UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

CARLOS AUGUSTO BOHM AGUSTI

ELECTRONIC DIFFERENTIAL FOR ELECTRIC VEHICLES POWERTRAIN CONTROL

Porto Alegre
2018

UNIVERSITÉ DE MONTPELLIER
POLYTECH MONTPELLIER
MEA - Microélectronique et Automatique

PROJET INDUSTRIEL DE FIN D'ÉTUDES

Carlos Augusto BOHM AGUSTI

Porto Alegre, Brésil

2018

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ELECTRONIC DIFFERENTIAL FOR ELECTRIC VEHICLES POWERTRAIN CONTROL

Projeto de diplomação apresentado para o Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, parte dos requisitos para Graduação em Engenharia Elétrica.

Orientador:
Prof. Dr. Luiz Tiarajú dos Reis Loureiro

Porto Alegre
2018

CARLOS AUGUSTO BOHM AGUSTI

ELECTRONIC DIFFERENTIAL FOR ELECTRIC VEHICLES POWERTRAIN CONTROL

Este projeto foi julgado adequado para fazer jus aos créditos da Disciplina de "Projeto de Diplomação", do Departamento de Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Prof. Dr. Luiz Tiarajú dos Reis Loureiro, UFRGS
Orientador - UFRGS

Banca Examinadora:

Prof. Dr. Paulo Roberto Eckert
Doutor pela UFRGS - Porto Alegre, Brasil

Prof Dr. Roberto Petry Homrich
Doutor pela UNICAMP - Campinas, Brasil

Prof. MSc. Igor Pasa Wiltuschnig
Mestre pela UFRGS - Porto Alegre, Brasil

Dedication

Dedico esse trabalho aos meus pais, João Carlos e Silvia, por toda dedicação e apoio, durante minha vida e graduação. Dedico também, aos meus avôs, Waldir e Santos, por serem homens sonhadores e empreendedores, características que marcaram seus descendentes.

Acknowledgments

Firstly, I would like to express my gratitude to my Brafitec scholarship advisor Prof. Dr. Eng. Altair Sória Pereira for the continuous support and guidance when I was in France, studying for my double degree in Electrical Engineering.

My deepest appreciation to Dr. Eng. Luiz Tiarajú Loureiro, my thesis and internship advisor, for his insightful comments and encouragement, but also for the hard questioning that pushed me to widen my research from various perspectives.

My sincere acknowledgment to professors PhD. Ing. Mariane Comte, PhD. Ing. Laurent Latorre and the whole Université de Montpellier faculty, great masters and teachers, for the warm welcome, advice, knowledge and outstanding support when Cassiano, João Victor and I were in France . It was a great honor to be a Polytech Montpellier student.

I am so grateful to have a family which has always supported me, giving the opportunity to follow my goals. João Carlos and Silvia, my parents and inspiration, continually stimulated my development and taught me virtues that I will carry for my entire life.

I thank my fellow colleagues for the sleepless nights we spent together before deadlines and all the fun we have had in the last six years. In addition, Gustavo, André, Marcelo, Gilmar, Luiza, Sara and Victor, my dearest friends, were very important to my personal and professional development. They have had a decisive role in my life, improving my way of thinking, giving me the courage to follow my dreams always looking for my well-being. Often, our endless conversations helped me to become who I am and who I want to be.

Furthermore, I would like to thank Leomar, Vinícius, Carlo and e-Power UFRGS Racing Team for their friendship, understanding, vision and assistance in this project.

"You can't connect the dots looking forward; you can only connect them looking backwards. So you have to trust that the dots will somehow connect in your future. You have to trust in something - your gut, destiny, life, karma, whatever."

— Steve Jobs

"With regard to performance, commitment, effort, dedication, there is no middle ground. Or you do something well or not at all."

— Ayrton Senna da Silva

Contents

List of Figures

List of Tables

List of equations

List of Abbreviations

AC/DC	Alternative current/direct current
AC	Alternative current
APP	Acceleration pedal position
CAN	Controller Area Network
DC	Direct current
ECU	Electronic control unit
EDS	Electronic Differential System
ED	Electronic Differential
EM	Electric machine
EPS	Electronic Power Steering
ESC	Electronic Stability Control
EVs	Electric cars
EV	Electric vehicle
FL	Fuzzy Logic
FSAEBR	Formula SAE Brazil
FSAEE	Formula SAE Electric
FSAE	Formula SAE
I/O	Input/Output

ICE	Internal combustion engines
IM	Induction motors
MCU	Microcontroller Unit
MIMO	Multi-input and multi-output
NFS	Neuro-fuzzy system
OP	Open differential
PID	Proportional-integral-derivative
PMSM	Permanent magnet synchronous machines
PT	Power-train
PWML	Left control pulse width modulation
PWMR	Right control pulse width modulation
RPM	Revolution per minute
SAE	Society of automotive engineers
SISO	Single input - single output
SPI	Serial Peripheral Interface
SWAS	Steering wheel angle sensor
SWA	Steering wheel angle
SW	Steering wheel

Nomenclature

δ_1	Outer steering wheel angle

δ_2	Inner steering wheel angle
ω	Gravity center angular speed in rad/s
ω_1	Front left wheel angular speed
ω_2	Front right wheel angular speed
ω_3	Rear left wheel angular speed
ω_4	Rear right wheel angular speed
ω_v	Vehicle angular speed in rad/s
θ	Steering wheel turning angle
d_r	Distance between rear wheels
K	Distance between the left and right kingpin
m	Mass in kilograms (kg)
N	Number of counted pulses inside time observation window
N_{pr}	Number of pulses inside one complete revolution
R_1	Resistor 1
R_2	Resistor 2
R_{cg}	Radius of gravity center
T_{tw}	Time window during which speed is estimated based on the counted number of encoder pulses
V	Vehicle speed
V_{out}	Output Voltage
V_{Ref}	Reference Voltage

Abstract

Seeking for a more sustainable world, society is increasingly exploring the use of systems that have a lower environment impact. The electric vehicle has a great potential to reduce urban pollution and recently has gained prominence in automotive engineering research.

