Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Automotive Engineering Department
Masters in Automotive Engineering

DEVELOPMENT OF NEW METHODOLOGIES FOR ROAD ACCIDENT RECONSTRUCTION WITH CDR TOOL

ANDRÉ GASPAR FRANCISCO

Leiria, Setembro 2022

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Automotive Engineering Department
Masters in Automotive Engineering

DEVELOPMENT OF NEW METHODOLOGIES FOR ROAD ACCIDENT RECONSTRUCTION WITH CDR TOOL

ANDRÉ GASPAR FRANCISCO
Number: 2202281

Projeto Automóvel realizado sob orientação do Professor Sérgio Pereira dos Santos (ssantos@ipleiria.pt), e coorientação do Professor Carlos Daniel Henriques Ferreira (ferreira@ipleiria.pt).

Leiria, Setembro 2022

ACKNOWLEDGMENTS

I would like to thank all my teachers involved in this work, most notably, my supervisor professor Sérgio Pereira dos Santos for introducing me to this topic and co-supervisor Carlos Daniel Henriques Ferreira for helping me write this thesis, and their time and guidance throughout this learning experience.

Also would like to thank my family for their continuous support, I'm very thankful for their hard work and guiding me thorough this work. I would also like to thank my dearest Carolina, for all the love and comfort provided during my hardest moments.

Finally, I would also like to thank all my dearest friends for the good memories that will be remembered for years to come. I would like to make a special thanks to Telmo Lopes and José Miguel for all the help peer reviewing this thesis and for all the entertaining moments.

RESUMO

Os acidentes rodoviários são uma realidade nos dias de hoje. Com o intuito de reduzir e entender os mesmos e, consequentemente melhorar a segurança rodoviária, será feita uma análise não só aos métodos antigos e modernos de reconstrução de acidentes mas também à forma como estes podem ser melhorados.

Com a nova regulação implementada pela União Europeia a entrar em vigor no ano 2022, que obriga o uso de *Event Data Recorders* (EDR) em todos os veículos produzidos, surge um sistema que grava os dados dinâmicos do veículo e as ações do condutor, em caso de acidente. O objetivo deste projeto é analisar esses dados no contexto de um acidente e simplificar o seu processo de reconstrução.

Este Projeto visa a interpretação da cena do acidente e dos dados dinâmicos do veículo, de maneira a ser obtida uma melhor perceção do seu contexto tendo em conta os dados gravados no EDR.

Para além disto, serão estudadas as metodologias de interpretação da cena do acidente e as diversas ferramentas que podem ser usadas na sua reconstrução, culminando num sistema que digitaliza a cena do acidente para ser usado num *software* de computador.

Os dados dinâmicos relativos ao acidente podem ser divididos em três fases: pré-acidente, impacto e pós-acidente. Cada fase necessita de uma abordagem diferente dependendo dos tipos de dados, de modo a obter uma trajetória do veículo em todas as fases.

Keywords: Event Data Recorder, CDR, Collision, Reconstruction, Dynamics, Accident

ABSTRACT

Road accidents are a constant reality these days, therefore to reduce and understand these accidents and, as a result, improve road safety, an analysis will be carried out on modern and old accident reconstruction methods and how they can be improved.

With the new 2022 European Union regulation coming in, which enforces the use of *Event Data Recorders* (EDR) on newly made vehicles, suddenly there's a system that can record vehicle dynamic data and driver inputs, in case of an accident. This project's objective is to analyze this data in an accident context and simplify the reconstruction process.

This project aims for a seamless interpretation of the accident scene and vehicle dynamics to better understand the data in the scene's context. Some methodologies and tools used for the reconstruction of the accident scene will be studied, culminating in a system that digitizes the scene for later use in computer software.

The dynamic data from the accident can be divided into three phases: pre-crash, impact, and post-crash. Each phase requires a different approach which depends on the type of data to obtain the trajectory of the vehicle throughout the phases.

Keywords: Event Data Recorder, CDR, Collision, Reconstruction, Dynamics, Accident

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

2D	Two-Dimensional
3D	Tri-Dimensional
ABS	Anti-Lock Braking System
ACM	Airbag Control Module
ADAS	Advanced Driver Assistance Systems
BUS	Data-bus
CAD	Computer Aided Design
CAN	Controller Area Network
CDR	Crash Data Retriever
CRC	Cyclic Redundancy Check
D2M	Direct To Module
DLC	Data Link Connector
DOF	Degrees Of Freedom
DSLR	Digital Single-Lens Reflex
DTC	Diagnostic Trouble Code
ECU	Electronic Control Unit
EDM	Electronic Distance Measurement Instrument
EDR	Event Data Recorder
EEPROM	electrically erasable programmable read-only memory
EES	Equivalent Energy Speed
EU	European Union
GPS	Global Positioning System

GUI	Graphics User Interface
kbps	kilobits per second
LiDAR	Light Detection and Ranging
LIN	Local Interconnected Network
mbps	megabits per second
OBD	On-Board Diagnostics
OEM	Original Equipment Manufacturer
PCM	Powertrain Control Module
PDOF	Principle Direction Of Force
PID	Proportional–Integral–Derivative Controller
PPM	Pedestrian Protection Module
RAM	Random Access Memory
ROM	Read Only Memory
ROS	Roll Over Sensor
RPM	Rotation Per Minute
RTK GPS	Real Time Kinematic GPS
TOF	Time Of Flight
TPS	Throttle Position Sensor
UAV	Unmanned Aerial Vehicle
UI	User Interface
VCI	Vehicle Computer Interface
VIN	Vehicle Identification Number

1

INTRODUCTION

1.1 INTRODUCTION

Vehicle accidents are something that will always be present on the roads. In order to understand the events and improve the safety of pedestrians and passengers, this master thesis in Automotive Engineering will provide new methods for accident reconstruction regarding the newly added, as per 2022, European Union's regulations to include *Event Data Recorders* (EDR) in every vehicle, providing new datapoints for study. The report will focus on improving the workflow of accident reconstruction and introduce new procedures that takes advantage of this new tool. This new methodology aims to create a crash scenario that mimics the real life events as much as possible by presenting new ways of capturing data from the crash location and merge with the dynamic data from the EDR through a vehicle dynamic model.

Starting with an overview of the diferent methods and tools that authorities use when gathering data from the crash site, following with diferent parameters inherent to vehicle accidents such as Δv and *Equivalent Energy Speed.* Also showcasing the accident reconstruction computer program PC-CRASH™ which is an essential tool to accelerate crash reconstruction and evaluate them. Methods for assessing the damage done to the vehicles as well as evaluating the pedestrian data will also be viewed, culminating on an analysis of the process most crash reconstruction professionals use. Then for the overview of the vehicle's electronics, an analysis of the Intra-vehicle network communication lines and protocols and the creation of the automotive *Black Box*, predecessor to the EDR. This information will be followed with an extensive explanation on what consist an EDR and the Tools to extract data from it, as well as what consist the new European regulations. Finally, the overview of previous work ends by explaining a series of vehicle models and their use cases in this study.

As for the development phase of this report, a new workflow will be proposed by this study and it can be summarized in the following sequence of events:

1. Collect data from an accident scene.
2. Read data from EDR.
3. Apply data to a dynamic model.
4. Analyse results.

New ways to use the techniques for gathering information from the accident location as well as recreating the scenery and road geometry as a 3D set will be studied and what kind of advantages brings to this type of application.

The EDR data will be divided into 3 sequential phases. First phase is the pre-crash data, where the data recorded by the EDR will be used to recreate its movement and evaluate the driver's inputs in the moments before the impact. Second phase is the crash pulse, its the total velocity, and it direction, suffered by the vehicle upon impact, it indicates the severity of the accident. The third and final phase its the post-crash data, it the moments from the impact until the vehicles final resting position.

Using the dynamic models for each crash phase the movement of the vehicle is recreated and then matched with the accident scene data, giving context to the EDR data and providing the user with a clear recreation of the accident, the resulting simulation can be viewed as a computer generated movie.

1.2 THESIS OVERVIEW

What follows is a brief overview of each chapter's content.

- Chapter 1: Introduction - A overview of the whole report is made, introducing our primary objectives and methodology to achieve them, as well as explaining the context of the report.
- Chapter 2: State of Art - Past work as well as present methodologies for crash reconstruction are explained, in order to better understand what needs to be done to accommodate what is coming next.
- Chapter 3: Development - The required models to interpret the dynamic data are developed here as well as methodologies to better reconstruct the crash scene and improve simulations quality. An analysis of the subsequent result is also made.
- Chapter 4: Conclusions as Future Work - The ending chapter for this report includes what has been accomplished throughout this study and what can or needs to be done to improve it.

2

STATE OF ART

2.1 ACCIDENT SCENE DATA COLLECTING TOOLS

The ability to acquire data relative to the crash site is necessary for its reconstruction in a way that the deductions made about the accident reflect the real-life events as much as possible to determine the cause of an accident. This data is comprised of:

- Tire marks
- Vehicle position
- Body positioning
- Pavement marks
- Debris
- Others...

The amount of variables in an accident scene means that there will not be a clear solution for every case, therefore some tools will be introduced to gather the necessary information, but they all follow the same principle of measuring several distances to a reference point. Some of these methods are already in use in the United States of America, in a guide book published by the U.S. Department of Transportation Federal Highway Administration [1], which gives an excellent example for each tool.

2.1.1 *Classic Measurement Tools*

Using a measuring tape or a measuring wheel are crude ways of measuring distance and only the skill of the operator will make sure the measurements are accurate or not, also the usage of compasses or protractors to obtain angles.

This measurements tools are primarily used to build sketches, they can give good results in straight and plane sections of the road, decreasing quality as the road section complexity increases such as turns and bumps.

It's the easier way to acquire information and it's used by the majority of authorities, not requiring training or expensive devices, although in more complex circumstances the information might be of lower quality and can have human error. The process of measuring may take a while since the operator has to do it by hand which makes it vulnerable to traffic, unless the traffic flow can be interrupted.

2.1.2 *Photogrammetry*

With a basic-level DSLR (Digital Single Lens Reflex) camera, several pictures are taken from various points of view and with different markers in sight which are then processed by Photogrammetry software to build 2D and 3D Sketches.

To ensure good practice the operator must take at least 3 photos of the same object at different positions, ideally, some parts of the photo need to overlap each other. While the act of taking photos might not require skill, using the software might, in order to ensure the best possible result the operator needs to reproduce the crash scene as close to reality as possible. This method can reduce the staff involved and aid in the decongestion of the traffic flow and, as a camera can be quite common, there is no need to wait for special equipment. However in certain atmospheric conditions the quality of the photos might not be sufficient to go with this method.

2.1.3 *Light Detection and Ranging*

LiDAR is a tool that projects a laser onto a surface and by analyzing the time between emission and detection of the light reflected by the surface its possible to calculate the distance between the transmitter and surface. With the use of an angle encoder, it is possible to calculate polar coordinates and build a *pointcloud*. This method is more complex and requires knowledgeable operators to interpret the output of the sensor to verify if there was not any human error during assembly.

The LiDAR can be portable or mounted on a tripod, the latter being more precise in measurements. The operator can choose wherever to place the sensor which can be off the street, clearing it for traffic flow. Atmospheric conditions can reduce the effective distance of the *pointcloud*.

2.1.4 *Total Stations*

This type of station consists of a Theodolite, a precision instrument that measures both vertical and horizontal angles, an EDM (Electronic Distance Measurement) Instrument, that measures distances in real-time, an optical prism and a sensor. The prism is the target used to point the Theodolite to, in which the EDM will get the distance for each point. The gathered data is analyzed and a precise map of the scene is built. This equipment is more complex than other options requiring a trained professional to handle it. Similarly to some sensor based systems, it may have different readings depending on atmospheric conditions. Despise the concept being relatively simple, there are other variants of the same principle, among which:

- Reflectorless Total Stations

  Removes the need to have a prism as a target to a certain range, getting measurements by simply pointing the device.

- Semi-Robotic Total Stations

  By motorizing every axis we can use the laser to track and point the target automatically, removing the need for an operator in both ends.

- Robotic Total Stations

  Similarly to the Semi-Robotic but there's an added remote control, allowing for controlling the tracking of the station remotely

- Hybrid Total Stations

  By adding GPS and RTK GPS communication to a Robotic station it can calculate more precise polar coordinates from the site assisted by the RTK GPS.

- Imaging Stations

  A camera sensor is mounted to a Robotic Station, which then sweeps the area of interest and makes measurements.

2.1.5 *Global Positioning System*

The GPS (Global Positioning System) consist of two devices, a base and a rover. Both require a mobile data connection to communicate with a GPS constellation. If there's no connection, a second base can be used. Both devices communicate between them using *Bluetooth*. It works the same way as a measuring tape, but it uses the GPS triangulation to calculate the distance from the rover to the base.

2.1.6 *3D Scanning*

This technique uses a *Phase Shift* scanner, which measures the difference between the reflection phases of an emitted wave, and a TOF (Time of Flight) sensor, which measures the distance by projecting a certain pattern with an infrared laser and analyzing the reflection, to acquire points to build a *pointcloud*. A camera can also be attached to give each point a texture to create a more realistic *pointcloud*.

This scanner can be handheld and used by a single person, while the acquisition time depends on how dense the *pointcloud* needs to be. Some surfaces can be more difficult to read due to reflectivity.

2.1.7 *Unmanned Aerial Vehicle*

A UAV (Unmanned Aerial Vehicle), or most commonly Drones, are any aerial vehicle piloted remotely and usually fitted with a camera that nowadays is easy to obtain. In addition to piloting the drone remotely, since the device already has a GPS link, the operator can choose to make a flight plan to which the drone will fulfill, scanning the selected area and taking the necessary photograph. Once that's completed the same procedure, as discussed in Chapter 2.1.2, can be applied and create a topographic map or a *pointcloud* with GPS data embedded.

Atmospheric conditions can limit the implementation of this method since clear skies are needed for flight and good quality photos but it's the safest for the operator since he doesn't need to be inside the roadway.

2.2 ANALYTICAL CINEMATIC METHODS AND COMPUTER SIMULATIONS FOR CRASH RECONSTRUCTION

To help with accident reconstructions, some tools, both analytical and computational were created to better understand the dynamics inherent to vehicle collisions, primarily the use of the Law of conservation of Momentum.

2.2.1 *Delta v*

Δv is a utility that emerged in the 70's to improve crash reconstruction analysis and exists to measure a traffic collision's severity, defined as a change between the pre-impact velocity and post-impact velocity of a vehicle [2] in a specified time frame. The same change in velocity requires more stopping power if it happens during a shorter time frame than a longer one.

In its simpler form one can think of a planar crash consisting of a vehicle 1 with a mass of m_1 traveling at a certain velocity v_1 encounters vehicle 2 moving slower, v_2, with a mass of m_2 and at a certain T_0 the vehicles collide and exit the impact with matching velocities, $\tilde{v}$ and change their trajectories accordingly.

Assuming the collision can be considered inelastic and applying the law of conservation of momentum, since it is a closed system between the two vehicles, the momentum of both vehicles is given by mv and the momentum of both vehicles post-impact is $(m_1 + m_2)\tilde{v}$ and thus the post-impact velocity can be discovered using the equation (1) and rewritten to find Δv for each vehicle as in equation (2).

$$\tilde{v} = \frac{v_1 m_1 + v_2 m_2}{m_1 + m_2} \tag{1}$$

$$\begin{cases} \Delta v_1 = \frac{m_2}{m_1+m_2}\left(v_2 - v_1\right) \\ \Delta v_2 = \frac{m_1}{m_1+m_2}\left(v_1 - v_2\right) \end{cases} \tag{2}$$

Although this process is quite simple, it has 10% to 30% error factor, according to [2], since it treats the collision as a one-dimensional scenario. Usually two trajectories are involved and treating the collision as inelastic, where in reality collisions are elastic and presenting some sort of rebound, and some of the energy is dissipated in deforming the vehicle's chassis.

However there's another type of approach [3], since a vehicle in a crash can be described using vectors such as: an approach velocity vector, which indicates the speed and direction of the crashed vehicle, the change in velocity vector, which is the force impaired on the primary vehicle from the second vehicle in the moment of impact, and the departure velocity vector which depicts the speed and direction of the primary vehicle right after the impact. As demonstrated in figure 1, on the left is shown what happens during an impact, vehicle 1, in orange, is hit by vehicle 2, in green, at a certain Principle Direction of Force angle, PDOF, changing the trajectory of the vehicle 1. This can be modified into a vector triangle, facilitating the study of impact angles and angle of collision.

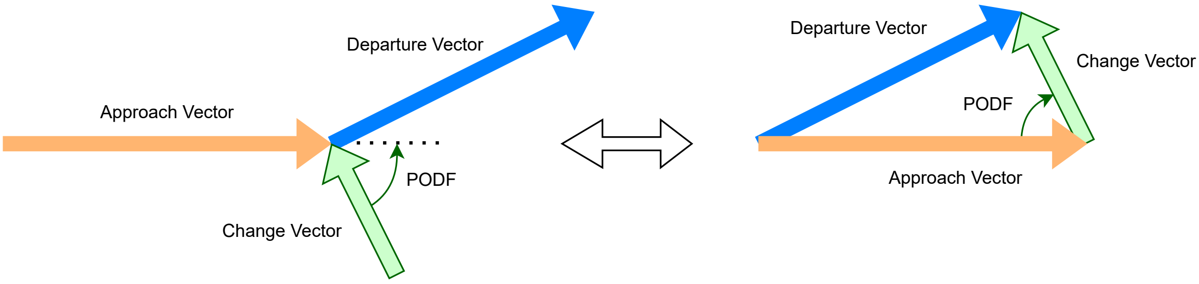


Figure 1: Triangle of Vectors

With this approach the pre and post-impact speed and direction of both intervening vehicles can be known provided that some of the information can be retrieved from the accident.

2.2.2 *Equivalent Energy Speed*

One of the most important piece of data required to make a comprehensive crash analysis is the determination of the impact velocity of the vehicle. In the most extreme cases of limited information, the EES (Energy Equivalent Speed) [4][5].

This parameter is defined by the theoretical speed, and therefore kinetic energy, that will be used as deformation energy during a collision with a rigid wall, however this only works if the deformation energy is known previously or achieved through different methods, either by calculating the damages or using speed-deformation curves.

Since the EES is defined as a form of kinetic energy, the equation for said energy can be rearranged to give EES as a function of E (deformations energy). As shown in equation (3).

$$E = \frac{1}{2}mv^2 \iff \frac{2E}{m} = v^2 \iff ESS = \sqrt{\frac{2E}{m}} \tag{3}$$

2.2.3 *Impulse Momentum Methods*

Impulse-Momentum Theory [6, Chapter 6] is a commonly used method in vehicle crash reconstruction that uses the conservation of linear momentum to analyze the pre and post-collision trajectories and achieve the impact velocities for each vehicle. Assuming that when two moving bodies make contact for a relatively short amount of time, their contact forces greatly exceed the other forces acting on the body, therefore this force is the dominant impulse and Newton's law of impulse and momentum can be applied.

Planar Impact method does not make assumptions and therefore it treats the rotational momentum applied to the vehicle as well as the vehicle's coefficient of restitution, the rebound of the vehicle upon collision, the impulse ratio coefficient and the ratio between the tangential and normal components of the impulse. In figure 2 it's shown the common free body diagram of the two vehicles upon collision. Vehicle 1 and 2 with mass m_1 and I_1 and inertia m_2 and I_2 respectively collide at point C forming the impact plane tn which offsets from the xy plane by an angle of Γ. Point C is at a distance of d_1 and d_2 and angle of ϕ_1 and ϕ_2 for both vehicles.

