------

PLURAL:
Method_Postv1,Method_Postv2,Method_Impact,Method_Prev1,Method_Prev2, Summary1,Summary2,Warnings;

PLURAL: The_Post_Spds1,The_Post_Spds2,The_Pre_Spds1,The_Pre_Spds2, Post_Actv1,Post_Actv2;

PLURAL: Select_Postv1,Select_Postv2,Select_Impact,Select_Prev1,Select_Prev2;

PLURAL:
Post_Analysis1,Post_Analysis2,Imp_Analysis,Pre_Analysis1,Pre_Analysis2,
Warning;

APPENDIX D. Police Measurement Sheet

#	X	X	Y	REMARKS
				FIRE HYDRANT N-E CORNER OF REFERENCE POINT
			2.33	NORTH CURB ADJACENT TO R11
	-0.75		5.0	START OF RIGHT SKID MARK
	-5.10		6.25 ~~5.20~~	START OF LEFT SKID MARK
		10.0	5.10	RIGHT SKID 10 METERS WEST OF START
		10.0	6.48	LEFT SKID "
		15.72	5.20	RIGHT SKID AT DEVIATION
		16.58	6.65	LEFT SKID AT DEVIATION
		16.10	5.0	LEFT REAR SCUFF (BEGINNING)
		23.2	8.60	LEFT REAR DATSUN
		23.85	7.25	RIGHT REAR DATSUN
		27.05	8.51	RIGHT FRONT DATSUN
		30.0	8.53	LEFT FRONT VOLVO
		34.12	6.70	LEFT REAR VOLVO
		34.86	5.03	RIGHT REAR VOLVO
			6.0	WIDTH W/B - 2 LANE
			9.76	WIDTH W/B - 1 LANE
			17.70	SOUTH MEDIAN CURB
		8.20		EAST CURB
		11.25		WEST CURB
			20.88	BROKEN LINE E/B
			27.42	SOUTH CURB
		6.10		WEST POINT OF MEDIAN EAST OF
		19.6		
				DRAG SLED = 17.0K
				11, 11, 12, 13, 12
				12, 13, 13, 13,