The differential plays an important role for vehicles. In multi-drive systems the motor controllers must additionally be configured to provide an electronic differential effect. Electronic differential is a device which controls the electric machine's rotational speed and provides the required torque for each driving wheel, allowing different wheel speeds. It is used in multi-drive systems due to current vehicle velocity, steering wheel angle and accelerator pedal position.

This work analyzes the possibility of using an electronic differential in a Formula SAE vehicle. In order to do so, a system composed by an electronic central unit, sensors, signal conditioners, drivers and electric machines, is proposed for verification of the differential effect in electric vehicles.

In this system, tests have been carried out: steering wheel angle sensor, accelerator pedal position, PWM pulse width comparison for electric machines speed control and electric machines rotational speed. Also, data was collected in order to characterize the device.

The results present an expressive similarity according to simulations, validating the system, thus enabling the possibility for electric machines control in multi-drive vehicles.

Keywords: **Electronic Differential, Electric Machines, Electric Vehicles, Fuzzy Controller, Fuzzy Logic, Multi-Drive Systems, Powertrain Control.**

Résumé

À la recherche d'un monde plus durable, la société étudie de plus en plus l'utilisation de systèmes ayant un impact moindre sur l'environnement. Le véhicule électrique a un grand potentiel pour réduire la pollution urbaine et a récemment gagné en importance dans les recherches en ingénierie automobile.

Le différentiel joue un rôle important pour les véhicules. Dans les multi-drive systèmes, les contrôle des machines electriques doivent être configurés pour fournir un effet différentiel électronique. Le différentiel électronique est un dispositif qui contrôle la vitesse de rotation des machines électriques et fournit le couple requis pour chaque roue motrice, permetent différentes vitesses de rotation. Il est utilisé dans le multi-drive systèmes en raison de la vitesse actuelle du véhicule, de l'angle du volant et de la pédale d'accélérateur.

Ce travail analyse la possibilité d'utiliser un différentiel électronique dans un véhicule de Formule SAE. Pour ce faire, un système composé d'une centrale électronique, de capteurs et leurs circuits, drivers et machines électriques, est proposé pour vérifier l'effet différentiel dans les véhicules électriques.

Dans ce système, des tests ont été effectués avec : le capteur d'angle du volant, position de la pédale d'accélérateur, étude de la largeur du pulse PWM pour le contrôle de la vitesse des machines électriques et la vitesse de rotation des machines électriques. De plus, des données ont été collectées afin de caractériser le dispositif.

Les résultats présentent une expressive similarité selon les simulations, validant le système, permettant ainsi la possibilité de contrôler les machines électriques dans les multi-drive véhicules.

Mots-clés : **Commande du groupe motopropulseur, Différentiel électronique, Multi-drive systèmes, Machines électriques, véhicules électriques, logique Fuzzy.**

1 Introduction

Seeking for a more sustainable world, electric vehicles are helping to generate a cleaner, more energy-efficient future. Considering recent years, this sector has been growing exponentially, not only by the increase in market demand, but also for technological development and the growing policy of reducing pollutants emission, reinforcing world community environmental awareness.

In road transportation systems, differential plays an important role for vehicles. It contributes to a different wheel torque distribution while the car is turning or being driven in irregular terrain. There are two distinct categories: single−drive system and multi−drive systems. The first one, mechanic differential, is mostly used in internal combustion engine vehicles. Whereas, in multi-drive systems the motor controllers must additionally be configured to provide an electronic differential effect.

Furthermore, electric vehicles are not particularly suitable for those who use separate drives for both rear wheels. The electronic differential system (EDS) constitutes the latest technological developments in the design of electric vehicles, adding better control and balancing of a vehicle in curvy roads [1].

Electronic differential is a device which controls electric machines rotational speed and provides the required torque for each driving wheel, allowing different wheel speeds. It is used in place of the mechanical differential in multi-drive systems due to current vehicle velocity, steering wheel angle and accelerator pedal position. When a vehicle is turning, due to wheels turning radius, the inner and outer wheels rotate at different speeds so them were supplied with the torque they need. Considering our vehicle, three packages of sensors were placed in order to collect these values, side by side with control circuits, to collect current car data and send it to an electronic control unit. ECU is connected to an inverter, the interface between power source and rotating machinery which has the amount of energy and power needed to drive a race car, establishing an expected and reliable operation.

Several applications of this technology have proven successful and have increased vehicle performance. Dynamic characteristics of each mechanical system influence the performance of traction control/stability capability. The proposed electric machine based differential system based traction control/stability presents a significant improvement compared to conventional systems due to fast torque response in electric machines [2].