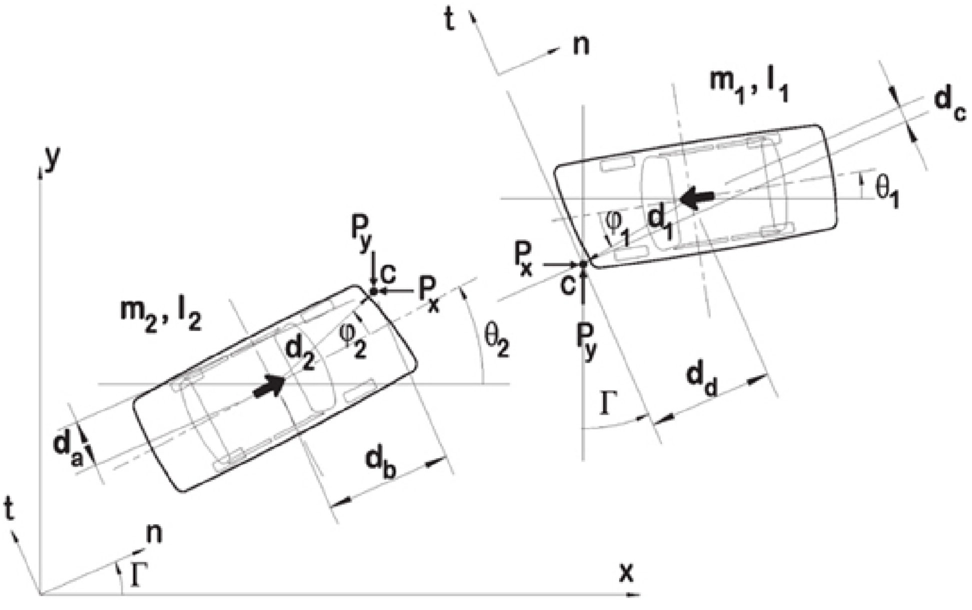


Figure 2: Free Body Diagram of Planar Impact Between Two Vehicles, adapted from [6, figure 6.5]

Following the set of equations, its possible to find the final velocity, but this can be quite complex to do by hand. Since sometimes the vehicle's rotation can be negligible, its possible to reduce the rigid body to a single point allowing the use of the Point-Mass method [7] [6, Chapter 7]. Using the conservation of momentum it's assumed that the sum of kinetic energies before the crash is the same as the sum of kinetic energies after the crash plus the deformation energy dissipated in the crash. Using this approach, some considerations must be made, particularly, exterior forces or tire forces are not considered on impact and therefore not used in the conservation of momentum, vehicle's mass and its inertia are constant during impact, there's no restitution and the impact force is considered as an instant pulse of maximum force.

Using the next sketch, figure 3, of a crash example where a vehicle B crashes into vehicle A on its left side at an angle. This impact deforms the vehicle along the plane of deformation with angle α, which is determined by analyzing the damage. The PDOF (Principle Direction of Force) is applied perpendicularly to this plane.

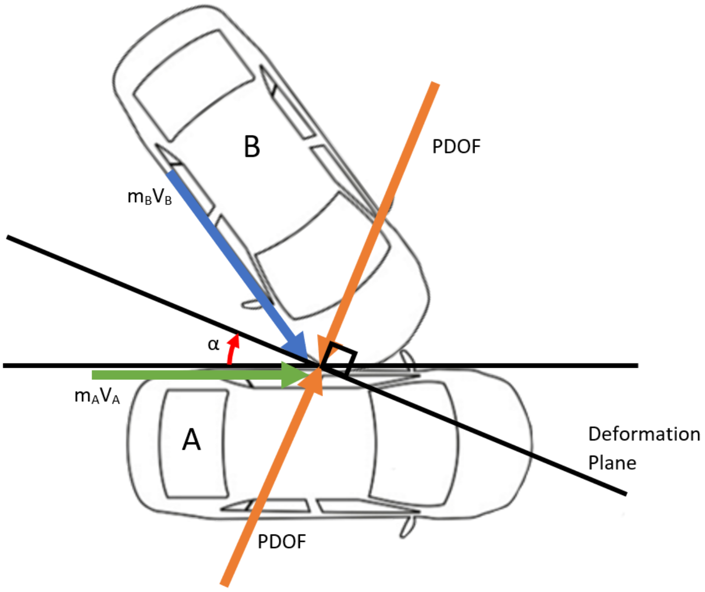


Figure 3: Example Sketch from a Collision

Assuming that the following variables are known, or can be easily deducted, the model can be applied.

1. θ_A and θ_B corresponding to the initial direction of motion for vehicles A and B respectively.
2. φ_A and φ_B correspond to the final direction of motion for vehicles A and B respectively which is known by analyzing the final position of the vehicle.
3. d_A and d_B corresponding to the distance traveled, in a straight line, from the impact point to the resting position for vehicles A and B respectively.
4. f_A and f_B correspond to the drag coefficients between tires and road, for vehicles A and B respectively.
5. m_A and m_B correspond to the weight for vehicles A and B respectively.
6. V_A' and V_B' corresponding to the post-impact velocities for vehicles A and B respectively.

As previously stated, this solves for pre impact velocities of V_A and V_B. V_A' and V_B' can be easily obtained by treating the post crash movement as a uniform decelerated movement and therefore using equation (4), the square root of 2 times gravity times coefficient of friction times the distance in straight line.

$$V = \sqrt{2gfd} \tag{4}$$

Therefore, only applying the equation (5) one can have a gross approximation for the vehicle's velocities just before the impact.

$$\begin{cases} m_A V_A \cos(\theta_A) + m_B V_B \cos(\theta_B) = m_A V'_A \cos(\varphi_A) + m_B V'_B \cos(\varphi_B) \\ m_A V_A \sin(\theta_A) + m_B V_B \sin(\theta_B) = m_A V'_A \sin(\varphi_A) + m_B V'_B \sin(\varphi_B) \end{cases} \tag{5}$$

However this assumes that no energy is lost in the deformation of material and that the collisions are completely elastic. Therefore improving on the Point-Mass Method, there's the Monte Carlo Method [8]. By adding a T vector to equation (5), which creates equation (6). This vector indicates the total energy used in deforming the material applied in the direction of the impact plane, taking into account the total Δv. The next figure 4, shows how a referential split into two sectors can be used to visualize the crash, with the left sector showing the pre-crash directions and velocities and, to the right, the post-crash vectors including the vector of maximum force.

$$\begin{cases} m_A V_A \cos(\theta_A) + m_B V_B \cos(\theta_B) = m_A V'_A \cos(\varphi_A) + m_B V'_B \cos(\varphi_B) + T\cos(\alpha) \\ m_A V_A \sin(\theta_A) + m_B V_B \sin(\theta_B) = m_A V'_A \sin(\varphi_A) + m_B V'_B \sin(\varphi_B) + T\sin(\alpha) \end{cases} \tag{6}$$

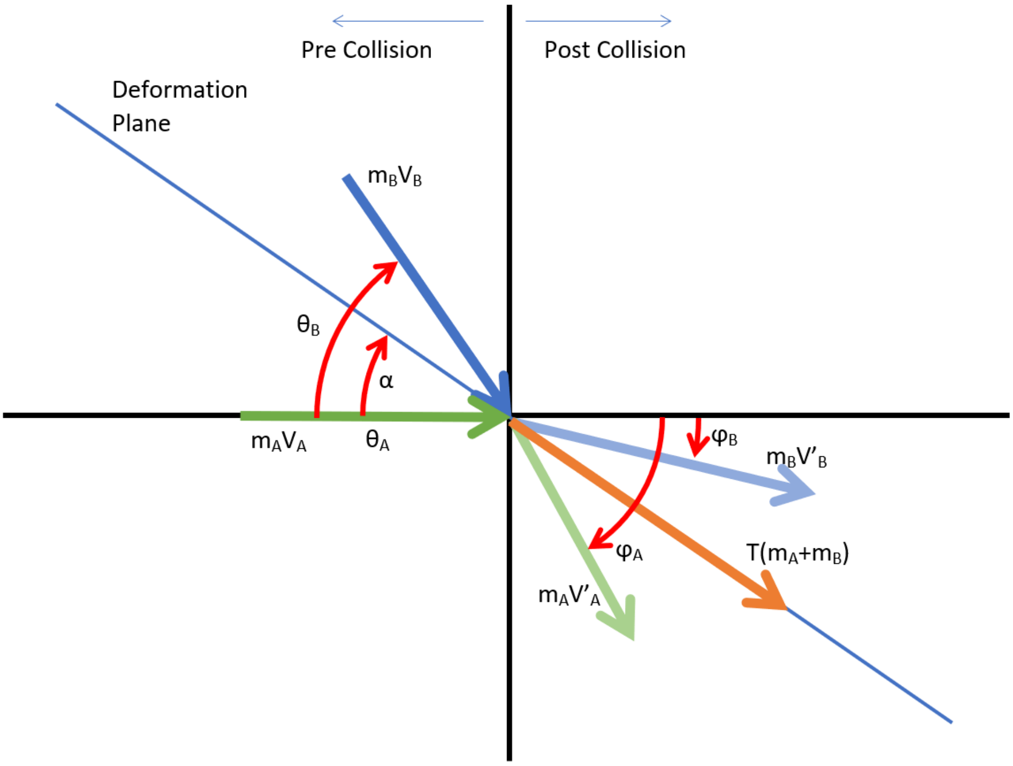


Figure 4: Point-Mass Impact Diagram

2.2.4 *The PC-CRASH™ Simulation Software*

The PC-CRASH™ [9] is the leading computer software for road accident reconstructions and study. Able to run kinematic and vehicle dynamics simulations combined with momentum-based models, this way accidents are simulated until all vehicles are stopped. The use of a GUI (Graphics User Interface) makes it easier to visualize the results and with the possibility to visualize the results in 3D and import them as a video.

A wide variety of tools are provided to make sure the simulation can both be realistic and visually pleasing, including drawing tools to produce diagrams and sketches from the accident. Its also able to load 2D and 3D predefined assets from a library which includes various choices of vehicles, houses, walls, fences, rails, poles and traffic signals, and other objects that might appear on the road.

It also allows importing a 3D scenario from a *pointcloud*, modulating the terrain's inclination and geometry, providing better realism to the simulations.

The catalog of vehicles included in the program is also extensive, with more than 50000 available for simulation, and some might include a 3D model to go with, which can improve simulation results while using the ”impact mesh” option.

The program mainly uses a rigid body model, however, multibody models are also available, that move under the influence of external forces. The system has a fixed inertia axis and coordinate direction. The following forces influence the body's movement, in which the program presents some models for:

- Tire Forces.
- Air Resistance.
- Gravity.
- Trailer Coupling.

The suspension models assumes that both the components and tires weight is zero, so the normal tire forces can be calculated directly through suspension movement and speed. The wind forces are taken into account for vehicles with a higher center of gravity.

After all the external forces are accounted for, the equations that define movement can be summarized into the following equation (7) from [9, equation (30-32)].

$$\begin{cases} I_{x'}\dot{\omega}_{x'} = \Sigma M_{x'} + I_{z'}\omega_{y'}\omega_{z'} + I_{y'}\omega_{y'}\omega_{z'} \\ I_{y'}\dot{\omega}_{y'} = \Sigma M_{y'} + I_{z'}\omega_{x'}\omega_{z'} + I_{x'}\omega_{x'}\omega_{z'} \\ I_{z'}\dot{\omega}_{z'} = \Sigma M_{z'} + I_{y'}\omega_{x'}\omega_{y'} + I_{x'}\omega_{x'}\omega_{y'} \end{cases} \tag{7}$$

The program provides several collision models, where the user chooses one to run the simulation, the most used it's the momentum-based collision model using the 3D model of the vehicle to simulate the collision. Other can be chosen like multibody, Ellipsoid or mesh.

- Momentum Based Collision Model

  It's assumed that the impact forces are concentrated in a single point called the impulse point, instead of resolving the impact forces over time, only using time integral called momentums. The impact consists of two phases, compression, and restitution. At the end of the compression phase, both vehicle speeds are equal to the impulse point and the elasticity of the vehicle will make them separate. Both 2D and 3D models are available in the impact simulation, changing only the coordinates system. All 3 components of speed and rotational velocity are accounted for.

- Full Impact model

  In case of a full impact, two additional assumptions are added, this being that there is not any relative movement between both vehicles in the impulse point at the end of the compression phase and that the averages between the compression and restitution momentum are defined by the coefficient of restitution.

- Sliding impact model

  In some collisions, both vehicles will never achieve equal velocities at the impulse point and, in those cases, a contact plane needs to be defined along the direction where the vehicles slide and must coincide with the impulse point. As such there are two assumptions to be made: there is not any movement in the normal direction to the plane at the end of the compression phase and the direction of the moment is limited by friction. Good overlap between both bodies will yield better results.

- Force-based impact model

  The impact forces in both vehicles are calculated with contacts between ellipsoids. This means that the geometrical shape of the vehicle is formed by a series of ellipsoids that have a certain friction, restitution, and stiffness value.

- Ellipsoid to ellipsoid contact

  Contact between ellipsoids is defined by assuming the impulse point has always been on a line between 2 points belonging to each ellipsoid. The tangential planes to each point are parallel and at a minimal distance. The exact collision point can be calculated with the stiffness value of the bodies.

- Ellipsoid to plane contacts

  In this case, the planes are part of the 3D environment and they do not deform, it is instead calculated using a tangential plane parallel to the contact plane.

- Mesh model

  The geometrical shape of the vehicle is formed by a mesh of knots and lines, forming faces. The impact force is calculated for each adjacent knot based on its deformation. Each knot presents different values of stiffens, friction, and deformation.

- Vehicle to ground/ Slope/ Polygons

In case of contact with unmovable objects, the contact is calculated through the deformation of each knot knowing that the plane doesn't move.

- Vehicle-to-Vehicle contacts

  In a collision between two vehicles, one will be categorized as a master and the other as the slave in which its deformation values will be asserted by the contact with the master vehicle. Each iteration changes the categorization of each vehicle alternately.

As for the multibody, these are generally reserved for more complex simulations in which the simulated body do not behave as a rigid body, such as motorcycles, bicycles, and humans. Multibody are small bodies connected through each other with joints that can move and brake to a certain point, able to model complex geometry and have realistic kinematic simulations.

2.3 VEHICLE DAMAGE RECOVERY

With the EES, explained in Chapter 2.2.2, being a good way to categorize the energy involved in a crash, it still needs to be calculated by assessing vehicle damage, thus some methods to access vehicle damage will be studied.

2.3.1 *Constant Stiffness and Linear Spring Model*

For a barrier crash test [10], and certain width of the vehicle (W), the crushing energy (C) and velocity (V) have linear behavior, therefore the assumption that the vehicle behaves like a spring, and as such has a certain stiffness value for a length of (B) and when the force is zero there's a defined parameter A, resulting in the equation (8). By integrating along the whole width of the crash, from W_0 to W, the model can be extended to support non-uniform crush, adding a constant of integration G which can be calculated with equation (9) retrieved from the kinematic energy formula. The result is equation (10).

$$\frac{F}{w} = A + BC \tag{8}$$

$$G = \frac{A^2}{2B} \tag{9}$$

$$Ed = w\left(AC + \frac{1}{2}BC^2 + G\right) \tag{10}$$

For a more widespread method, an offset is necessary to compensate for the breaking point between the plastic and elastic deformation and add a correction factor for oblique collisions, which results in equation (11).

$$Ed = \frac{1}{2}B\left(C + \frac{A}{B}\right)^2 w\left(1 + tan^2\theta\right) \tag{11}$$

This only works for one point of measurement, meaning that it would not be feasible for the majority of crashes. Fortunately several points of measurement can be added to further describe the deformation of the vehicle, as an example:

Equation (12) for two points of measurement.

$$Ed = w\left(G + \frac{A}{2}\left(C_1 + C_2\right) + \frac{B}{6}\left(C_1^2 + C_2^2 + C_1C_2\right)\right)\left(1 + tan^2\theta\right) \tag{12}$$

Equation (13) for four points of measurement.

$$Ed = \frac{w}{6}\left(6G + A\left(C_1 + 2C_2 + 2C_3 + C_4\right) + \frac{B}{3}\left(C_1^2 + 2C_2^2 + 2C_3^2 + C_4^2 + C_1C_2 + C_2C_3 + C_3C_4\right)\right)\left(1 + tan^2\theta\right) \tag{13}$$

2.3.2 *Neural Networks for EES*

Neural Networks [11] are powerful mathematical tools that are able to establish a non-linear connection between a series of inputs to one or more outputs. This can be extremely advantageous and has been proved beneficial in a variety of other fields, therefore, in the crash reconstruction theme, this technology can facilitate and improve the measurements of EES.

This approach is already implemented in PC-CRASH™ where the software comes with an option to import a number of photos of the damaged area, to which the neural networks compare those images with the ones from an extensive database and gives a probable EES for that crash. More photos means that the error factor in the comparisons decreases, improving the resulting EES.

2.3.3 *Visual Inspection*

This is the most empirical method and can only achieved by some technicians with previous knowledge about the events that lead to certain types of damage.

The majority of vehicle accidents reconstruction data can be quite scarce, therefore the technicians need to make deductions according to the damages present in the vehicles bodywork such as: points of contact, the direction, scratches, tire marks, dents, and other types of damage. Then an educated guess can be made on how the events before the crash took place.

For example, wheel marks on the side of the vehicle can help predict the difference between the velocities of both vehicles since the ellipses marked on the side have different radii depending on speed.

Scratches on the vehicle's panels can demonstrate in which direction the impact came through. By analyzing changes in fading on scratched paint, being more pronounced in the initial impact than after the contact is made.

Normally pictures are taken with a reference ruler, this tool has a big scale that can be seen from afar. By comparing the photos from a crashed car with the original, damage can be easily accessed by comparing the two.

2.4 ROAD ACCIDENTS INVOLVING PEDESTRIANS

The most concerning type of vehicle accidents are the ones involving pedestrians or cyclists since they don't have a survival cell around them to protect from the impact. This gives the pedestrian's body an unpredictable movement after the impact, so some methods have been created to assess the dynamics and causes of such accidents and correlate the pedestrians injuries and the parameters of the crash.

It's important to acquire the most information possible like the final body position, impact point, and other points of interest, especially on the vehicle, like dents and other traces, that can help discover the impact angle. Since, in the majority of the cases, the pedestrian is walking, the body will have some moment at the point of impact therefore the body is going to be caught by the front of the vehicle and projected forward, or over the top.

Analytical methods can work depending on the parameters of the accident, and are relatively easy methods to apply to a single body. The following methods [12] can be applied to a pedestrian to find the impact's velocity. μ represents the drag coefficient between the body and the road, h the pedestrian center of gravity's height, d_t the throw distance, and g the gravity's accelerations. Schmidt and Nagel, at equation (14) is used to calculate the speed that correlates the thrown distance with the pedestrians h while Searle, at equation (15), finds a maximum and minimum velocity correlating to a certain throw.

- Schmidt and Nagel.

$$v = \sqrt{\mu^2 h + 2\mu g d_t - \mu h} \tag{14}$$

- Searle.

$$\begin{gathered} v_{min} = \sqrt{\frac{2\mu g d_t}{1+\mu^2}} \\ v_{max} = \sqrt{2\mu g d_t} \end{gathered} \tag{15}$$

However these types of reconstructions can prove to be complex to solve using only the analytical method, therefore some assistance from 3D biomechanical models of the human body is required. These types of analysis can be expensive, so a computer simulation using a multibody approach is usually the norm. This type of simulation consists in creating a body geometry with several smaller parts, with each part constrained to its neighbors, which makes it possible to approximate the body's behavior to real life.