Additionally, electronic differential has some advantages over a mechanical differential: allows distributed regenerative braking options, faster response times, accurate knowledge of traction torque per wheel and gives the possibility to include new features according to driver's preferences. Also, it avoids gearbox and other mechanical parts and contributes to better traction and stability control.

Following the previous statements, this thesis' main objective is to evaluate a new electronic differential control system and consequently an electric vehicle powertrain management. First, it is proposed to complete a study based on existent topologies and technologies. Subsequently, simulations will be carried out in order to analyze the response of proposed system, always focusing on future development and easy device tuning, according to EV determined settings. That electronic control unit will achieve data from sensors and translates it into a signal that could be used by inverters to electric machines speed control.

In order to characterize the electronic differential system for an electric vehicle, driven by two electric machines attached to the rear wheels, using direct torque fuzzy control, multiple simulations have been carried out: electronic differential effect and fuzzy accelerator. These tools have also been used to design the electric circuits, software programming and the system design. The simulation platform is validated experimentally as an accurate representation of the prototype and its performance in terms of realistic vehicle dynamics.

The results will be presented in order to allow a future development for the vehicle of UFRGS ePower Racing team.

2 Literature Review

In this chapter we will discuss a review of electric machines basic concepts, a vehicle mathematical model, instrumentation, electronics and power electronics. These ideas will be approached according to the proposed area for application of this work, that is to say, electronic differential applications in an electric vehicle. Also to be presented are sensors, methods and work results already done in respective area of interest.

2.1 Electric vehicles general properties

In a world where electric cars have been identified as being a key technology, several devices should be developed to improve cars reliability and safety.

Considering electric cars, the biggest discrepancy compared to vehicles powered by internal combustion engines are: transmission, energy storage, power-train and drive-train systems.

- **Batteries**:

  Batteries are used to power the propulsion of EV. Observing the different types of batteries available on market, we can mention: Lead-acid, Nickel metal hydride and Lithium-ion.

  Lead-acid batteries have their benefits , and as a result, have been widely adopted for current internal combustion engine (ICE) vehicles, for a specific purpose. Lead-acid batteries have a long shelf life, are inexpensive, reliable, easily recyclable, and are safe when properly handled and maintained [13].

  Nickel-metal hydride (NiMH) battery is another rechargeable battery widely produced for commercial applications in EV. NiMH batteries were used to supply energy to the electric motor because they offered a higher specific energy than lead-acid batteries and had a much better energy density (watt-hours per liter) compared to lead-acid batteries. NiMH batteries are also highly reliable, and safe similar to lead-acid batteries. Also, they are composed of non-corrosive substances resulting in safer handling and recycling.

e-Power race car utilize lithium-ion batteries to supply enough energy needed for electric machines. Observing it, large quantities of battery cells are needed, which means that the battery system is one of the heaviest components of the car. It is very important for a race car to have a low weight, hence Lithium Ion battery is chosen as tractive system accumulators for its high energy density.

The main factors that should be considered before selecting the battery are: specific energy and its maximum discharge current [14].

Figure 1 demonstrates the impact of battery pack cost for electrical cars.

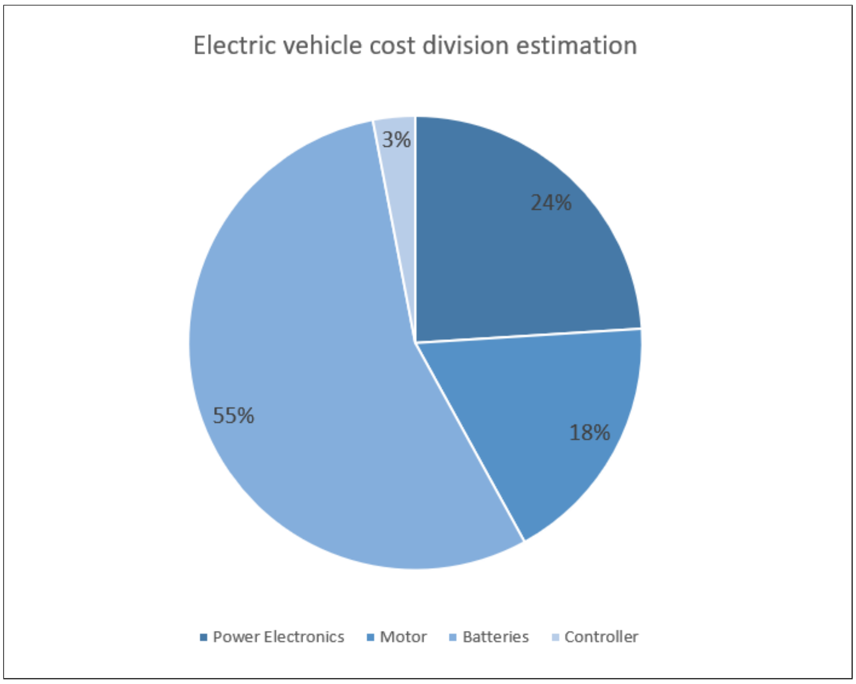

Figure 1. Cost estimation of an EV.

Source: Technology - Synthesis Partners, LLC (2011) [15]

.

- **Drive-train system**:

While an internal combustion vehicle has a multi-speed gearbox with numerous ratios, nearly every electric car has a single-speed transmission. This occurs because an electric motor delivers its maximum torque at zero RPM. Unlike an internal combustion engine, it doesn't need a system to disconnect it from the drive-train to allow it to idle while the vehicle is stopped.

Additionally, most of electric motors have a much larger RPM range than the typical internal combustion engine. Unlike a gas or diesel engine, an electric motor makes its best power output over an incredibly RPM range. Therefore, designers of electric cars just pick a gear ratio that provides a good compromise between acceleration and top speed. The typical electric motor is capable of sustaining high revolutions per minute, so the top speed often is not even a limiting factor.

When a multi-drive configuration is selected, the presence of an electronic differential is mandatory for drive-train system.

- **Power-train system**:

Electric vehicle power-train is composed of different types of electrical machines and electric components. There are four types that have been adopted for this kind of car, namely the DC machines, induction, switched reluctance machine and permanent magnet brushless machine. PT is usually selected due to size, power and proposed application. For example, Tesla Model S has a three-phase, four pole AC induction machine, rear mounted. Also, this power-train provides regenerative braking power of more than 60 kW which reduces both energy consumption and improves brake lifetime.

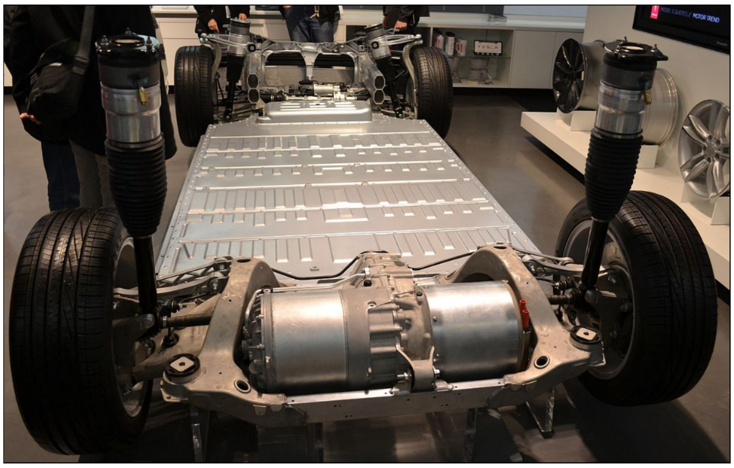

Figure 2. Tesla Motors Model S base

Source: Oleg Alexandrov (2014) [16]

.

2.1.1 Advantages and disadvantages

Production and utilization of EV bring many benefits and differences when compared to combustion models. These differences causes environmental impacts, changes in the operation costs, efficiency and power grid lines adaptation.

Environmental: Conventional cars with internal combustion engines (ICE) are a major source of air pollutants such as carbon dioxide (CO_2.), nitrogen oxides (NOx), black carbon (BC). From an environmental point of view, the replacement of ICE cars by electric vehicles, (EV) may be beneficial for the climate because of the reduction of greenhouse gas emissions, particularly CO_2. On the other hand EV are not 100% clean. When EV are charged, the electricity needed, is produced by a wide range of different power plants which may also be sources of air pollution [17].

Electric grids: EVs may be considered as active loads, increasing the demand on the network during charging, and as generators, when operating in regeneration mode. Due to this, large deployment of EVs is expected to lead to potential problems for existing power networks. Under certain operating conditions, they may also lead to power quality problems and voltage imbalance. EV interface devices may be designed to minimize or even eliminate the effects of EVs on the network. In fact, with appropriate control and communication with the grid, EVs could be designed to operate as part of a 'smart grid' to provide services such as supply/demand matching and voltage/frequency control [20]. Furthermore, current electricity infrastructure may need to cope with increasing shares of variable-output power sources. This variability could be addressed by adjusting the speed at which EV batteries are charged.

Operating cost: According to Shisheng Huang, it appears that the first generation EVs will not be economically competitive with conventional and hybrid vehicles, even with government incentives. Its competitiveness is further hampered by the current structure of the electricity pricing structure. The high costs of electricity compromise EVs ability to replace gasoline with electricity in a cost-efficient manner. Developing new technologies and consequently lowering EVs prices, vehicles will become competitive and look more interesting to the market.

Efficiency: EV efficiency is about a factor of 3 higher than internal combustion engine vehicles. Energy is not consumed while the vehicle is stationary, unlike internal combustion engines which consume fuel while idling. However, looking at the efficiency of EVs, their total emissions are closer to an efficient gasoline or diesel in most countries where electricity generation relies on fossil fuels [22].

2.1.2 Formula SAE electric race vehicle

This work's main objective, will be to build and validate a Formula SAE car electronic differential. FSAEE has some specific characteristics and rules to compete, which are certified by Society of Automotive Engineers.

FSAE e-Power UFRGS car is currently at project phase and it was entirely developed by university students. The main purpose of this project is to develop the electric machine itself and its controls (including electronic differential) that will be put to test in FSAEBR championship.

An open-wheels electric car has some particular characteristics, as shown in the following table:

e-Power Racing car overview	
Brakes	4-wheel hydraulic disc brakes
Suspension	Double wishbone with coils springs, oil damper
Power-train	Two PMSM electric machines
Drive-train	Electronic differential, inverters and sensors
Steering system	Rack-pinion system
Fairing	Fiberglass

Table 1. UFRGS electric vehicle general properties

At first, the vehicle design was made with only one electric machine, using a mechanical differential as torque transmission for wheels. However, it was identified that component would bring excessive weight to the car, so an application of electronic differential, the purpose of this project, has become a major improvement solution.