The multibody approach [13] proves to be easy to implement provided that the conditions are known and the quality of the simulations are excellent, although getting the exact parameters that match the real-life crash can be quite inconsistent.

2.5 ROAD ACCIDENT ANALYSIS METHOD

Nowadays the Accident Analyst job is to interpret the outcomes of an accident and deduct the events that lead to the impact. This process has several to achieve something that can be presented in court or insurance firm, which is the main objective for the analyst. It must be noted that the data presented in such cases needs to be shown as a possibility of events and not as the truth.

The first contact is made by the authorities and, assuming that the victims are in safety, it begins by to make appointments of the relative positions of several points of interest around the accident scene which are called a sketch. Relative positions are used because the authorities use a fixed point from where all measurements are taken from, drawing them, not to scale, in order to represent to its best the accident scene. They also collect information and interview everyone that intervened in the crash, writing detailed testimonials about the crash and take enough photos to document the event.

These points of interest can be pieces of evidence like asphalt marks, blood marks, vehicle parts, tire marks, final vehicle positions, and other types. Normally two measurements are taken to help place them in a xy plane and eliminate ambiguity in the direction of the object or its placement. More measurements can be taken, but taking only one can hamper later processes. This information is later forwarded to the responsible personnel.

If the case requires, the analyst will receive this information and will review the data while adding its own. It can range from going to the place of the accident and retrieving more information from there, to finding the crashed vehicles and taking more photos of the damages, and inspecting them to a greater degree. This process is time sensitive since if the analyst takes too long to retrieve more data, said data can cease to exist because the vehicle will eventually be repaired or scraped.

With all the data gathered, the job of the analyst starts by studying the evidence and coming up with all hypotheses to the sequence of events from that data, then collecting information about the vehicles and analyzing their damages using the several methods described in section 2.3. Then the analyst tries to deduct the placement of the vehicles moments before the impact and corroborate it with any of the hypothesis created before. If more then one seems probable it's good practice to check them all in the computer simulation.

With the research done, the analyst must create a computer simulation that follows the data to its best. Most of the process relies on the ability to make the vehicles take the impact in a specific way so that the final positions match the real world events. *Software's* like PC-CRASH™, mentioned before, allows great control of the vehicle before the crash and several modes to calculate the impact alongside an extensive vehicle database to make sure the simulation is as close to the real world as possible.

The analyst then redacts a report exposing its findings from all simulations in the most objective way, the report is sent to the relevant personnel that might accept, or not, the study, depending on how thorough and acceptable the analyst work was.

2.6 AUTOMOTIVE ELECTRONICS

Since the beginning of the automotive industry, the technology surrounding the vehicle has evolved at a fast pace. Several sensors, actuators, and processing units were added for more efficient and safer operations. Control units need a way to send information around the vehicle to different control modules fast and reliably, since disturbances and vibrations of the surrounding environment can cause interference. If this system were to fail it, could put the lives of the driver and occupants in danger.

2.6.1 *Intra-Vehicular Networks*

A vehicle can have two types of networks. Intra-Vehicular Network refers to the communication that happens inside the vehicle, between the internal control modules, while an Inter-Vehicular network refers to the communication that happens from the vehicle to the outside, like radio and GPS communication.

In the beginning, due to simpler technology, each sensor and actuator required an individual connection to the drivers interface inside the cockpit or engine bay. This way became unsustainable when the number of devices increased dramatically, increasing the wiring harness in size, weight, cost, and complexity which would make any diagnostic of the electronics a nightmare.

Vehicle networks came to solve this problem, with only one ECU (Electronic Control Module), managing each subsystem of the vehicle and additionally having all the ECUs interconnected. At first, a mesh style connection was used with each ECU connected directly to another. Later on evolving to a centralized BUS style of communication. Initially, every constructor presented a different solution to the same problem, increasing the markets complexity, therefore some standardized methods [14] were introduced, and the most used ones are described as follows.

- Controller Area Network

  The CAN (Controller Area Network) Bus [14, Chapter 2.2.2.2] uses a pair of wires (CAN High and CAN Low) to share information within the vehicle, sharing the same wires as shown by figure 5. Both wires transmit the exact same message, although one of them is inverted. This allows them to send through long distances without signal degradation. The ECU can decode the message using the difference in voltage between both wires.

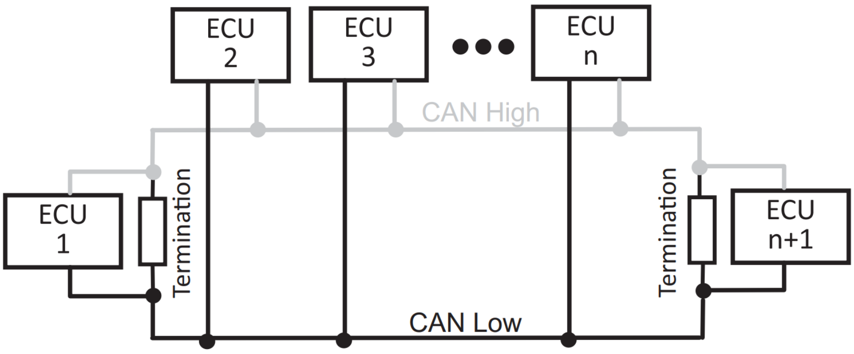


Figure 5: Controller Area Network, adapted from: [14, Chapter 2.2.2]

A CAN frame can be viewed in figure 6. It includes a *header* where the identification number from the sender ECU and information about the message resides, a *Data field* which, depending on the version of CAN used, can transmit messages of 8 bytes or 64 bytes. CRC (Cyclic Redundancy Check) is used to detect errors within the message and the Acknowledge is a single bit that each ECU sends to prove that the last message was received.

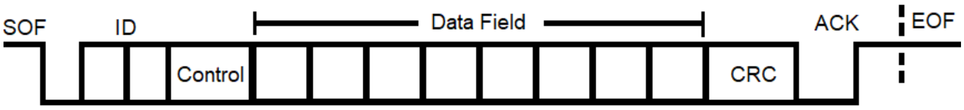


Figure 6: Structure of a CAN Frame

CAN bus works using priorities given by the message ID, which means that there is a possibility of lower priority messages getting locked from transmitting. Usually, the solution involves designing the CAN for a maximum load of 50%. The speeds used in this communication are usually between 125 kbps and 2 Mbps although it can get faster, or slower, with different protocols.

- Local Interconnected Network

  Some components inside the vehicle only require simple sensor-actuator control schemes, like climate-control systems, to which CAN would be overkill, therefore the LIN (Local Interconnected Network) bus uses a simpler frame and only one wire to control several ECUs shown in figure 7.

  Using a LIN Bus, a *Master* ECU can control up to 16 other *slave* ECUs. The master ECU is the only one who can start communication and indicates to the slaves what to do. Message Scheduling is made beforehand. In a vehicle architecture, it is normal to encounter ECU connected to a CAN bus that in turn uses LIN to control several other peripherals.

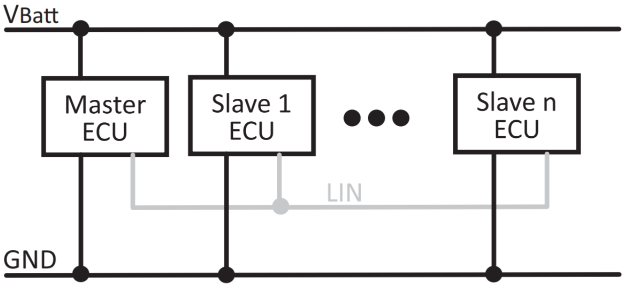


Figure 7: Local Interconnected Network, adapted from:[14, Chapter 2.2.3]

- FlexRay

  FlexRay [14, Chapter 2.2.5] came to answer the excessive need for electronic communication that newer devices and control modules require, especially for X-By-Wire applications which removes the old mechanical link in safety-critical uses and replaces them with electronic control. As such reliability is the priority, a driver can not lose his steering, braking, throttle, etc. when driving.

  This system uses cycles and time slots, each cycle consists of a static time slot and a dynamic one. A static time slot is reserved for ECUs that require constant use. If it has nothing to transmit, it will output a ”null” frame so the system won’t think the connection was interrupted. In the dynamic time slot, the ECUs will wait until they have something to transmit and, once they do, it functions similarly to a CAN bus.

  A Flexray packet is similar to a CAN frame shown in figure 6 having only different configurations in the *header* and a 0-256 byte *Datafield*, CAN bus only has up to 64 bytes, as such it can achieve speeds up to 10mbps.

  In a FlexRay network the ECUs are connected sequentially with the same procedure as in a CAN bus but able to get 2 pairs of wires for redundancy. Due to the higher bitrate required, FlexRay can only connect 5 or 6 ECU in series before the signal starts to decay. As such there’s the possibility of multiple topologies with the use of the *Active Coupler/Star Coupler* that enables the connection of several series topologies in parallel to make one big network as shown in figure 8. The coupler copies the data from one branch to another with relatively small latency.

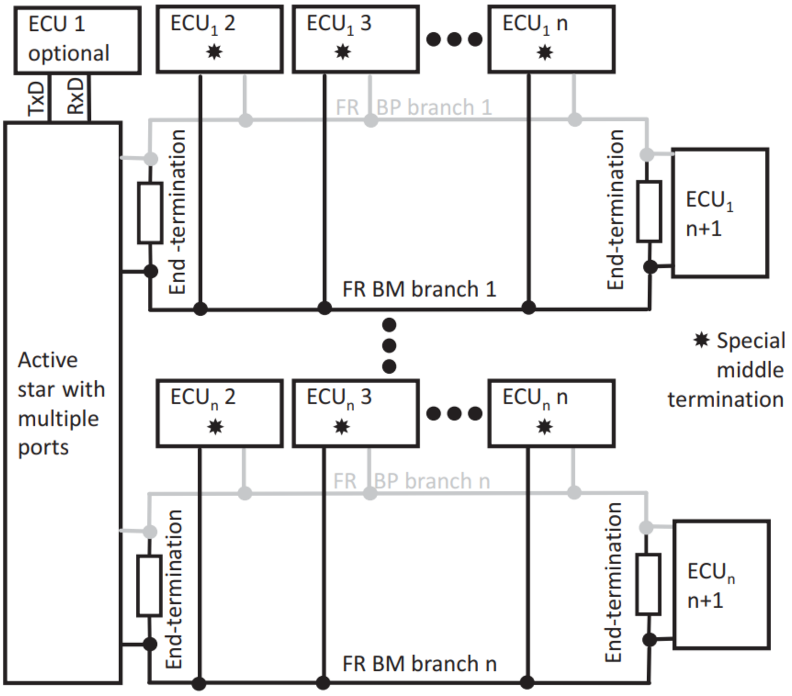


Figure 8: FlexRay, adapted from: [14, Chapter 2.2.5]

These are some of the most used networks for Intra-Vehicular communication with other methods reserved for infotainment and other applications. A comparison between network properties will be presented in table 1.

Table 1: Properties of Intra-Vehicular Networks [14, Chapter 2.2.8]

Technology	Paradigm	Bitrate	Main use case
CAN	Priorities	0.125-2 mbps	ECU control
LIN	Master-Slave	19.2 kbps	Low-Cost Control
FlexRay	Flexible	10 mbps	Real-time control, X-by-Wire

As for the future, these networks might change with the introduction of ADAS (Advanced Driver Assistance Systems), being more present every year along with the use of more optical sensors, Radars, and LiDAR can push the contemporary networks to their limits, hence new technologies are building upon the Ethernet standard that is already in use in global communication, with the *Automotive Ethernet* promising to solve these limitations [14, Chapter 3.1] for media and diagnosis.

2.6.2 *On Board Diagnostics*

With the increasing computerization of vehicles, the maintenance and repair tasks, which were previously all mechanics, did not offer an easy way to diagnose the electrical components. OBD (On Board Diagnostics) was created to diagnose the vehicle's problems before they happen and understand why they happened. OBD was the first generation which was originally externally connected to the vehicle while its second generation, OBDII, is integrated. OBDII became a standard and mandatory for all vehicles post 2001 in USA and 2003 for the EU (European Union). OBD can access several systems, like powertrain or Chassis control, and provide different DTC (Diagnostic Trouble Code) for each subsystem. Some of these DTCs are standard but the manufacturers have liberty to add their own. Nowadays the diagnostic is done with a simple diagnostic tool connected to the standard OBDII connector and accessing a database with all the vehicle information and DTC.

2.6.3 *The Automotive "Black Box"*

Similar to a plane, vehicles can also have a "black box" that records the dynamics and driver inputs, although it goes by the name of *Event Data Recorders*, EDR. Those devices have been around since the 70's [15] with General Motors being at the forefront of this technology. At that time it was used as a way to test airbag deployment. With this in mind, it only required to run an algorithm that checks whether the accelerometers data is above a certain threshold. This later evolved into the ACM (Airbag Control Module). Previously only OEMs (Original Equipment Manufacturer) could read the data stored inside the ACM and it only became available to the public to retrieve that information in the 2000s. this created a new way to perceive accidents from a vehicle dynamics point of view, since the driver may not recall correctly what happened. Later on, with the increasing number of ECUs inside the vehicle and a more robust Intra-vehicular network more information became available for the EDR to store.

Nowadays the presence and free access to the EDR data became required by law, with the United States of America being the driving force and now the European Union following suit in 2022 with a certain number of data points to be recorded and read required by law.

2.7 EVENT DATA RECORDER

As previously discussed EDRs are an emerging way to investigate road accidents providing more information to the police force. This chapter will look at how it works, the tools required, and the expected output of an EDR.

2.7.1 *How the EDR Works*

As the name implies, an EDR will record vehicle data after some events are triggered. The main variable is the Δv, in other words, the total change of velocity that's sustained by the vehicle in a certain time frame, usually this threshold is specific to each OEM. The Δv is determined using accelerometers present inside the ACM measuring accelerations longitudinally and laterally. Since it is impossible for the ACM to know when an accident is about to happen, it's always recording the data to the EDR's RAM (Random Access Memory), normally between 5 seconds before the event T_0 (time of event), which can differ from OEM. After the algorithm is enabled, this set of "before the crash" data will be recorded to the ROM (Read Only Memory) of the EDR while also recording in a faster pace data after the accident. This data set is recorded up to 300 milliseconds after the accident.

These data points are provided by the several ECU connected via the Intra-Vehicular network, particularly the data from vehicle dynamics such as *pitch, roll* and *yaw* rates can be provided by the Electronic Stability or Traction Control modules. Data from the wheels such as speed and braking comes from the ABS (Anti-Lock Braking System) module and most Engine data comes from the Engine ECU as well as other modules present in the network. The acquired data is then stored in an EEPROM (Electrically Erasable Programmable Read-only Memory) inside the rugged box of the ACM. This means that it should sustain respectable physical damage before a loss of power, however, should the connector be damaged, it can make the imaging process more tricky. During the event, energy loss can occur depending on the type of accident or if the battery disconnects by force or design, which can interrupt the recording progress. Therefore inside the ACM, there is a set of capacitors that store energy continuously and can fire the safety measures and power the ACM long enough to store the data.

The variables required by the EU regulation [16] are present in tables 2, 3 and 4. Although this data is mandatory, there's nothing that opposes adding more data. The table will present each element such as: type of requirement (if it is mandatory or optional), the time interval the variable must be recorded, the rate of recording, the maximum sensor range, and minimum sensor accuracy. The elements that do not have a rate, mean that it's only written once at the time of crash.

Table 2: Elements required by EU regulation Part 1 [16]

Element	**Requirement**	**Interval**	**Rate**	**Range**	**Accuracy**
ΔV longitudinal	Mandatory	0 - 250 ms	100 /s	-100 km/h to 100 km/h	±10 %
Maximum ΔV longitudinal	Mandatory	0 - 300 ms	N/A	-100 km/h to 100 km/h	±10 %
Time, Maximum ΔV longitudinal	Mandatory	0 - 300 ms	N/A	0 - 300 ms	±3 ms
Speed	Mandatory	-5.0 s to T0	2 /s	0 km/h to 250 km/h	±1 km/h
Engine Throttle	Mandatory	-5.0 s to T0	2 /s	0 to 100 %	±1 %
Service Brake	Mandatory	-5.0 s to T0	2 /s	ON/OFF	N/A
Ignition Cycle @ Crash	Mandatory	-1.0 s	N/A	0 to 60000	±1 cycle
Ignition Cycle @ Download	Mandatory	At time of download	N/A	0 to 600000	±1 cycle
Safety Belt Status, Driver and Passenger	Mandatory	-1.0 s	N/A	Fastened/Not Fastened	N/A
Air bag Warning Lamp	Mandatory	-1.0 s	N/A	ON/OFF	N/A
Frontal Air bag time to deploy for driver and passenger	Mandatory	Event	N/A	0 to 250 ms	±2 ms
Multi Event Crash, Number of event	Optional	Event	N/A	1 or more	N/A
Time Between Events	Mandatory	As needed	N/A	0 to 5.0 s	±100 s
Complete File Recorded	Mandatory	Following Other Data	N/A	YES / NO	N/A
Lateral acceleration post crash	If Recorded	0-250 ms	500 /s	-50 G - 50 G	±10 %
Longitudinal acceleration post crash	If Recorded	0-250 ms	500 /s	-50 G to 50 G	±10 %
Normal acceleration	if recorded	-1 s to 5 s	10 /s	-5 G to 5 G	±10 %

Table 3: Elements required by EU regulation Part 2 [16]

Element	**Requirement**	**Interval**	**Rate**	**Range**	**Accuracy**
Maximum ΔV lateral	Mandatory	0 - 300 ms	N/A	-100 km/h to 100 km/h	±10 %
Time, Maximum ΔV longitudinal	Mandatory	0 - 300 ms	N/A	0 - 300 ms	±3 ms
Time, Maximum ΔV Total	Mandatory	0 - 300 ms	N/A	0 - 300 ms	±3 ms
Engine RPM	Mandatory	-5.0 s to T0	2 /s	0 - 10000 rpm	±100 rpm
Vehicle roll angle	Mandatory	-1.0 s to 5 s	10/s	-1080° to 1080°	±10 %
Vehicle roll rate	Mandatory	-1.0 s to 5 s	10/s	-240° to 240°	±5 %
ABS Activity	Mandatory	-5.0 s to T0	2/s	Faulted / Active / Intervening	N/A
Stability Control	Mandatory	-5.0 s to T0	2/s	Faulted / On / Off / Intervening	N/A
Steering	Mandatory	-5.0 s to T0	2 /s	-250° to 250°	±5 %
Passenger Airbag Suppression Status	Mandatory	-1 s	N/A	Suppressed / Not Suppressed	N/A
Frontal Airbag deployment time no nth stage	Mandatory if fitted	Event	N/A	0 to 250 ms	±2 ms
Side Airbag time to deploy	Mandatory	Event	N/A	0 to 250 ms	±2 ms
Side Curtain Air Bag time to deploy	Mandatory	Event	N/A	0 to 250 ms	±2 ms
Tire Pressure	Mandatory	-1 s relative to T0	N/A	N/A	N/A
Yaw Rate	Mandatory	-5 s to T0	2 /s	-75 to 75 degrees/s	±10 % of sensor range
Tire Pressure	Mandatory	-1 s relative to T0	N/A	N/A	N/A
Safety belt rear passengers	Mandatory	-1 s	N/A	Fastened/ Not Fastened	N/A

Table 4: Elements required by EU regulation Part 3 [16]

Element	**Requirement**	**Interval**	**Rate**	**Range**	**Accuracy**
Longitudinal / Lateral acceleration, pre crash	Mandatory	-5 s to T0	2 /s	-1.5 G to 1.5 G	±10 %
Traction Control Status	Mandatory if not fitted with ESC	-5 s to T0	2 /s	N/A	N/A
AEBS status	Mandatory	-5.0 s to T0	2 /s	N/A	N/A
(Adaptive) Cruise Control Center	Mandatory	-5.0 s to T0	2 /s	N/A	N/A
VRU secondary safety system time to deploy	Mandatory	Event	N/A	0 to 25 ms	±2 %
VRU secondary safety system status	Mandatory	-1.1 s to T0	N/A	N/A	N/A
Far Side Impact Center Airbag	Mandatory	Event	N/A	0 to 250 ms	±2 ms
Lane Departure warning system status	Mandatory	-5.0 s to T0	2 /s	N/A	N/A
Corrective steering function	Mandatory	-5.0 s to T0	2 /s	N/A	N/A
Automatically Commanded Steering function status (category: A, B1, B2, C, D, E)	Mandatory	-5.0 s to T0	2 /s	N/A	N/A
Accident Emergency call status	Mandatory	Event	N/A	N/A	N/A
Safety belt rear passengers	Mandatory	-1 s	N/A	Fastened/ Not Fastened	N/A
ΔV lateral	Mandatory	0 - 250 ms	100 /s	-100 km/h to 100 km/h	±10 %
Pretensioner, time to fire	Mandatory	Event	N/A	0 to 250 ms	±2 ms
Seat Track Position	Mandatory if fitted	-1 s	N/A	YES/NO	N/A
Occupant Size	Optional	-1 s	N/A		N/A

2.7.2 *Hardware*

To extract the data from an EDR a trained technician connects the imaging tool to the OBD connector, or directly to the module, in a process named *imaging.* This tool can either be manufacture specific, which greatly expands the amount of information but may not have compatibility with other brands, making the analyst buy one machine for each brand or having to send the ACM to the manufacturer, or be available to the public at a cost like the Bosch CDR 900, which is an extension of the Bosch CDR (Crash Data Retriever) kit.