The developed vehicle is an electric mono-post, with electric motors of up to 80kW, fed by a pack of batteries. Some of the events that test the car are: design, manufacturing, cost and business estimations, acceleration and endurance. Figure 3 presents an example of FSAEE car.

Figure 3. Unicamp E-Racing electric vehicle prototype

Source: Unicamp e-team media (2017) [18]

.

2.1.2.1 Traction control and vehicle stability

Vehicle stability is subdivided into lateral stability, yaw stability and roll stability. Vehicle stability control assists in reducing under steer, over steer, instability around yaw axis and roll over conditions. Vehicle research shows that active front steering and rear steering provides improvement in stability control capability [5]. Electric machine differential provides an additional degree of control due to the independent wheel torque control capability. Furthermore an electric machine provides a faster torque response.

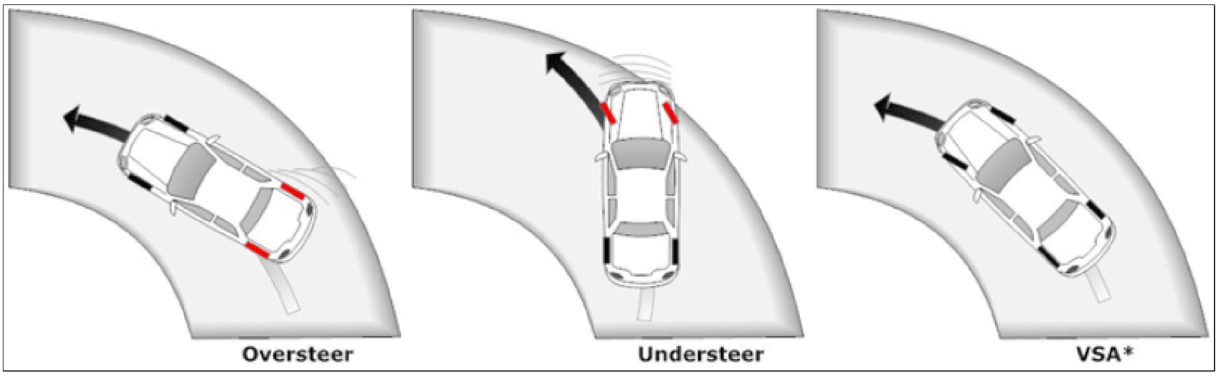

Figure 4. Under steer and over steer movements.

Research has shown that 20 - 25% of car accidents resulting injuries or fatalities were due to spinning cars. In other words, it presents the fundamental importance of control car systems in passenger safety. To solve and improve turning issues, an electronic differential was created for multi-drive systems control, where a mechanical differential configuration is not adequate for torque management.

Fuzzy logic control optimizes the slip rate on specified limit. Fuzzy based ED with third harmonic voltage integration is highly suitable for the two in wheel motor drive electric vehicle which gives high range of stability with improved efficiency [12]. This logic will be implemented in our electronic differential to motor control as well as improve traction control and stability.

2.2 Vehicle Differential

A differential is a device, usually but not necessarily employing gears, capable of transmitting torque and rotation through three shafts, almost always used in one of two ways: in one way, it receives one input and provides two outputs - this is found in most automobiles - and in the other way, it combines two inputs to create an output that is the sum, difference, or average, of the inputs. In automobiles and other wheeled vehicles, the differential allows each of the driving road-wheels, for most vehicles, supplying equal torque to each of them [3].

Basically, considering vehicles, the differential allows the outer drive wheel to rotate faster than the inner drive wheel during a turn. This is necessary when the vehicle turns, making the wheel that is traveling around the outside of the turning curve roll farther and faster than the other [4]. An increase in the speed of one wheel is balanced by a decrease in the other. During a corner, when the car is turning left, left wheels have a lower angular velocity than the other ones from right side. Figure 6 presents the difference between vehicle wheels.

The Ackermann-Jeantand model discovered by Rudolf Ackermann in the 19th century is used for EDS design [7]. When an electric vehicle is driven on a curved road, this model is generally preferred at low speeds due to the effect of centrifugal force and centripetal forces. The derived equations from the Ackerman-Jeantand model are as follows:

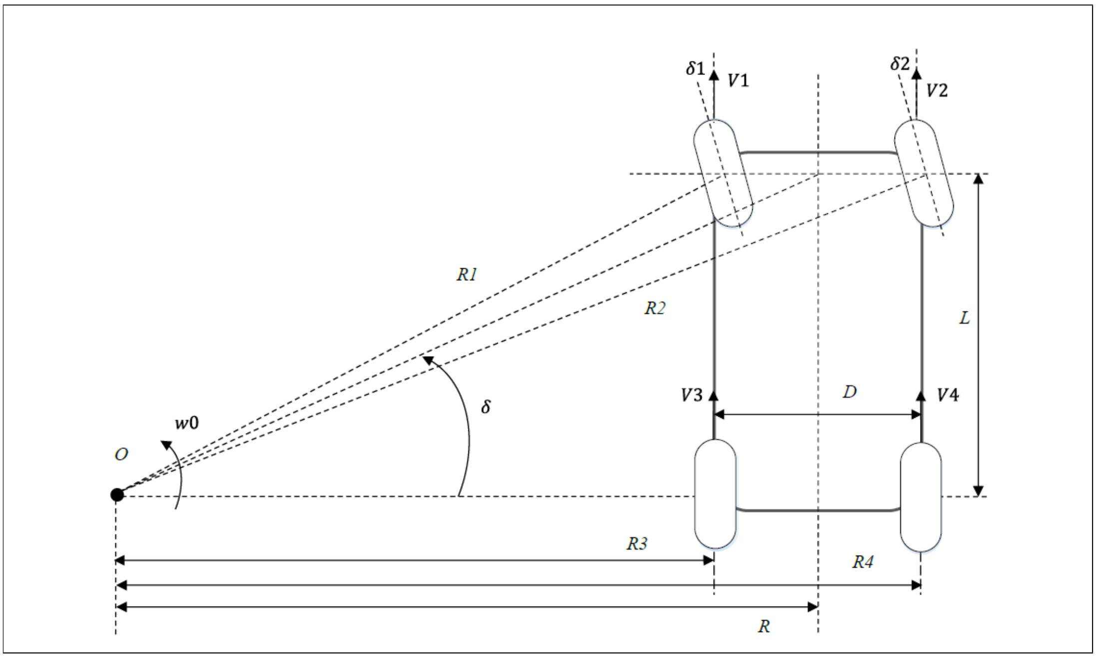

Figure 5. Vehicle model proposed system in left turning maneuver.

Source: EMD For Vehicle Traction Control And Stability Control [6]

.

$$\delta_1 = arctan\left[\frac{L \cdot tan(\delta)}{L + ((K/2) \cdot tan(\delta))}\right] \tag{1}$$

Outer steering wheel angle of the front wheel

$$\delta_2 = arctan\left[\frac{L \cdot tan(\delta)}{L - ((K/2) \cdot tan(\delta))}\right] \tag{2}$$

Inner steering wheel angle of the front wheel

According to Ackermann-Jeantand, where K is the right and left kingpins distance, to have an estimation of each wheel speed, the turning radius of inner and outer wheels could be presented as following equations (3), (4), (5), (6):

$$R_1 = \frac{L}{sin(\delta_1)} \tag{3}$$

$$R_2 = \frac{L}{sin(\delta_2)} \tag{4}$$

$$R_3 = \frac{L}{tan(\delta)} - \frac{d_r}{2} \tag{5}$$

$$R_4 = \frac{L}{tan(\delta)} + \frac{d_r}{2} \tag{6}$$

where d_r is the distance between rear wheels.

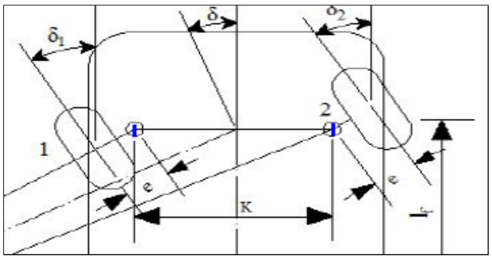

Figure 6. Right and Left kingpins distance (K).

Source: Design of EDS with In-Wheel Motor [8]

.

where K is the distance between the left and right kingpin, L is the distance between the front and rear wheel.

The radius of the center of gravity R_{cg} of an EV is:

$$R_{cg} = \sqrt{(R_3 + (d_r/2))^2 + (l_r)^2} \tag{7}$$

where l_r is the distance between the rear wheel and center of gravity [8].

According to that mathematical modeling, it is possible to express angular speeds of inner and outer wheels:

$$\omega_1 = \frac{V \cdot R_1}{R_{cg} \cdot r} \tag{8}$$

$$\omega_2 = \frac{V \cdot R_2}{R_{cg} \cdot r} \tag{9}$$

$$\omega_3 = \frac{V \cdot R_3}{R_{cg} \cdot r} \tag{10}$$

$$\omega_4 = \frac{V \cdot R_4}{R_{cg} \cdot r} \tag{11}$$

where r is the radius of the wheel and V is the vehicle speed in m/s.

Also, steering angle signal indicates turning angle:

$$\delta > 0 \text{ turning right}$$

$$\delta = 0 \text{ going straight}$$

$$\delta < 0 \text{ turning left}$$

The difference between angular speed of wheel drives is expressed by equation 12, where ω is the vehicle center of gravity angular speed [32].

$$\Delta\omega = \omega_1 - \omega_2 = \frac{d \cdot tan(\delta)}{L} \cdot \omega \tag{12}$$

Equations 13 and 14 represents, respectively, the left and right rear wheels angular velocity in rad/s.

$$\omega_{Left} = \omega_v \cdot \left[\frac{L + ((\frac{D}{2}) \cdot tan(\delta))}{L} \right] \tag{13}$$

Left rear wheel angular velocity

$$\omega_{Right} = \omega_v \cdot \left[\frac{L - ((\frac{D}{2}) \cdot tan(\delta))}{L} \right] \tag{14}$$

Right rear wheel angular velocity

Equations from Ackermann-Jeantand will be used to set and apply vehicle parameters for Formula SAE EV and simulations.

2.2.1 Mechanical Differential

Differential plays an important role for vehicles. It contributes to a different wheel torque distribution while the car is turning or driving in irregular terrain. Mechanical differential is mainly used by internal combustion engine vehicles, most wide-spread type of commercial cars.

The modern differential was invented in 1827 by Pecquer and was first used on steam traction engines for the same reason that it is still used-cornering with the drive wheels mounted on a single shaft (spool) requiring a tire slip, especially on tight turns [9].