The Bosch CDR 900 is a type of VCI (Vehicle Computer Interface) or Vehicle Computer Interface, and it has two modes of operation, DLC (Direct Link Connector) and D2M (Direct to Module). DLC consists in connecting the CDR 900 directly to the OBD port of the vehicle to extract data, while D2M requires the extraction of the ACM module to extract the data directly from it. The tool also asks for the VIN (Vehicle Identification Number) and checks if it's supported by the tool. Once the extraction sequence is started, the data is downloaded 3 times and checked automatically for errors. The information is sent using bits that are translated into measurable information through a translation protocol that defines where the value is in memory, how are they converted into useful information and the description of each value. After this, the information is stored in a file. This is sent to a computer that analyzes it using specific software, presented in chapter 2.7.3, and an easy-to-interpret report is generated.

The versatility of the tool and the ability to plug into any OBD connector to retrieve the data makes it an easy way to implement in today's accident reconstruction technicians. However, due to the lack of regulations pre 2022, at least in Europe, the constructors were not forced to make the data readily available, this means that some vehicles might not be compatible with the tool and data can not be retrieved.

The D2M procedure consists in removing the ACM (or another compatible) from the vehicle and using a specialized connector to link the VCI to said module. This process is more time-consuming since it requires the extraction of the module from the vehicle, and some paneling must be disassembled. But this method requires extra caution since the module is powered outside the vehicle and a major hit on the module can start an event and overwrite the old one. Also the lack of standardization of the ACM modules means that several cables for each vehicle are required if a direct-to-module connection is required.

2.7.3 *Software*

The CDR tool has a companion software that allows the user to interface with the VCI, make readings, update VCIs firmware, and generate the report, among other things. Upon having a legitimate license file and registering the VCI, the software is ready to make readings. Once connected to the vehicle, the user makes a new reading in which it has to specify the current vehicle brand and VIN, which then needs to wait if it's supported or not. If it is, the user must make a choice reading from the ACM, PCM (Powertrain Control Module), PPM (Pedestrian Protection Module) or ROS (Roll Over Sensor) data. The software will then extract the data and generate a user-friendly report which then can be saved as a PDF or as a ".csv" file.

One of the most useful features of this software it's the extensive help page. Here the user can find all the supported vehicles with respective information about which module the data is recorded. The requirements for connection such as DLC adaptor, and respective cable and adapter for a D2M connection, location of the module inside the vehicle, and information about the supported markets, since one vehicle can be technically supported but may be in an unsupported market. Also here there is a step-by-step tutorial and guides for connections and reading procedures available, what cautions one should have, and troubleshooting among other things.

2.7.4 *Data Output*

As explained in the previous point, the data can either be exported as a ".csv" file with the raw data of every datapoint or in a PDF format which exports a report with the data transformed into graphics and tables, completed with the corresponding data limitations of the EDR. The layout of the PDF report generated by the *software* is as follows:

"CDR File Information" is the first page, the user can find information such as: *software* used, its user and respective license, the information introduced on the computer side like the crash data and current data along with the VIN, the type of tool used and EDR, some comments written by the user and some data limitations regarding this particular vehicle's EDR and information about the data sign convention.

"System Status at Event" shows the information stored inside the vehicle's EDR about itself and some characteristics of the crash. Because some accidents record several events, this information repeats for each one of them, unless a separate category is created to represent fixed data as some do.

- VIN:

  Here is stored the real VIN which can be compared to the input VIN and determine if the vehicle has been tampered with and the EDR has been switched.

- Ignition cycle at Download and Event:

  Indicates the times the ignition has been switched on. When reading, it might not be exactly the same number since the ignition of a vehicle can be turned on or off in the time between crash and reading. However, if there's a big discrepancy between both these numbers, there's reason to suspect tampering.

- Number of Events and time between them:

Since some accidents trigger multiple events, this indicates the time between the current event and the last can help place each event on the timeline.

- Vehicle and ECU information:

  Shows the Mileage, Operating time, ACM part number, software and hardware version, supplier, and other information.

- Algorithm information:

  Shows if the Algorithm to detect frontal crashes, rear crashes, lateral crashes, and rollovers were started or reset.

- Δv information:

  Shows the maximum lateral and longitudinal Δv sustained by the vehicle and for how much time.

- Complete File Recorded:

  Tells the user if the EDR recording was completed or interrupted during the crash.

”Deployment Command” saves information about the time required for all the types of restraining measures such as driver’s frontal, side, and knee airbag and respective pretensioner, front passenger’s frontal airbag and pretensioner, side curtains (both sides), and battery disconnect.

”Pre-Crash Data -1 sec” shows the information recorded one second before the crash which consists of the status of safety belts, whether they’re fastened or not, and information about the airbag status for each event.

”Pre-Crash Data -5 to 0 sec” presents, in a table, all the mandatory data specified in the tables 2,3 and 4 from -5 to T_0 for each event. This varies from one vehicle to another, particularly in older vehicles not covered by the new regulations.

”Crash Pulse” shows both longitudinal and lateral acceleration pulses during the impacts 300 ms timeframe in both graphical and tabled format for each event.

”Roll Angle” presents the vehicle’s rotations and velocity along the x axis in both graphical and tabled format for each event, helping in the analysis of roll-over situations.

”Data Dump” is the final portion of the report and consists of the raw data stored inside e EDR’s EEPROM in Hexadecimal data.

Other types of data will be explored further while talking about vehicle dynamics in Chapter 3.

2.8 VEHICLE DYNAMICS MODEL

Vehicle Dynamics is a group of forces and moments that acts on the vehicle and defines its stability on the various subsystems of a vehicle. The most important of which is the tires, since these are the elements that make contact with the ground and define the whole vehicles movement and dynamics, such as turning, accelerating, braking, sliding, etc. This can be influenced by the driver's input, the vehicle's geometry, gravity, aerodynamics, and road condition.

The vehicle can be divided into two subsystems, a sprung mass, and an unsprung mass. The first one defines the chassis, bodywork, engine, engine and everything above the suspension, itself included, while the unsprung masses are the wheels. The suspension allows for a somewhat free movement in all axis, this means the momentum along the yy axis, defined as pitch, xx axis, defined as roll and zz axis, defined as yawn, is enabled while always maintaining full contact with the road and absorbing all the irregularities present on its surface.

When a vehicle is doing any kind of maneuver, its weight shifts around, for instance, braking shifts the weight forwards and accelerating shifts it backward, which may lead to some wheels having their contact points weakened, or outright lost. This leads to unexpected vehicle movements like understeer, oversteer or slides. Understeer happens when the front end of the vehicle loses traction and the vehicle turns less than intended and oversteer happens when the back end loses traction and the vehicle turns more than intended. In certain cases, all wheels may lose traction due to external forces.

There are several ways to simulate the dynamics of a vehicle [17][18] and the complexity of the model normally correlates with how realistic it will be.

2.8.1 *Tire Model*

For a long time the research of tire behavior [18] and handling has been studied and emulated with several different models with different approaches. These can be grouped into two different categories: empirical models with a lot of full-scale tire test data and theoretically models developed with structural behavior using finite element techniques.

Empirical models are over-parameterized and may be difficult to use without sufficient data while theoretical models describe the steady-state and transient behaviors in great detail.

2.8.2 *Vertical Dynamics Models*

The simplest form of this model [17] emulates the dynamics with the suspension system, a quarter of a vehicle. This model has only two degrees of freedom and its represented by a spring-mass-damper similar to the system shown in figure 9.

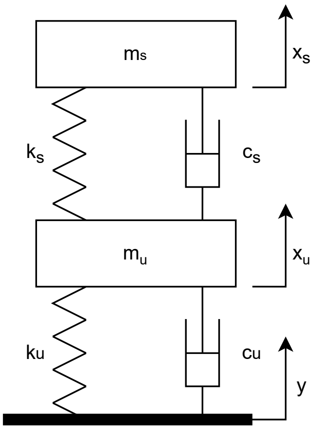


Figure 9: Vertical Dynamics Model, Quarter vehicle

The superior mass is the respective weight of the vehicle at that point. This mass is Attached to the lower mass, the tire, with a spring and a damper representing the suspension system. Between the tire and floor a spring and damper system simulating the natural proprieties of rubber tire wall, that also has energy absorption properties.

By adding another similar system alongside it, one can emulate two wheels and pitch/roll, depending on each wheel is added. This can be doubled furthermore and emulate all four wheels of a vehicle and 7 degrees of freedom, rotations in every axis and displacement of each wheel, as seen in figure 10.

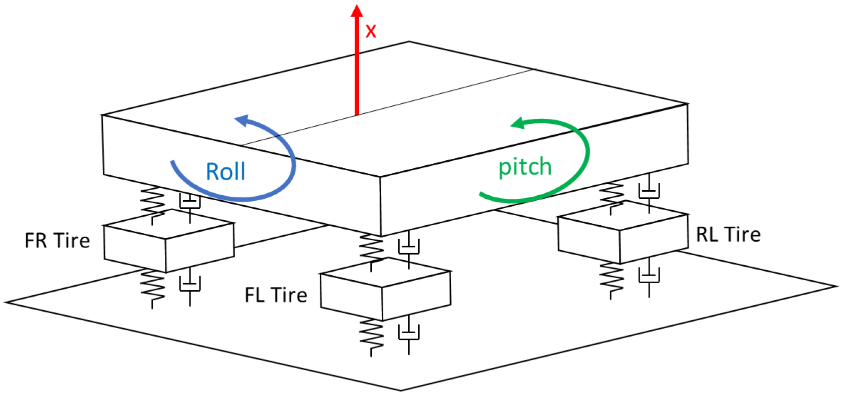


Figure 10: Vertical Dynamics Model Full Vehicle

But this is only a vibration models, used to test the vehicle suspension response to outside forces.

2.8.3 *Bicycle Model*

This is one [17][18] of the simplest model to exhibit planar motion, reducing the vehicle into a two-wheeled model, with front and back wheels, with three degrees of freedom, lateral and longitudinal displacement, and rotational movement around the normal axis, as shown in figure 11 This model is more inclined to study the trajectory of a vehicle during a turning maneuver or control methods for automated driving.

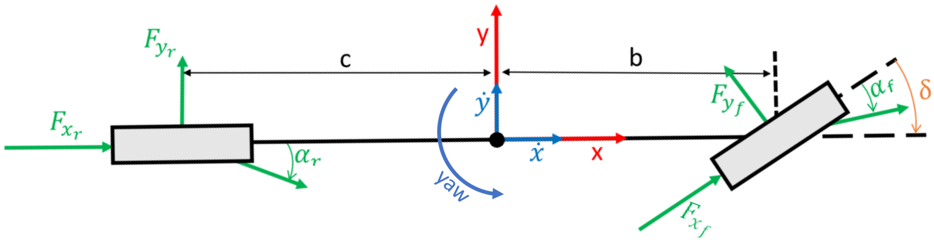


Figure 11: Bicycle Model Vehicle Dynamics

2.8.4 *Two Track Model*

The model represented in figure 12 is the upgrade over the bicycle model [18] and it takes into account out-of-road plane and in-road plane motion. It also presents 14 degrees of freedom, lateral and longitudinal displacement, rolls pitch and yaw rotations, tire rotations and displacements, giving a good representation of the real vehicle and suitable for integrated control.

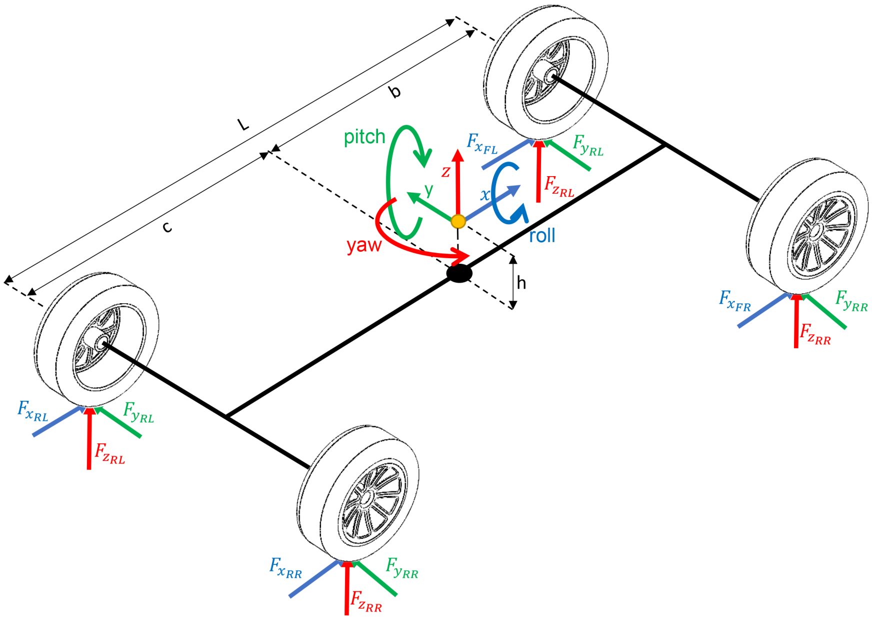


Figure 12: Two Track Vehicle Model

Applying Newton's equations for the equilibrium of forces and momentum, the vehicle's velocities and accelerations can be calculated. The main forces on the vehicle are generated by the tire-road contact and must be balanced with the inertial momentum.

The equations presented in system (16) can be used to calculate several dynamic parameters and some may require characteristics of the suspension cataloged.

$$\begin{cases} \sum_{wheel=1}^{4} F_{xi} + F_{drag} + F_{rollingres} + F_{slope} + F_{ext.x} = m * a_x \\ \sum_{wheel=1}^{4} F_{yi} + F_{ext.y} = m * a_y \\ \sum_{wheel=1}^{4} T_{zi} + = I_z * a_z \end{cases} \tag{16}$$

3

DEVELOPMENT

3.1 PROJECT ROADMAP

As previously presented in chapter 1 this project aims to improve the accident reconstruction method by adding to what's already being done nowadays. As shown in chapter 2, the several tools and methods, both analytical and computerized, to recover the information from the crash scene and its participants, and now with new regulations the implementation of the EDR data and making the process much more streamlined.

To achieve that, all the data from the Accident scene, Vehicle Information, Damage, Pedestrian Data, and testimonials along with the new EDR Crash Data. This datapoints need to be organized as inputs of a computer program. Said program needs to know how to treat each information accordingly, and this information can be separated into two different groups, scene data, and vehicle data. These inputs will be taken into account to achieve a reliable accident reconstruction. This process can be exemplified in the diagram in figure 13.

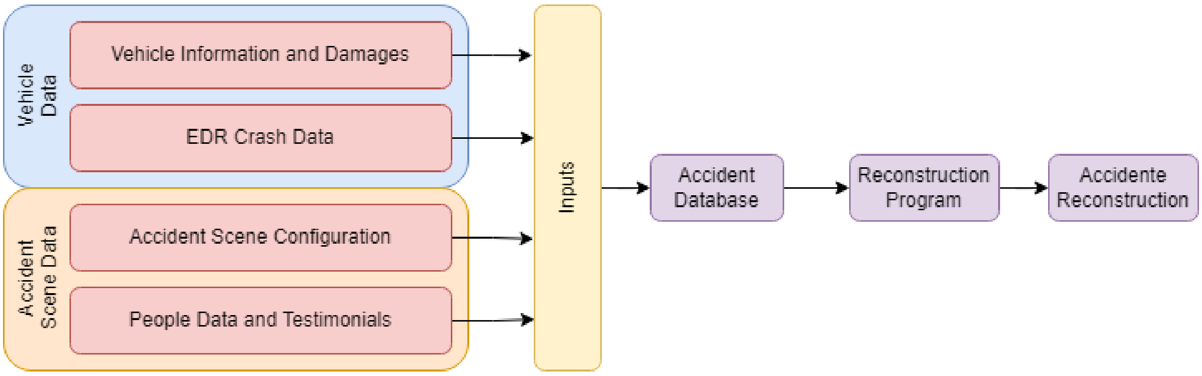


Figure 13: Project Roadmap

3.1.1 *Accident Scene Data*

The Accident Scene data covers everything from the location of the accident, its geography, and characteristics of the road, with the additions of people's testimonials and other types of evidence found. It's by far the most empiric type data, since the only hard data measured is the road's physical properties and layout, everything else like the location of debris, probable points of impact, and other road damages or marks comes from intuition and witnesses. Some methods to catalog and better recreate digitally the environment around the accident will be explored.

3.1.2 *Vehicle Data*

This data encompasses everything that can be obtained from a vehicle. Such as the physical damage sustained and its analysis, exemplified in chapter 2.3, the configuration of the vehicle that affects the dynamics of the vehicle such as wheelbase, weight, vehicle type, height, etc. And now the data recorded from the EDR. Some methods to catalog and recreate the vehicle's movement through dynamic data will be explored.

3.1.3 *Objective*

As explained beforehand, the objective of this project is to create a new tool that brings all these elements together and streamlines the workflow for accident reconstruction, starting from the data recovered from the first contact, to recreating the movement of the vehicle.

3.2 ACCIDENT SCENE CONTEXT

Once the authorities arrive at the location, their primary objective is to recover most of the available data from witnesses and pieces of evidence around the crash. This initial contact is indispensable to crash reconstruction since it gives the context needed to frame the subsequent EDR data.

3.2.1 *Scenario Sketch*

The process of creating a sketch consists in selecting one or more fixed points of reference and measuring the distance of a specific point of interest. This way the authorities can easily describe the accident scenario, and the quality of the data is improve by taking more measurements or using more fixed points of reference. One flaw of this system is subject to the skill or time availability of the person doing it, meaning sketches will differ a lot from one to another with some missing crucial information.

The measurements are usually made using tools mentioned above, in chapter 2.1, and while there's a lot of variety to choose from, the great majority of them are made with a simple measuring tape. This tool its the most readily available tool that still gives good enough results for what is pretended, while the others may be too cumbersome to carry around anywhere and maybe the end doesn't justify the initial investment necessary to give this equipment for everyone. This means that more expensive and specialized tools are reserved for technicians who specialize in accident reconstructions.

Although this doesn't mean that the method can't be improved, therefore this project proposes the use of a tool that catalogs the accident scene in a coordinate system (x,y and z) by using points and lines, and to achieve that a relay system is proposed.

This system consists of a device that, through the relative motion between its starting position and end position, calculates the distance between those two points. This is achieved with the use of accelerometers and gyroscopes, and assuming the device starts in a stationary state, the act of moving the device starts the algorithm that takes measurements along the trip until the device stops. With this method, the total displacement of the device can be calculated. Due to the cumulative error from gyroscopes and accelerometers, this method is only feasible for short distances, however, this drawback can be eliminated by adding relay stations similar to a cellphone network or GPS system. The position of the device can be triangulated using three or more fixed towers radio, those then communicate with the device and by knowing how much time a "ping" takes to travel from the one tower to the device. Since the three distances are known, triangulation can take place. The measurements are then cataloged by user input so that when the software reads from the generated file, the sketch is automatically generated.

3.2.2 *Accident Scene Scanning*

For some cases, if a more refined accident scene is required especially if the end product is a video recreation of the accident, and for that, a 3D render of the scenery is the most efficient way to achieve this. As explained in chapter 2.1, processes such as photogrammetry, 3D scanning tools, and LiDAR allow us to do such things. In table 5 we can see a rating of these tools considering their initial and operating prices. If a tool is cheaper means its more availability and more technicians can afford it. Expertise means the skill level and understanding the Technician must have to use the tool and recover useful data, and for last safety which ranks the road traffic, or other dangers, the technicians must expose themselves to in order to capture the accident scene.

Table 5: Comparison between scanning processes [1]

Process	**Price**	**Expertise**	**Safety**	**Rating**
Photogrammetry (Close Range)	5	4	3	4
Photogrammetry (Drone/UAV)	5	3	5	4.3
3D Scanning	2	2	3	2.3
LiDAR	3	2	3	3
On a scale from 1 to 5, with 5 being the best overall				

Looking at the table 5 we can see that both photogrammetry processes are the best choice for a cheap, easy and safe way to scan an accident scene, most notably the UAV variant of the process. With this process, a technician only has to set up a flight path for a drone, keeping in mind its surroundings, and then the UAV will go on its way and take as many pictures of the accident scene as necessary, this way the technician doesn't even have to interact with the traffic flow.

The batch of images that comes from the UAV will can be processed by a photogrammetry program. This program can either be open source for personal use or commercial solutions from engineering and industrial applications, either way, the process is the same, the program will look into the batch of photos and, by comparing the overlapping photos and analyzing its surroundings, merge them in such a way that a *pointcloud*, similar to the figure 14, is built. This way a specified amount of points, depending on the resolution pretended, are positioned in a space which means each point has a set of cartesian coordinates (x, y and z) and, sometimes, color information, this points can be turned into 3D models by connecting the points into polygons making a surface which then can be exported into a plethora of file types for different CAD (Computer Aided Design)*softwares.*

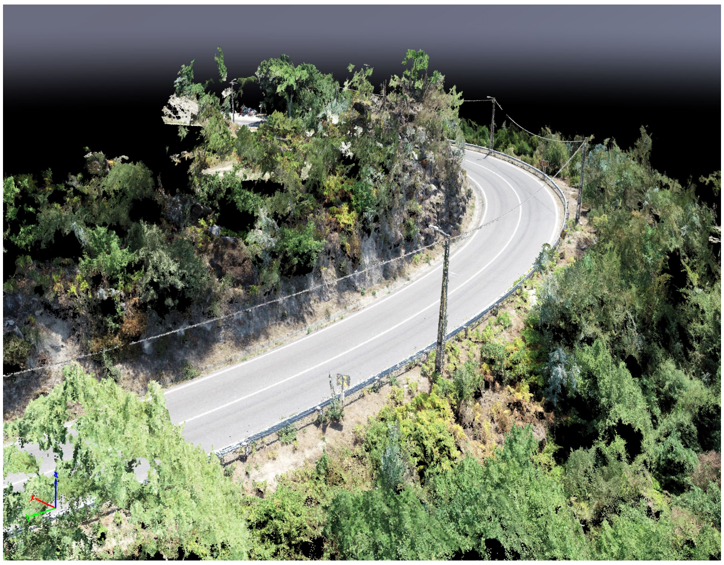


Figure 14: Example of a 3D *pointcloud*

3.2.3 *Implementation*

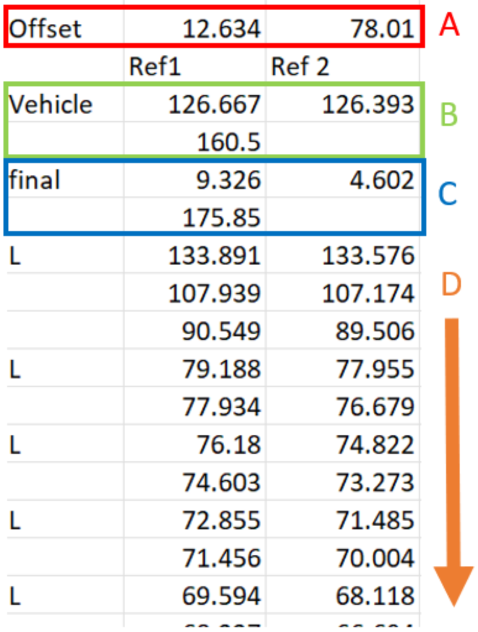
For the purposes of this work, a 2D sketch is more than enough, therefore, we will use the methodology presented in Chapter: 3.2.1. This can, very easily, model the outlines of a road and save it as in a file as vertices. The database presented in figure 15 its this works implementation. It consists in 4 groups, A, B, C and D.

Group A is where the fixed point of references is defined by the distance between two fixed points. This will form the y axis, and the angle offset of itself. The x axis is always perpendicular e positive to the right.

Group B is where the initial position of the vehicle is defined, as well as its heading. Followed by group C which defines the final positions and heading.

Group D is the vertexes and faces that make up the road. Each line segment is separated by an L, therefore a line must be made of at least 2 point, with more points be added to form a curve, the resolution of the curve is proportional to the number of points measured.

Each point is measured by the distance to both fixed references, then a the coordinates in *xy* space can be calculated by triangulation, its important to notice that triangulating based on two points, gives two possible coordinates, therefore its necessary that the measured points need to be all to the left, or right, of the fixed points of reference. To mitigate, one can always add a third fixed point of reference, the same process applies.



Offset	12.634	78.01
	Ref1	Ref 2
Vehicle	126.667	126.393
	160.5	
final	9.326	4.602
	175.85	
L	133.891	133.576
	107.939	107.174
	90.549	89.506
L	79.188	77.955
	77.934	76.679
L	76.18	74.822
	74.603	73.273
L	72.855	71.485
	71.456	70.004
L	69.594	68.118

Figure 15: Example of Database for Accident Scenario

3.3 EDR CRASH DATA

The EDR data from a vehicle can be divided into 3 separate phases: the 5 seconds before the impact point, the impact point, and the post-crash movement. Each phase requires a unique analysis process since the data points are different between them, and some don't have any as is the case post-crash.

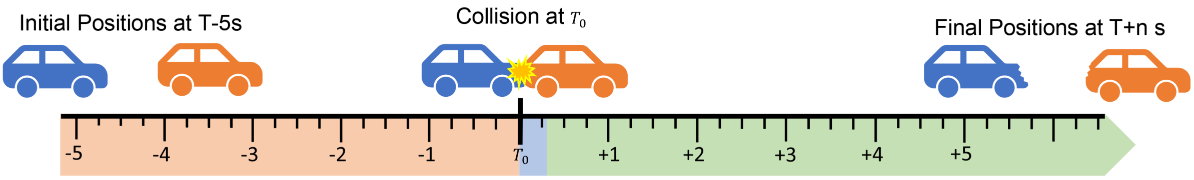


Figure 16: Collision Timeline

Figure 16 illustrates the timeline of an accident, we can distinguish easily the 3 phases separated by the colors. The variables recorded, as per the mandatory EU regulation indicated in tables 2, 3 and 4 related to vehicle dynamics, are as follows:

Pre-crash data from -5 seconds to T_0:

- Speed - Indicates the longitudinal speed at which the vehicle is traveling.
- Braking - Indicates if the brake pedal is pressed or not.
- ABS / Stability Control Activity - Indicates the status of the anti-lock brakes and Traction Control systems.
- TPS - Indicates the Position of the throttle pedal.
- RPM - Indicates the rotational speed of the engine.
- Steering Angle - Indicates the angle of the steering wheel (not the wheel angles).
- Lateral and Longitudinal Accelerations.
- Yaw rate - Indicates the Rotational Momentum along the normal.

Impact data from T_0 to 300 milliseconds is the Longitudinal and Lateral Accelerations recorded at a frequency of 100Hz, this data is integrated to determine the total Δv sustained by the vehicle.

For the post-crash, 300 milliseconds to T + n seconds (with n being time for the vehicle to achieve its resting position), there's not a lot of data to use. The EDR only provides roll rate and angle, movement, and momentum along the xx axis from -1 s to +5 s, mostly used for rollover studies.

3.3.1 *Pre-crash 5 seconds model*

As stated above, the objective of this model is to recreate the movement of a vehicle and, contrary to a most dynamic model which gets data points with the vehicle following a specified path, as demonstrated in the models specified in chapter 2.8. However, our circumstances require the reverse, and take the dynamic data as inputs and recreate the movement in a 2D, or even 3D environment.

In summary, in order to achieve this objective, the model requires a determined set of inputs taken directly from the EDR data followed by the processing of said data and running them through a series of movement equations and output of the location of the vehicle in a certain space frame.

3.3.1.1 *Inputs*

For the model inputs, some of the recorded pre-crash data points are going to be used and other can be ignored as they don't have a dynamic component. those are:

- Speed:

  This describes, on a time basis, the vehicle longitudinal speed (V_x), acquired by the wheel speed sensors belonging to the ABS control module, which is then shown to the driver in the dashboard. This is useful information that allows us to know the speed in the moments before impact. However this datapoint might not be always an absolute measurement, this happens during long instances of wheel slippage or lock up. Since the sensor measures the wheel speed directly through a hall effect sensors, if one or more wheels are spinning at different speeds, or stops altogether, it may record erratic data. This is a known data limitation that needs to be taken into account, although some vehicles already compensate for this wheel slip.

- Braking:

  This element describes the service's brake pedal activation, this means if either the pedal is pressed or isn't. This data is acquired by the brake switch on the brake system (the same switch that turns on the brake lights). The amount of force pressed on the pedal its what determines the braking force, and therefore its not related to the pedal travel. With this in mind, it's not possible to know the true intent of the driver and only the binary output. This data can be useful to determine the reaction time of the driver and give more context to the maneuvers performed by the driver moments before impact. Although not directly a dynamics variable, it can help set up the context.

- ABS / Stability Control Activity:

  Almost every new vehicle is fitted with the ABS and Stability Control Systems, this element logs its activity moments before the impact.

  The ABS logs information as being Faulted, in case an error code is detected, Active if the system is on, and Intervening, during a wheel lock situation.

  Stability Control has similar states to the ABS (Faulted, On, and Intervening) with the added option to record if the system is outright disabled (off).

  This datapoint does not constitute a direct correlation to the vehicle movement but it can add context to the driver's input such as, being able to know if the vehicle was induced into a spin or drift through the stability control or if the wheels lock up or the vehicle is during a slippage.

- TPS:

  The TPS (Throttle Position System), describes the throttle position in a value between 0 % and 100 % and thus the intention for the driver to accelerate.

  This variable is used to control the fuel/air mixture delivered into the engine and thus its power output through the Engine's ECU. This data is acquired by two potentiometers present inside the throttle pedal assembly, changing its resistance linearly with pedal travel.

Although not directly tied to the dynamics of the vehicle, it still can be used to give context to the driver's intentions.

- RPM:

  This element describes the engine's rotational speed in rotations per minute. This data comes from a hall effect sensor present in the crankshaft.

  This variable isn't that useful since there's no transmission gear, or clutch pedal positions, recorded so it can't be directly correlated with anything.

- Steering Angle:

  This element describes the driver's intention to turn, by measuring the angle of the steering wheel. This data comes from an electromechanical sensor attached to the steering column, these can be optical, magnetic, or inductive, and outputs the absolute steering angle, rate and torque applied. The angle obtained doesn't match the wheel angle or turning angle, so unless the steering rack gear ratio is known there can't be a direct correlation between the driver input and the rotation of the wheels, and even worst, this datapoint doesn't take into account situations of oversteer/understeer where the wheels lose tracking and the vehicle doesn't turn the intended way, losing the back end, or front end respectively.

- Lateral and Longitudinal Accelerations:

  This element describes the accelerations sustained by the vehicle moments before the impact along its x and y axis, longitudinal and lateral respectively. This data comes from the accelerometers present inside the ACM or Stability Control Module, which its used to control the Airbag deployment or the actuation of stability control countermeasures.

  This variable can be used to give context to the vehicle maneuvers since it gives the forces applied to the vehicle's center of gravity at that time.

- Yaw Rate:

  This datapoint evaluates the rotational momentum of the vehicle along the normal axis, z, logging the change in rotation, not the absolute value. the data comes from the stability control unit gyroscopic sensor, a device that can measure angular displacement in various axis by rotating a known mass in a gimbal with 3 DOFs (Degrees of Freedom) and measuring its displacement.

  Yaw rate is indispensable to recreate the movement of the vehicle, allowing an accurate recreation of the turning motion of the vehicle since it records events like under/oversteering and others that the rest of the sensors don't catch.

All of this data is recorded in the time before the accident, -5 seconds to T_0, at a rate of 2Hz, to the EDR's EEPROM inside the ACM. Some manufacturers might record more than 2 Hz, for example, some newer Fords record at 10 Hz, much more reliable data since there's less time between the data recording and much more likely to catch sudden changes. This is also important data that allows us to pinpoint the changes in a time frame.

One data limitations of this ”Time” is that T_0 does not always match the impact point, this happens because of the way the data is saved to the EDR. If the impact happens somewhere between the time frame of saved data and the next data, the EDR cuts directly into recording the crash pulse, without recording the next data. Some manufactures do present this type of data as a ”time since last recording” or by adding a new datapoint right when the algorithm is enabled, but its important to notice that its not standard between different brands.

Once the data is extracted using the CDR tool, it can be imported directly into our model with minimal changes since the program already exports the raw translated data. The most useful data that will be taken as inputs are, as explained above, data with dynamic relevancy such as:

- Time - t
- Speed - V_x
- Steering - δ
- Longitudinal Accelerations - a_x
- Lateral Accelerations. - a_y
- Yaw rate. - $\dot{\theta}$

The rest of the variables can be used as added context overlaid onto the model.

3.3.1.2 *Outputs*

With the inputs specified, it’s also important to determine what the outputs of the model are. Since this is a reverse of the normal dynamics model, it makes sense that the output is a trajectory. This trajectory can be in a 2D plane or 3D space, depending on the use case, which can change depending on the type of scene recreation process its used, as exemplified in Chapter 3.2. The process should stay the same but instead of the 3 DOF of the 2D model, the 3D model will have 6.

Starting with the 2D movement of the vehicle is best described by keeping track of the positional coordinates on the x and y axis of the vehicle’s geometric center and its rotation along the z axis, $\dot{\theta}$. This requires two sets of coordinate systems, one for the environment position and another for the vehicle internal calculations, ξ and η

3.3.1.3 *2D Model*

The model can be built, according to the inputs and outputs previously defined, to calculate its movement along a 2D plane, since there’s only data for 3 DOF. The relevant acquired data points are the Speed, Yaw rate and Steering input which can be interpolated with time, the end result will be an equation that describes the inputs of the driver through time, and can be referred to as $v_{(t)}, \dot{\theta}_{(t)}, \sigma_{(t)}$, with t being the time frame between -5 seconds to T_0.

Since this is only a 2D plane, the bicycle model [19], presented previously in chapter: 2.8.3, is enough to calculate the x and y movement of the vehicle along with its rotation. By reducing the vehicle to a single point, which coincides with the center of gravity, the movement can be described as using planar motion [20]. A certain point in space, which is located from the global coordinate system, (x,y), by an r_i vector. This point has its own local coordinate system,(ξ,η) that offsets from the global system by an angle of ϕ. The point will move according to a vector s_i applied to the local system. To translate this movement to the global system, the new point coordinates would be r, by adding the movement vector rotated by the offset angle to the old coordinates. This is shown in equation (17), with A as the rotation matrix.

$$r = r_i + A \cdot s_i \tag{17}$$

The previous equation can be written as equation (18)

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}_i + \begin{bmatrix} cos\phi & -sin\phi \\ sin\phi & cos\phi \end{bmatrix}_i \begin{bmatrix} \eta \\ \xi \end{bmatrix}_i \tag{18}$$

Therefore the bicycle model with both global and local coordinate systems can be represented with figure: 17 .