Furthermore, when pivots have no friction and the torque split must be even the lever is rotated due to differences in output velocity [9].

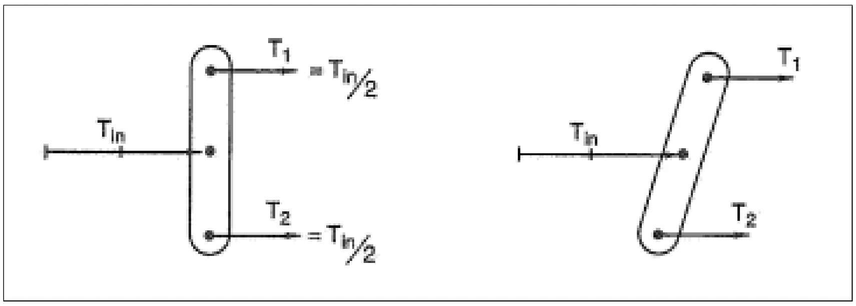

Figure 7. Lever analogy to different gear actions.

Source: Race car vehicle dynamics [9]

.

In most cases, the mechanical differential are the following types: Open differential, limited slip differential (LSD), locking differential and spool.

Open differential: An open differential architecture gear cleverly allows the sum of the outside and inside wheel rotational velocities to equal a constant while allowing one wheel to speed up and the other to slow down. Also, it allows the vehicle to go around corners without dragging the outside wheel. Another result of the differential mechanism is that torque is split equally between the two sides [9]. In addition, OP is the most common type among differentials and also the least expensive.

Limited slip differential: Limited slip differential's purpose is to limit or prevent the wheel with less traction to spin and the higher traction to remain immobile. Torque is transmitted to the wheel with less grip, to equalize the different speeds of rotation of the wheels on each side of the axle when the vehicle makes a turn. An example is presented in figure 8.

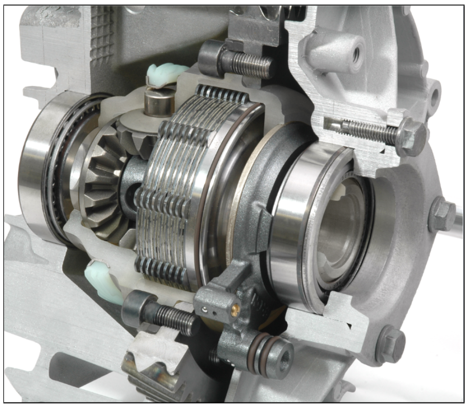

Figure 8. Eaton drivetrain limited slip differential

Source: Race car vehicle dynamics [9]

.

Locking differential: It is able to lock the two drive wheels on an axle together. The advantage is that both wheels have power at all times. The disadvantage is that turning is much more difficult since both wheels must turn at the same RPM. So most lockers must be disengaged when making sharp turns [10].

Spool: An open differential in which the axles have been mechanically fastened together. This does not allow either wheel to go faster or slower around corners. It is cheap and adds little or no weight to the vehicle, but is usually limited to off-road.

2.2.2 Electronic Differential

Electronic differential is a device that controls the electric machines rotational speed and provides the required torque for each driving wheel, allowing different wheel speeds. It is used in place of the mechanical differential in multi-drive systems due to current vehicle velocity, steering wheel angle and accelerator pedal position. In addition, cars equipped with electronic differential save weight and reduce drive-line complexity, a considerable improvement when we are considering high performance vehicles.

ED equips an unique platform for vehicle stability control, improving passenger comfort and range extension of an electric vehicle (EV). Less torque causes deceleration of the wheel resulting in re-adhesion and hence better traction is achieved. Also, the investigation is extended to an optimal slip ratio control. A road condition estimator is utilized to estimate the best slip command for the optimal slip ratio controller.Simulations show that lateral motion stabilization can be achieved with slip ratio based control on a four wheel driven electric vehicle by implementing slip ratio control [40].

Furthermore, based on extensive research led by Fujimoto, the battery life of an electric vehicle is a critical factor in evaluating the performance of an EV. Miles per charge of an EV estimate the battery pack capacity of it. Limitations in battery capacity is one of the foremost obstacles that engineers are facing, when an electric vehicle competes with the driving range capability of a gasoline based automobile. Researchers confirm a 3% reduction in energy loss by implementing a range extending traction control algorithm [41].

Figure 9 displays the ED block diagram and electronic control unit sensors schema, respectively.