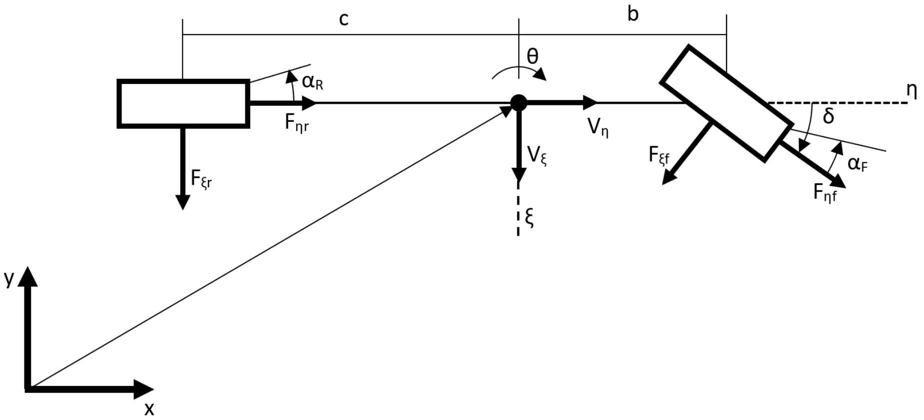


Figure 17: Bicycle Model in Global Coordinate System

To obtain the position and heading of the vehicle, its necessary to obtain it's longitudinal and lateral velocity, V_η and V_ξ respectively, and heading of the vehicle, θ. This can be calculated by analyzing each DOF:

- Longitudinal Dynamics

  Corresponds to V_η and its equal to speed of the vehicle given by $v_{(t)}$

- Lateral Dynamics

  Are the forces the actuate on the side of the vehicle, more specifically the tire forces on the y direction. The sum of the tire forces are equal to the mass of the vehicle times its lateral acceleration, equation (19), and this acceleration can be given by the sum between lateral and longitudinal velocities, equation (20).

$$ma_y = F_{\xi_R} + F_{\xi_F} \cdot cos(\delta) \tag{19}$$

$$a_y = V_\xi + \dot{\theta} V_\eta \tag{20}$$

Therefore by adding both equation (19) and (20), the following equation (21) for lateral acceleration is achieved.

$$\dot{V}_\xi = \frac{F_{\xi_R}}{m} + \frac{F_{\xi_F}}{m} - \dot{\theta} V_\eta \tag{21}$$

- Yaw Dynamics

  These are the forces that induce rotation on the vehicle, it achieved by the sum of the torque produced by the lateral tire forces, as shown in equation (22)

$$I\ddot{\theta} = b \cdot F_{\xi_F} - c \cdot F_{\xi_R} \Leftrightarrow \ddot{\theta} = \frac{b \cdot F_{\xi_F}}{I} - \frac{c \cdot F_{\xi_R}}{I} \tag{22}$$

- Tire Forces

  These are the forces that actuate on the tires. There are several models that have been created, but for a linear behaviour, the linear Model is the most appropriate. For small slip angles the behaviour of the tire can be assumed as linear, and therefore the lateral tire forces is a product between the slip angle, α_i and the tire slip stiffness C_α, the latter being the elastic property for each tire and represents the stiffness of the tire itself. This value is normally between the range of 30000 and 50000 N/rad [21]. This can be represented with equation (23) for front and rear lateral tire forces.

$$F_{\xi_F} = C_\alpha \cdot \alpha_F \qquad F_{\xi_R} = C_\alpha \cdot \alpha_R \tag{23}$$

  The slip angle for both tires can be thought as the ratio between the lateral velocities and longitudinal velocities, with the exception of the steering offset necessary for the front axle. Hence the slip angle for rear and front tires are described respectively in equations (24) and (25).

$$\alpha_R = \frac{V_{lat} - c \cdot \dot{\theta}}{V_{lon}} \tag{24}$$

$$\alpha_F = \delta - \frac{V_{lat} - b \cdot \dot{\theta}}{V_{lon}} \tag{25}$$

Now all the components to solve equations (21) and (22) and obtain the lateral acceleration and yaw acceleration, which can be integrated and calculate the lateral velocity and yaw rate for a given time delta, as exemplified in equation (28).

$$V_{lat} = V_{lat_i} + \dot{V}_{lat} \cdot t \quad \dot{\theta} = \dot{\theta}_i + \ddot{\theta} \cdot t \tag{26}$$

To finalize, this movement must be translated from the local coordinate system of the vehicle to the global coordinate system of the road, therefore applying this equations to the planar motion, equation (27) represents the vehicle movement in a 2D plane.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}_i + \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}_i \begin{bmatrix} V_{lon} \\ V_{lat} \end{bmatrix}_i \tag{27}$$

With θ as the heading of the vehicle and therefore:

$$\theta = \theta_i + \dot{\theta} \cdot t \tag{28}$$

3.3.2 *Crash Impulse Model*

The impact time frame is only a fraction of a second, after the detection of the crash, the EDR will begin to record two variables at an increased rate of 100 Hz. This 300 ms time frame only exists to log information about the vehicle's extreme accelerations sustained during a collision. The high recording rate of these longitudinal and lateral accelerations is required since those will be used to calculate the Δv for both directions by integrating the acceleration data which is useful information to analyze the impact plane, direction and severity, as explained in chapter 2.2.1.

Using the acceleration data exported from the EDR its possible to determine the direction and magnitude of the change of velocity and therefore the direction from where the force was applied. By using an integral to get a change in velocity from acceleration data and knowing the time between each reading, in this case 10 ms, and if velocity is acceleration multiplied by time, $v = at$, the total sum of EDR readings will represent the total Δv in that particular direction sustained by the vehicle.

This can be viewed as calculating the area formed by the polygon between two consecutive accelerations and its x axis and then adding each consecutive polygon. According to the vehicle's EDR axis direction, above the x axis means the vehicle gained speed from the impact, and below means that it lost speed during the impact, and vice versa. This can be achieved by calculating the sum of the integral from each 10 ms range using equation (29), data resolution can be improved by applying linear interpolation to the data decreasing the Δv sum error.

$$\sum_{i}^{n=30} \int_{t_i}^{t_{i+1}} adt \tag{29}$$

But this Δv is not correct [22][23] due to the location of the ACM, where the accelerometers are located, isn't on the vehicle's center of gravity and more in the front under the dashboard. In a collision, if a vehicle gets hit on the off-axis, it impairs a rotation momentum to the vehicle and the Δv reading is influenced by the location of impact, if it happens closer to the center of gravity, the Δv is under-reported and if it happens closer to the ACM the Δv is over-reported.

Since this discrepancy is caused by rotation, a simple correction factor can be applied by subtracting, if Δv is over-reported, or adding, if it's under-reported, the rotational momentum to the EDR's Δv. With this its possible to obtain the Δv at the center of gravity. However since rotational momentum is not readily available, therefore the correction factor is based on the distance of the impact point to the center of gravity. Equation (30), which demonstrates the correction factor, can be combined with equation (31) for the conservation of angular momentum.

$$\Delta v^{CG} = \Delta v^{EDR} \pm r \Delta w_{yaw} \tag{30}$$

$$I \Delta w = m h \Delta v \tag{31}$$

For this general purpose, and assuming the EDR is located in the center of the vehicle longitudinal axis which is true in most cases, and therefore only requiring offset to the lateral Δv the equation (32) can be used to do the aforementioned offset. Where k is the radius of gyration, equation (33), the Δv_{EDR} is the Δv from the EDR, r_y^{cent} is the lateral displacement from the damage centroid to the center of gravity, r_x^{cent} is the longitudinal displacement from the damage centroid to the center of gravity, r_x^{EDR} is the longitudinal displacement of the EDR and r_y^{EDR} is the lateral displacement of the EDR, but due to this approach, it is usually zero. These measurements can be better illustrated in figure 18.

$$\Delta v_y^{CG} = \frac{\Delta v_y^{EDR}.k^2 + \Delta v_x^{EDR}.r_y^{cent}.r_x^{EDR}}{k^2 + r_x^{cent}.r_x^{EDR}} \tag{32}$$

$$k = \frac{Ig}{Weight} \tag{33}$$

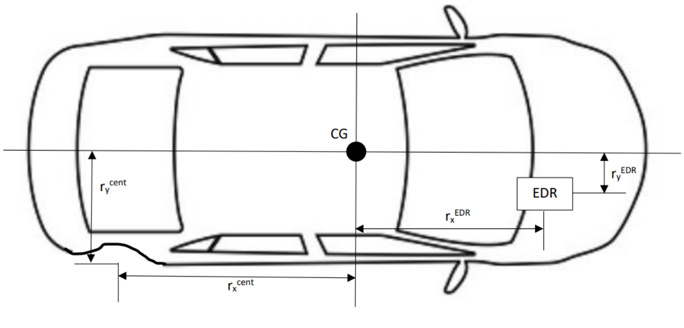


Figure 18: Example of Offset Measurement

After the offset is applied to the cumulative Δv for the lateral and longitudinal direction of the vehicle, simple trigonometry can be applied to find the overall resulting Δv and its direction, known as PDOF, explained in chapter 2.2.1. With the PDOF the plane of impact can be discovered, since the plane is always perpendicular to the PDOF, and all of the collision parameters are known, provided that the vehicle's point of collision is known, this is the point of impact that can be determined from the vehicle's damage.

With the information from the impact pulse model, change vector, and the information from the pre-collision 5 seconds model explained in chapter 3.3.1, change vector, and using the triangle approach explained in chapter 2.2.1 the departure vector by applying the crash-pulse to the impact plane, finding the exit angle, while its magnitude is the Δv. Therefore all the conditions are gathered and the parameters from the impact can be clearly described as shown in figure 19

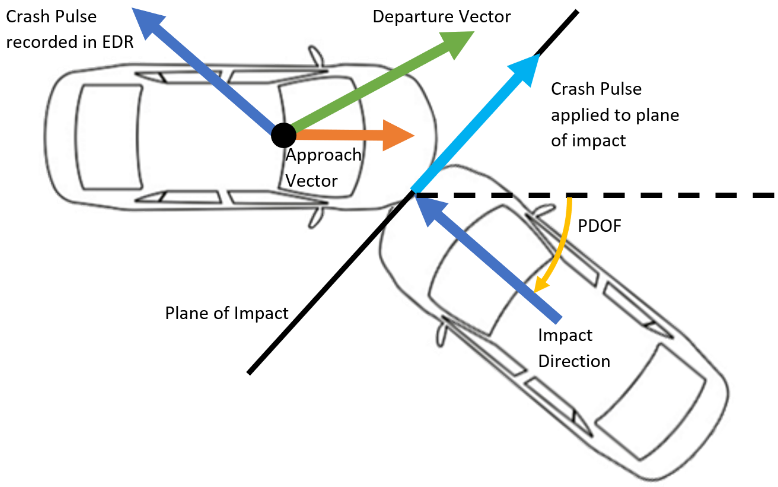


Figure 19: Use of Crash-Pulse Data in Crash Reconstruction

In sum, the Crash Impulse Model can be described in the flowchart in figure 20 alongside all the inputs and outputs of this process.

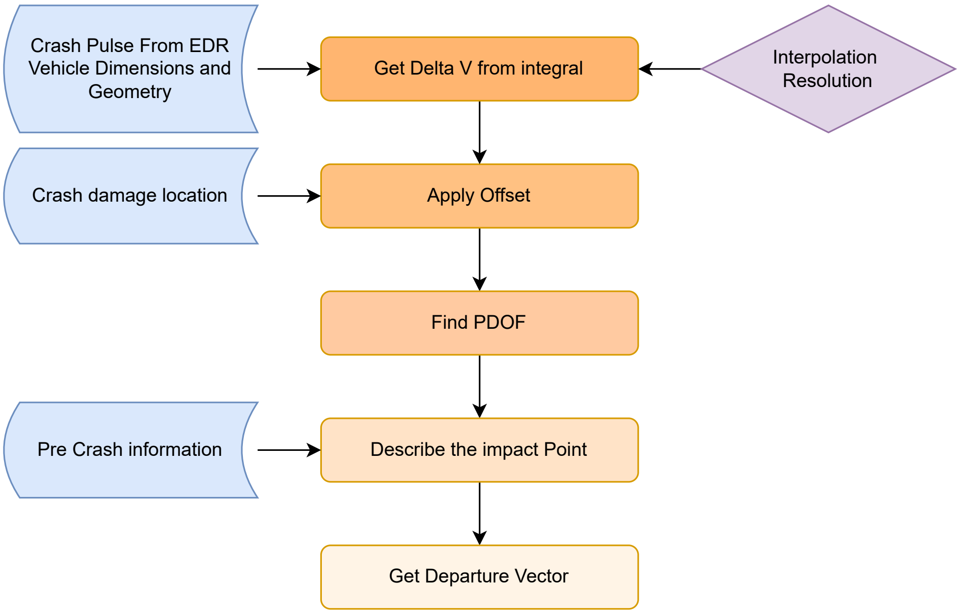


Figure 20: Process for the Impact Pulse Model

3.3.3 *Post-crash model*

After a collision the vehicles follow a decelerated motion until its final resting positions, this happens because the driver can become stunned after the collision or is mostly worried about stopping the car, which means stepping on the bakes. After analysing the crash pulse data, the departure vector, which indicates the direction and speed of the vehicle after the accident, can be used as a starting point and, because the end point is known by analyzing the crash location. Therefore since both initial and final positions and headings are known, a simple path finding algorithm can be used to try and steer the vehicle to the final position and heading, while decelerating uniformly during the whole maneuver. To achieve this, it needs to be between certain parameters that are possible to a vehicle to achieve, this means that turning the wheels from one side to another and accelerations must not exceed whats physically possible.

The algorithm works by incrementing a specified time-step until the kinetic energy of the vehicle is zero, this means the vehicle has stopped and its velocity is zero. During this time the vehicle will try and follow a trajectory between the collision point and a specified end point using a PID (Proportional-Integral-Derivative) algorithm for yaw control.

A PID controller works by iterating by a specific time-step and check the error between the real value and the desired value and applies a correction according to a proportional, integral and derivative factors of said error.

3.4 MODEL TEST

To put this model to the test we need some real-world data taken from a real accident, unfortunately, there wasn't any situation where we could have obtained this data, so a different methodology was devised. This new approach comprises of simulating a mock-up crash in PC-CRASH™ *software.*

The functionalities of this *software* are better explained in chapter 2.2.4 but it is necessary to reference that this simulation used the *stiffness based impact model* as opposed to the other options, since this model assumes the vehicle as a rigid body and the contacts between the surrounding bodies are calculated using ellipsoids, also a coefficient of restitution is defined to allow some elasticity during the impact, improving the overall realism of the simulation including the crash pulse which will be studied next. These parameters are unique for every vehicle present in the database.

3.4.1 *Crash Scenario*

The next figure, figure 21, represents the simulated crash scenario, consisting of a national road, in Portuguese designation. The road has two lanes, one for each direction, after a subtle left turn corner the road is intersected by a minor road that doesn't have traffic priority. The red vehicle (V1) is the vehicle equipped with EDR and its initial position is represented in yellow where the EDR's recording starts, at T= -5 s. The blue vehicle's (V2) initial position, marked as orange, is near the intersection and starts stationary since the car had a stop sign in front of it.

Both vehicles will try to follow their designated trajectories, red line represents the intended trajectory for V1, and, depending on grip and speed, it will follow the line by changing the steering wheel angle through PID. The same behavior happens to V2.

The scenario represents a situation in which V1 is above the road's speed limit of 50 km/h and in its left lane there is a traffic jam which impairs vision to the intersection ahead, V2 decides to join the main road, specifically the lane V1 is using. At T = -1 s V1 is able to see V2 coming to its lane and it immediately presses the brake, without locking out the brakes since if it was a real scenario, locking the brakes would result in erroneous speed data.

At T_0 the impact happens, marked in blue, with V1's driver not being able to stop the vehicle it hits V2 in its rear. After the impact, both vehicles stop in their final positions marked in red on the sketch.

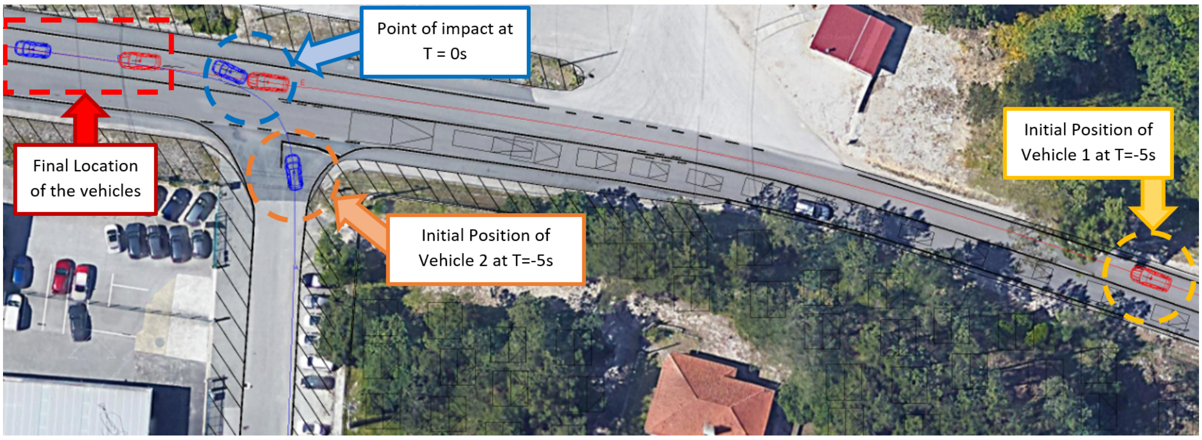


Figure 21: Crash Scenario

The vehicle maneuvers intrinsic to this simulation are as follows. From T = 5 s to T = 1 s, V1 will maintain its initial velocity of 80 km/h, until where the brake is pressed, inducing a deceleration of 5.77 ms^{-2} to the vehicle. The impact happens at T = 0 s, referred as T_0. At this instance the crash pulse starts and lasts 300 ms. Here the driver stops controlling the vehicle, this happens to emulate the initial stun of the crash for the driver and so that the acceleration data for the crash pulse isn't stained with other accelerations so a more clear reading can be made.

As for V2, it accelerates at a rate of 2.1 ms^{-2} from stationary since followed by the 300 ms buffer, similar to V1. After the impact both vehicles decelerate at a rate of 6.06 ms^{-2} and 4.36 ms^{-2} for V1 and V2 respectively, otherwise, the vehicles wouldn't have any stopping force and they would just keep moving.

3.4.2 *Exporting EDR data*

With the computational simulation completed, the dynamic data can be studied. PC-CRASH™ allows to plot data from each vehicles. This data is comprised of velocity, yaw rate, roll rate, pitch rate, steering angle, brake actuation, tire forces, accelerations, EES, and Energy for each vehicle, and therefore the relevant data stated prior can be exported in a way that resembles the real EDR data. To achieve that, the data from velocity, yaw rate, lateral and longitudinal accelerations, brakes, and steering angle will be exported at a rate of 2Hz. The final exported data can be viewed in table 6.

Analyzing the data, one can deduce that the vehicle was cruising at 80km/h through a left corner, due to the steering angle being positive and the lateral acceleration also positive. This behaviour means that positive steering it turning left and positive lateral acceleration it's from left to right relative to the vehicles referential axis. This behavior continues for 4 seconds, where upon viewing the imminent impact, the driver steps on the brake decreasing its speed from 80 km/h to 69 km/h after 0.5 seconds and to 58 km/h after 1 second, which corresponds to the recorded longitudinal acceleration of 0.59 G recorded in the EDR. This data shows exactly what the simulation's maneuver imposes on the vehicle.

Table 6: Simulated EDR Data

Time [s]	**Speed** [km/h]	**Steering** [deg]	**Yaw** [rad/s]	**Lon. Accel.** [G]	**Lat. Accel.** [G]	**Brake**
0,0	80	19	0,02	0	0,11	0
0,5	80	3,5	0,07	0	0,11	0
1,0	80	5,5	0,04	0	0,11	0
1,5	80	11	0,07	0	0,13	0
2,0	80	8	0,08	0	0,16	0
2,5	80	5	0,06	0	0,15	0
3,0	80	1,5	0,03	0	0,11	0
3,5	80	1,5	0,02	0	0,07	0
4,0	80	0,5	0,01	-0,59	0,04	1
4,5	69	-3	0,01	-0,59	0,02	1
5,0	58	-2	-0,01	-0,59	0	1

As for the impact impulse the same procedure was followed, but instead of the previous variables, only the lateral and longitudinal accelerations are required at a rate of 100 Hz. Upon exporting the data, it can be viewed in the following figure 22. Analyzing the plot data, with the blue line being lateral acceleration and the red line being longitudinal acceleration, the impact can be clearly discerned. With most of the impact having a longitudinal acceleration component with a maximum deceleration of 5 G and abruptly ending when both vehicles no longer have contact. The residual accelerations after the peek can be attributed to the vehicle trying to get on its trajectory after the impact.

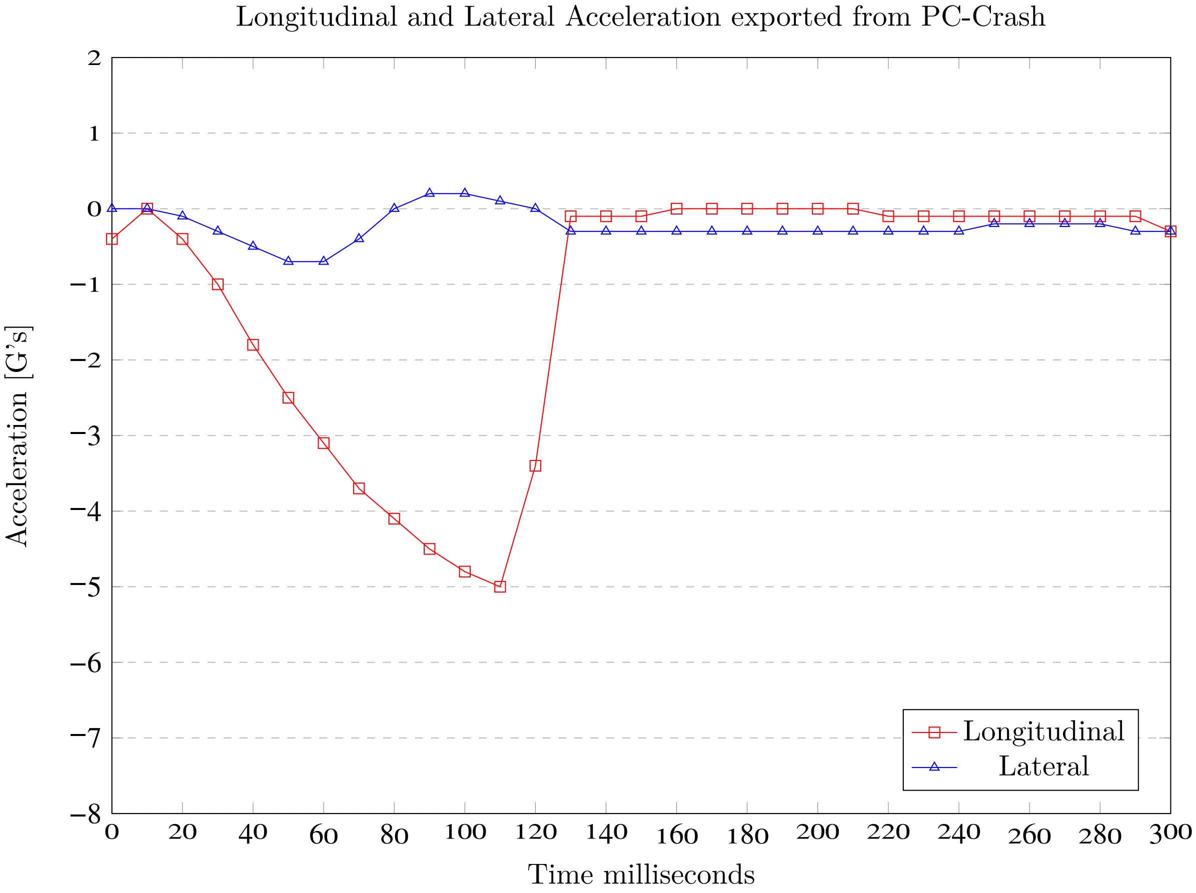


Figure 22: Crash Impulse Plot

And final data required is the vehicle and environment relative positions. With all the information gathered for the pre-impact, impact pulse, and post-impact situations, each respective model can be directly applied to each situation.

3.4.3 *Creating the Testing Environment*

To proceed with testing the methodology a testing environment needed to be devised. The program requires a graphical interface to display 2D graphics to a defined scale, the scale is important because of the context given by the road to the position of the vehicle. For this reason, a 3D Software Rendering Program was created using the computer language Python.

Python was chosen since it's a simple language that allows for easy prototyping of small programs and with a large available selection of different libraries for different purposes, in this case, some libraries for graphic control and data management were required. There are also a lot of python versions that can, or not, be compatible with the required python libraries. For this work, the python version used will be 3.10.4 with the following libraries showcased in table 7.

Table 7: Python Libraries

Name	**Description**	**Version**
MatPlotlib	Library for data visualizations	3.6.1
NumPy	Package for scientific computing with Python	1.23.4
Openpyxl	Read and write to Excels	3.0.10
Pandas	Data analysis and manipulation Tool	1.5.1
Pygame	Computer Graphics Library for games	2.1.2
Scipy	Algorithms for Scientific computing	1.9.3

The program is divided into two separate segments, the viewer and the data analysis. The viewer consists in a simple 3D software render, which can display vertices and faces in full 3D space to create wire-frame graphics, this means graphics are built using vertices and lines with no texture. This version is based on an open-source software 3D viewer created by GitHub user ”StanislavPetrovV” [24]. This publicly available solution was a great base to create this program since it comes to the basic process of using a ”view frustum” to transform the points from 3D space to the camera space and how to create wire-frame graphics. The open-source code was modified to cater to the project’s needs.

Each component of the accident scene is classified as a different object, such as the road and both vehicles can move independently from each other. The camera can be used to move around in the xy axis and zoom in the z axis, the camera can also rotate but it’s always in topographic view since only 2D graphics will be used.

The program displays an image that contains the road segment which was scanned, the two vehicles and their respective impact planes. The two vehicles are the two different approaches taken to calculate the vehicle’s movement, which calculating from the yaw data, and the other consists of calculating the vehicle’s yaw rate from the steering angle.

The data analysis segment is comprised of different algorithms to read the data from their respective databases which were formatted in Excel files as described in the previous chapters. Here the geometry for the road and vehicles is created as well as their center of gravity’s movement.

Figure 23 below presents a simplified overview of the algorithm responsible to run the program. It starts by configuring the application window, followed by the creation of objects in space, in this case, the camera and its projection matrices, axes, and both vehicles for both types of simulation control and creating the visual representation of the impact points for both vehicles. Then the algorithm starts a loop that continuously draws the new frame and moves the camera, using the specific camera controls. The program stops once the exit condition is met.

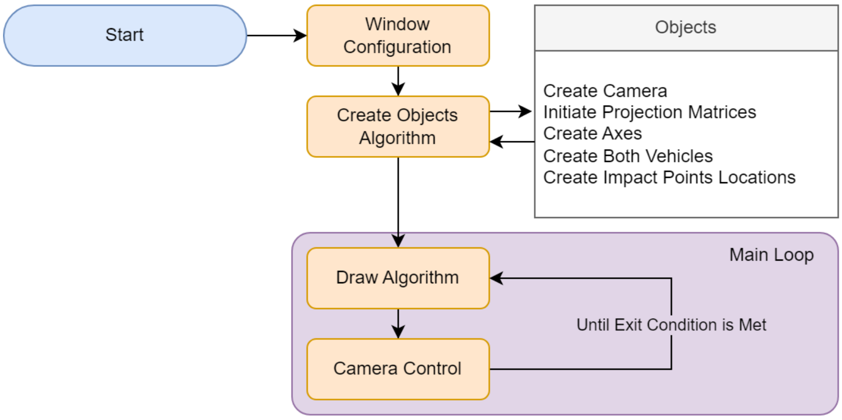


Figure 23: Main loop

The object class, shown in figure 24, represents the algorithm that updates the screen with the vehicle's movement. The vertices and faces that consist of the 3D object, created in the vehicle class, are introduced here and controlled.

The simulation control refers to the part of the program that processes users' input related to the simulation time, either stepping each data point or letting it play, this indicates which position the vehicle must be, based on the position array. The vehicle's mesh is then translated and rotated to reflect that movement, followed by updating the UI elements, and then every mesh goes through the screen projection subroutine where the points and vertices are transformed from the 3D space to a 2D image based on the perspective of the camera. This 2D image is then displayed on the screen

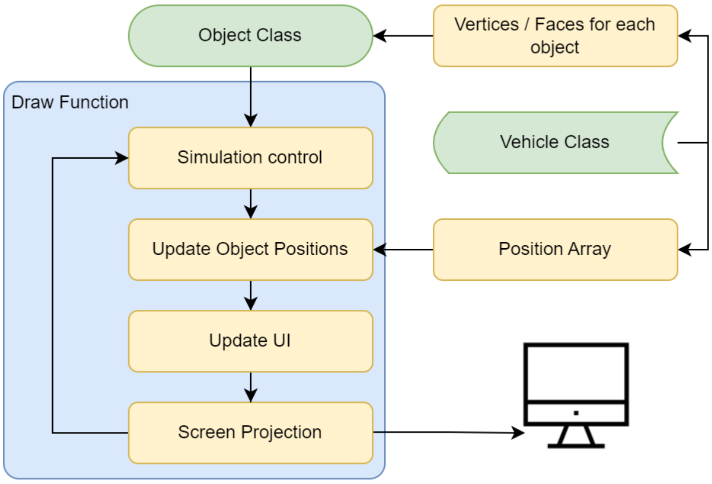


Figure 24: Object Class

In figure 25 it's possible to visualize the most complex module for this program. The Vehicle class comprises everything presented in chapter 3, and it takes part in creating the vehicle movement from the pre-crash, post-crash and crash-pulse data, creating the sketch from the road measurements and creating the visual representation of the vehicle.

The vehicle class starts by accessing the vehicle database, which describes the various geometries that comprise the vehicle that might be necessary, as well as the pretended simulation parameters. This is then passed thought the various functions, which can be described as follows.

- Vehicle Dynamics - Yaw Control

  This function is responsible for the calculation of vehicle position based on the EDR Data. The function accesses the data file with the EDR data and interpolates it from the standard 0.5s between each data point to the specified time step.

  Provided that the sample data file has the Yaw component, then the movement of the center of mass and heading is easily worked out by applying equation (27). After the pre-crash data has been processed, the program accesses the data from the crash-pulse, interprets the data, and creates the impact point configuration and the vehicle's post-crash speed and heading.

  To finalize, the post-crash movement is then calculated by trying to achieve the final position during a uniform decelerated motion.

  As for the program outputs, the vehicle position array consists of the position and heading of a vehicle, impact data is the impact plane with exit speed and direction. Vehicle data is information about the speed, acceleration, and steering thought time, and vehicle time is the total times from simulation start until the vehicle's speed reaches 0.

- Vehicle Dynamics - Steering Control

  This function works in the same way as the previous one, except for the implementation of the bicycle model to calculate yaw rate from steering input. This is for situations where yaw data isn't available or, if both are available, to compare the yaw created from the steering movement to the yaw recorded by the vehicle and find possible inconsistencies. Data output is equal.

- Sketch Generator

  This is the function that creates the mesh that makes up the road. It accesses the data file, as presented in chapter: 3.2.3, and retrieves information about the fixed reference points, the vehicle's starting and end position, and the vertices that make up the mesh. The points are transformed from the fixed point of reference to a xy axes by means of triangulation. The function builds the mesh by connecting each vertex to a face, or line. The separator ”L” lets the program know when a new line begins and another ends, grouping vertices to form curves and areas.

  The function outputs the mesh that forms the road, the start and end positions for the vehicles, and the origin point for the global axes.

- Vehicle Geometry Generator

  This function creates the visual representation of the vehicle by taking data from the vehicle database, consisting of information such as wheelbase, length, width, wheel positions, and center of gravity position to create a rough box representation of the outline of the vehicle. It outputs this geometry to be used by the rest of the program.

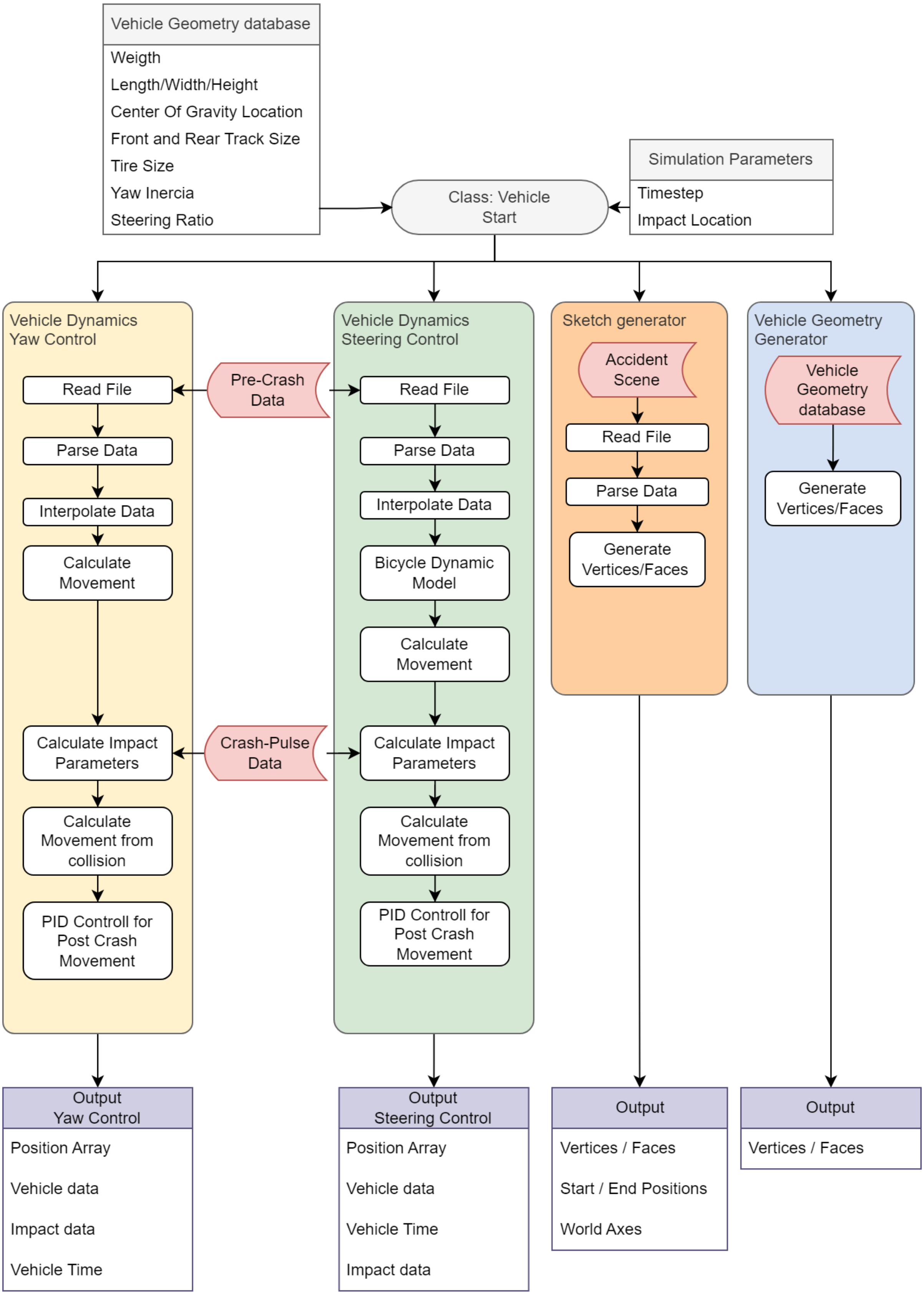
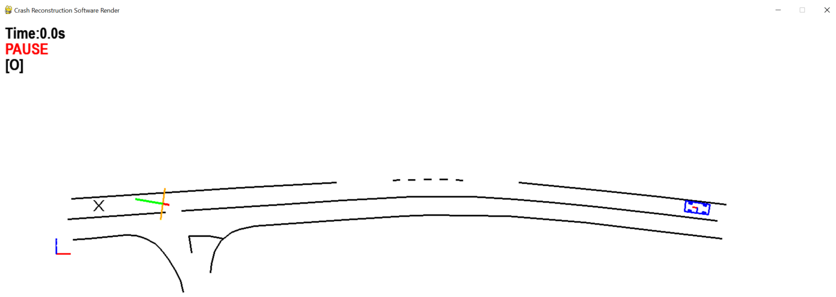


Figure 25: Vehicle Class

The finished software display output can be viewed in figure 26, and as intended, it makes an easy visual representation of the vehicle's movement in space, with roads delimitation', and the vehicle's intended final position marked as a cross, the impact planes, the vehicles in blue and red (red is currently behind blue since they're both at the starting position). The UI (User Interface) also displays the simulation's current time and its state.



```
Time:0.0s
PAUSE
[O]
```

Figure 26: Crash Reconstruction Software Program Window

Red vehicle represents the vehicle movement controlled by the steering, and the blue one corresponds to yaw control. The program also outputs a file with the data used for the vehicle's motion and impact data.

3.5 RESULTS AND ANALYSIS

With the program completed, this chapter will focus on the data calculated in the vehicle class, figure 25, data output since most of the dynamic data is treated in this function. The analysis will consist of comparisons between the movement through yaw control, steering control, and the PC-CRASH™ exported data.

Yaw control is relatively straightforward but, by contrast, steering control has the tire slip stiffness variable, which depending on the condition of the tires and it can influence the results. Another comparison will be made with the addition of Lateral Velocity for the bicycle model. The models included are as follows (The same nomenclature is used for subsequent plots).

- PC-Crash - Data from the PC-CRASH™program.
- Steering Control Cα 30000 - Bicycle Model with a tire slip stiffness of 30000 N/rad.
- Steering Control Cα 40000 - Bicycle Model with a tire slip stiffness of 40000 N/rad.
- Steering Control Cα 50000 - Bicycle Model with a tire slip stiffness of 50000 N/rad.
- Steering Control Cα 40000 With Vlat - Bicycle Model with a tire slip stiffness of 40000 N/rad and with lateral velocity accounted in the model

- Yaw Control - Position of the vehicle based on yaw rate.

The plot displayed in figure 27 represents the various model's vehicle trajectory, that is, the sequence of coordinates the vehicle follows in a *xy* plane. Unfortunately, PC-CRASH™ does not export the *xy* position data, therefore some measurements were made each half a second which is represented by the blue dots in the plot, as a means of comparison since interpolating the data, might not give the full picture. Also important to notice, the vehicle's movement is from the right side to the left side.

By analyzing the plot it's possible to visualize what effects the several models have on the vehicle's movement. Right from the start, the addition of lateral velocity only skews the movement even further, therefore it won't be accounted for in the rest of the models. The Steering control models, with a tire slip stiffness between 30000 to 40000 N/rad provide the best results, with the real movement of the vehicle situated in between them. A tire slip stiffness of 50000 N/rad provides almost the same results as the Yaw control, which also presents the best approximation for the end position of the vehicle. Its important to notice that the "Y position" axis does not have the same scale as the "X Position" axis, this is to better illustrate the differences between methods.

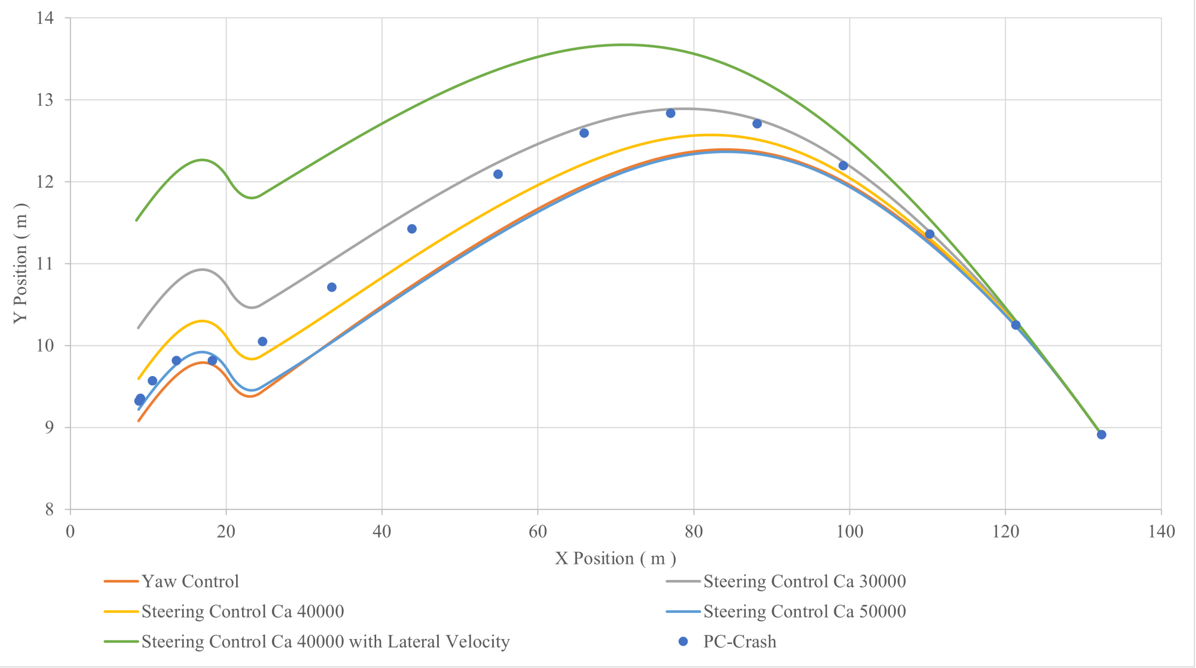


Figure 27: Vehicle Positions in xy Coordinates

The next plot, in figure 28 displays the yaw of the vehicle during the pre-crash 5 seconds. The major difference between the two types of models is clear, while the bicycle model with steering presents smooth curves, the yaw control displays linear segments that change their slope every half a second. This happens because the program interpolates data linearly. this also presents a major data limitation of recording at a rate of 2 Hz, the peak that occurs in the original data between 0 s and 0.5 s isn't recorded by the EDR and only records the data point in the falling edge of the peak.

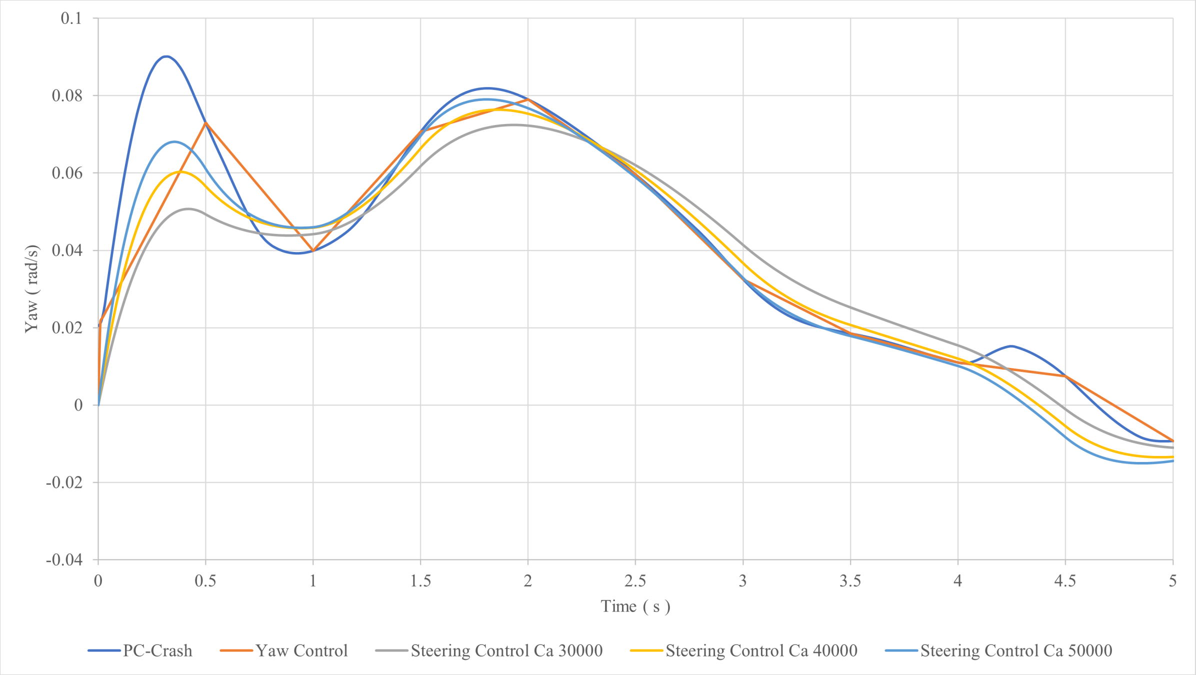


Figure 28: Yaw Pre-Crash Data

As for the data itself, none of the models recreate the initial peak, but after 1 second it settles and generates data that is similar to the original curve, with the bicycle model and a Cα at 50000 N/rad presenting the best data of the batch. Reducing Cα seems to flatten the curve, and doesn't represent fast changes in yaw. The point where the driver starts to brake is also present in the data, with the sudden rotation around 4 seconds.

The vehicle heading, or ψ angle describes the relative angle of the world reference point where the front of the vehicle is pointing, therefore, the plot in figure 29 displays the variation of heading for each model.

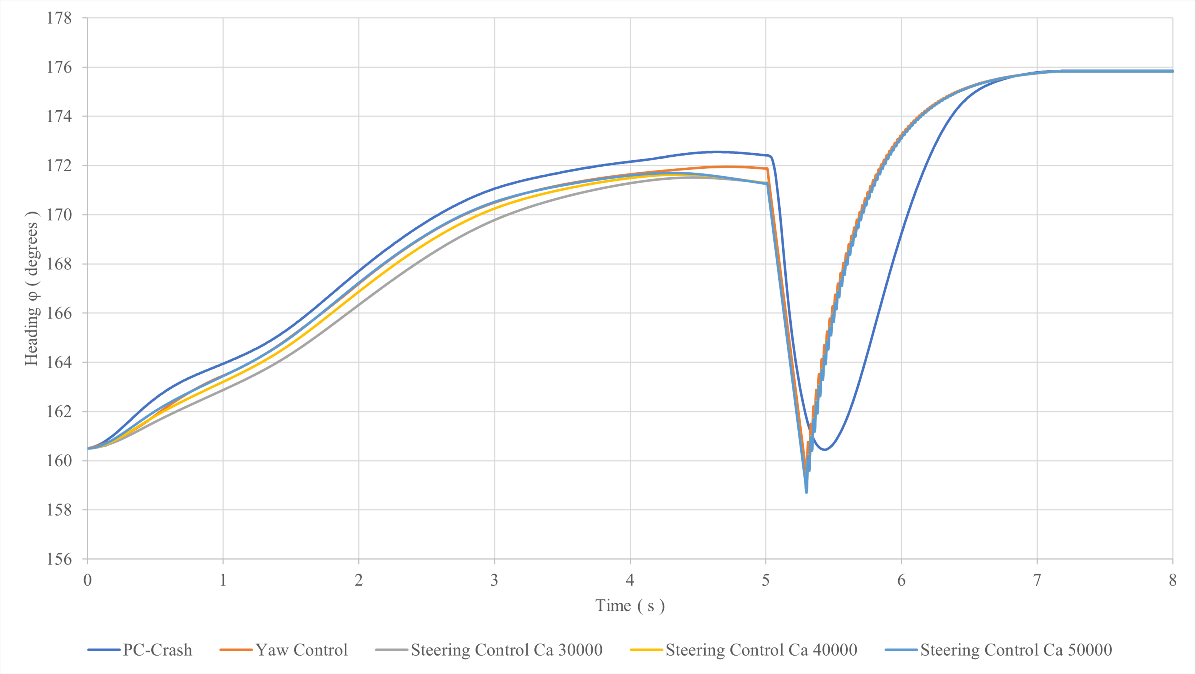


Figure 29: Vehicle Heading

Analysing the plot, we can verify that all follow the same variation as the PC-CRASH™ data, albeit with an offset during the pre-crash data. When approaching the impact point, the Yaw control follows the same line as PC-CRASH™, however, all of the models with steering control converge at the same angle.

The exit angle calculated from the crash pulse is also apparent, rotating the vehicle by approximately 11 degrees, the change on the PC-CRASH™ is more smooth compared to the linear approach used while rotating the vehicle. For the post-crash, the PID controller starts immediately turning the vehicle to the intended heading, hence the abrupt change in direction and not a smooth transition. All the models reached the intended vehicle heading, although that necessarily doesn't mean that they reached the final position.

Figure 30 is the final plot and it displays the change in speed of the vehicle through time of every model. As it can be observed, every model outputs a speed that closely matches the PC-crash™ output. One notable difference is that the after the impact, around the 5 second mark, it's possible to discern the Δv of the vehicle, as it loses speed upon impact. The slope of the variation is much higher on the PC-CRASH™ than on the rest of the models. This happens because the models are applying the Δv to the full 300 ms of recording, resulting in a much subtle drop in speed, while on the PC-CRASH™ side it loses speed much faster initially, and then plateaus for the rest of the 300 ms recording range, before starting to brake.

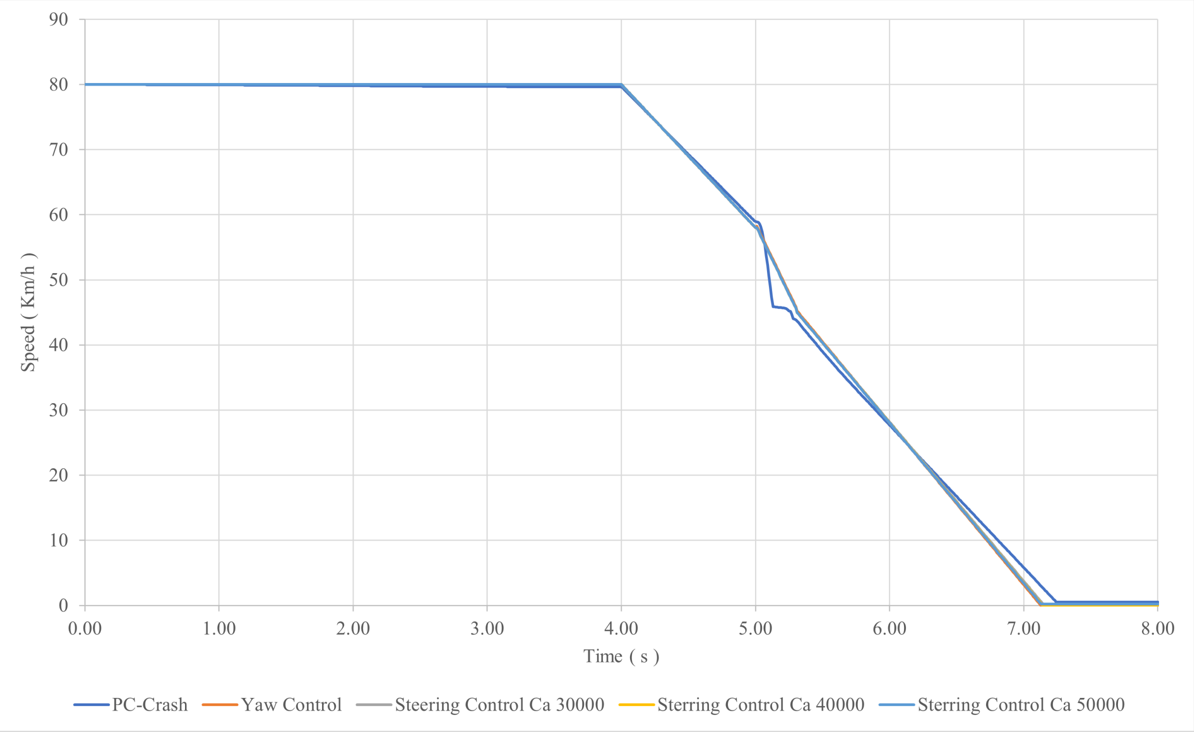


Figure 30: Vehicle Speed

These are all the data that were deemed relevant to describe the vehicle's movement through time and place. The various models output different data, but it's important to use them interchangeably, this means that possible differences between the models can show unpredictable behavior on the vehicle, such as understeer and oversteer.

4

CONCLUSION AND FUTURE WORK

4.1 FINAL CONCLUSIONS

This master Thesis in Automotive Engineering has allowed me to better understand the principles and dynamics of vehicle collisions and the practices used today for its analysis and recreation, and what's yet to come. The use of the CDR tool to extract the recorded data from the vehicle is an advantage not yet utilized to its maximum potential, and this study aims to introduce this data to collision reconstruction. I can confidently state that all the objectives stated during the introduction were accomplished, which means that the following conclusions were made.

The advantages of having detailed scenery improves greatly the quality of the simulation since the dynamic data without context isn't sufficient for its representations and therefore scanning the accident location proves to be a very reliable source of information. The analysis of the crash site using more modern tools means a better reconstruction can be achieved.

The data from EDR can be successfully reconstructed using mathematical models in all three phases of the crash.

The movement of the vehicle was successfully reconstructed in the pre-crash phase of the accident. The use of mathematical models allowed for reconstruction of the vehicle's trajectory through space with a certain margin of error. The use of Yaw control and steering control to calculate the motion of the vehicle provides two points of view of the data, one from the vehicle's prospective and another from driver's input. This difference might explain what lead to an accident and with the addition of other data points, such as traction control and ABS, and bring more context to the accident analysis.

For the impact pulse the data from the accelerometers proved to be useful in the recreation of the impact point, direction, and exit speed of the vehicle. With the crash pulse fully defined the pre and post-crash movement can be merged successfully.

The post-crash movement was greatly simplified and methodologies already used were implemented in this program. Because of lack of data for the post-crash recorded by the EDR, a the motion of the vehicle had to be assumed as a linear deceleration. However, despite this, the results were deemed reasonable, and the final positions of the vehicle can pre approximated using this method.

Finally merging all of the collected data and merging them on computer software, provided to be a helpful addition to the accident reconstruction workflow. By recreating the accident in graphical form, the analyst can better understand the context of the dynamic data with the accident scene, therefore providing with more evidences and explanations to the events.

Finalising this study has greatly improved my understanding of general vehicle dynamics and how it reacts during a collision, all the relevant questions were exposed and properly answered, providing me with enough information that helped me reach the all objectives proposed.

4.2 FUTURE WORK

When looking at the results of this study, one can conclude that the following improvements can be made to both the methodology and the computer software:

1. Use real-life events since this data was manufactured to create a mathematical model, although the data is similar, it doesn't reflect the real-life counterpart.
2. This model will assume that every data point is readily available with that standard structure, and in reality that doesn't happen that often, and it's very OEM dependent, with each one doing its own thing. So a catalog of several OEM translation protocols needs to be created.
3. A more seamless reconstruction can be made by utilizing a better physics engine and more realistic ways to display the road and vehicle geometry. The jump to 3D space also allows for a more pleasing video reconstruction.
4. The vehicle used in this case study was "hard coded" into the computer software, for the future a database can be created in order to have large amounts of vehicle specifications and geometry readily available.
5. The post-crash movement is using a simplified model in order to calculate it, therefore a more reliable and robust trajectory calculator must be implemented.

BIBLIOGRAPHY

[1] William James, Steve McKinzie, William Benson, et al. *Crash Investigation and Reconstruction Technologies and Best Practices*. English. FHWA-HOP-16-009. Oct. 2015. URL: `https://rosap.ntl.bts.gov/view/dot/50639`.

[2] Steven G. Shelby. *Delta-V as a Measure of traffic Conflict Severity*. Online; accessed 23/02/2022. 2021. URL: `https://onlinepubs.trb.org/onlinepubs/conferences/2011/RSS/1/Shelby,S.pdf`.

[3] Rick Ruth. “Using EDR Delta V in Traffic Crash Reconstruction”. In: (2019).

[4] Pavlína Moravcová, Kateřina Bucsuházy, Martin Bilík, et al. “Let It Crash! Energy Equivalent Speed Determination”. In: Jan. 2021, pp. 521–528. DOI: `10.5220/0010449005210528`.

[5] Rafal Burdzik, Piotr Folega, L. Konieczny, et al. “Analysis of material deformation work measures in determination of a vehicle’s collision speed”. In: *Archives of Materials Science and Engineering* 58 (Nov. 2012), pp. 13–21.

[6] Raymond M Brach and R Matthew Brach. *Vehicle accident analysis and reconstruction methods*. 2011. URL: `https://www.sae.org/publications/books/content/r-397/`.

[7] Raymond Matthew Brach and Michael Brach. “Analysis of Collisions: Point Mass Mechanics and Planar Impact Mechanics”. In: *Collision: the international compendium for crash research* 2.1 (2007), pp. 38–50. ISSN: 1934-8681.

[8] Wojciech Wach. “Monte Carlo Method in analysis of road accidents versus interpretations of calculation results”. In: (Dec. 2014).

[9] Hermann Steffan. “Accident reconstruction methods”. In: *Vehicle System Dynamics* 47.8 (2009), pp. 1049–1073. DOI: `10.1080/00423110903100440`. eprint: `https://doi.org/10.1080/00423110903100440`. URL: `https://doi.org/10.1080/00423110903100440`.

[10] R Woolley. “Non-Linear Damage Analysis in Accident Reconstruction”. In: *SAE Technical Paper 2001-01-0504* (2001). DOI: `10.4271/2001-01-0504.`. URL: `https://doi.org/10.4271/2001-01-0504`.

[11] C Riviere, Philippe Lauret, J.F. Ramsamy, et al. “A Bayesian Neural Network approach to estimating the Energy Equivalent Speed”. In: *Accident; analysis and prevention* 38 (Apr. 2006), pp. 248–59. DOI: `10.1016/j.aap.2005.08.008`.

[12] Ricardo J. F. Portal and João M. P. Dias. “Pedestrian reconstruction tools applied to pedestrian accidents in Portugal”. In: 2009.

[13] Costin D. Untaroiu, Mark U. Meissner, Jeff R. Crandall, et al. “Crash reconstruction of pedestrian accidents using optimization techniques”. In: *International Journal of Impact Engineering* 36.2 (2009), pp. 210–219. ISSN: 0734-743X. DOI: `https://doi.org/10.1016/j.ijimpeng.2008.01.012`. URL: `https://www.sciencedirect.com/science/article/pii/S0734743X08000742`.

[14] Kirsten Matheus and Thomas Königseder. *Automotive Ethernet*. 3rd ed. Cambridge University Press, 2021. DOI: `10.1017/9781108895248`.

[15] Dot Hs. “Analysis of Event Data Recorder Data for Vehicle Safety Improvement”. In: 2008.

[16] UNECE. *Proposal for the 01 series of amendments to UN Regulation No. [XXX]on Event Data Recorder*. Online; accessed 22/02/2022. 2021. URL: `https://www.eudarts-group.com/_files/ugd/fcfed0_23b2541abc7c4fcd8f17042b7f96e563.pdf`.

[17] VADIRAJ PATIL. “Generic and complete vehicle dynamic models for open-source platforms”. In: (2017). URL: `https://publications.lib.chalmers.se/records/fulltext/251558/251558.pdf`.

[18] JORGE GÓMEZ FERNÁNDEZ. “A Vehicle Dynamics Model for Driving Simulators”. In: (2012). URL: `https://publications.lib.chalmers.se/records/fulltext/160346.pdf`.

[19] Ramanata PP. “Vehicle Dynamics Modeling - Chapter 2”. In: (1998).

[20] Parviz E. Nikravesh. “Computer-Aided Analysis of Mechanical Systems”. In: (1988).

[21] Mansor Wan and Muhamad Wan. “Vehicle Steering Dynamic Calculation and Simulation”. In: (2018).

[22] IPTM Symposium on Traffic Safety. “State of EDR in the US CDR Update June 2022”. In: (2022). URL: `https://iptm.unf.edu/uploadedFiles/symposium/handouts/Ruth-USEDRStatus06032022.pdf`.

[23] Bob Scurlock, Andrew Rich, and Kyle Poe. “Corrections to Off-Axis Δv Measurements from Event Data Recorders”. working paper or preprint. Apr. 2021. URL: `https://hal.archives-ouvertes.fr/hal-02319921`.

[24] StanislavPetrovV. “Software 3D engine”. In: (). URL: `https://github.com/StanislavPetrovV/Software_3D_engine`.

DECLARATION

Declaro, sob compromisso de honra, que o trabalho apresentado nesta dissertação, com o título *“Development of new methodologies for road accident reconstruction with CDR Tool”*, é original e foi realizado por André Gaspar Francisco (2202281) sob orientação de Professor Sérgio Pereira dos Santos (ssantos@ipleiria.pt) e coorientação de Professor Carlos Daniel Henriques Ferreira (ferreira@ipleiria.pt).

Leiria, Setembro 2022

André Gaspar Francisco