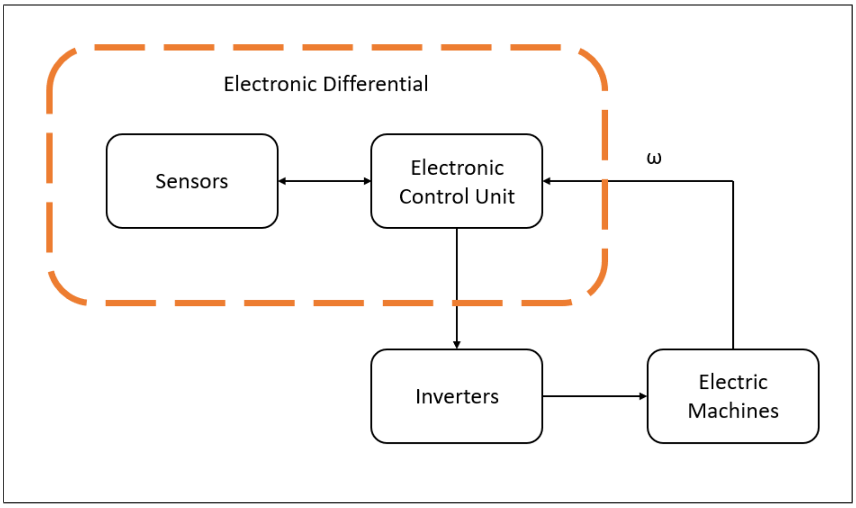

Figure 9. Electronic differential block diagram

Electronic control unit is responsible for analyzing the provided information and using a method to treat data of: steering wheel angle sensor, current vehicle speed and accelerator pedal pressure. Thereafter, Fuzzy method is applied to these variables and enables PWM outputs for inverter control.

Fuzzy logic, unlike the usual boolean logic, allows undetermined states to be treated by control devices. Thus, we could evaluate non-quantifiable concepts. For example, it is possible to estimate temperature (hot, warm, cold).

Control systems generally contain four elements - sensor, controller, actuator and system under control. ECU controller is usually a micro-controller which generates control signal in the form of PWM.

ED comprises many advantages and possibilities in comparison to mechanical differential. Fundamental advantages are the following [45]:

- Efficiency due to electric power train
- Ability to alter differential behavior with a control algorithm change
- Fast dynamic behavior due to electric machine
- Ability to implement traction control algorithms and ESP algorithms

2.3 Modeling the Electronic Differential

The usual configuration of electrical or non-electrical vehicles involves two wheels, driving one traction motor and a differential gear. In this work, we will adopt a structure with two independent wheel driving.

The in-wheel motor allows packaging flexibility by eliminating the central drive motor, associated transmission and driveline components in vehicles, including the mechanical differential.

The Electronic Differential consists of two electric machines, two driver units and a central control unit. Initially, ED design is for a rear wheel drive vehicle. An ECU controller is also interfaced to an accelerator pedal and a steering wheel input to provide information on driver's intent. Since the two rear wheels are directly driven by two separate motors, the speed of the wheel at the outer position of the curve will need to be greater than the speed of the inner wheel during curved steering (and vice versa). Also, this prevents the tires from losing traction when the vehicle is turning [32].

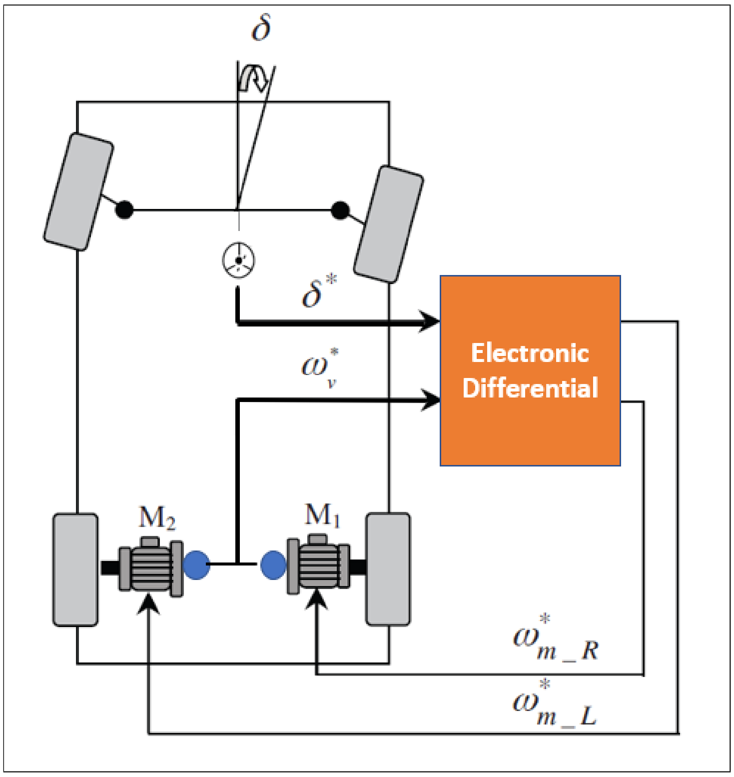

Figure 10. Electronic differential design model

Figure 10 demonstrates the electronic differential design model for a vehicle structure where M1 and M2 are electric machines, δ^* steering wheel angle, ω_v^* current electric machines rotational speed, $\omega_{m_R}^*$ and $\omega_{m_L}^*$ the right and left electric machine's angular velocity, respectively.

This ED system architecture permits the development of an electronic differential to: over a straight trajectory the two wheel drives roll exactly at the same velocity and in a curve trajectory the difference between the two wheel velocities assure that the vehicle is turning.

When a vehicle is cornering, due to the wheels' turning radius, the inner and outer wheels rotate at different speeds so they are supplied with the torque that they need. Considering a vehicle, two packs of sensors were placed in order to collect these values, side by side with control circuits, to obtain current car data and send it to an electronic control unit (ECU). It is connected to an inverter, the interface between power source and rotating machinery, having the power needed to drive a race car, establishing an expected and reliable operation. Also, several applications of this technology have proven successful and increase vehicle performance [2].

Input - Output ECU system overview	
ED system	Function
Input	Steering wheel angle sensor
	Accelerator pedal position sensor
Output	PWM to control left rear electric machine
	PWM to control right rear electric machine

Table 2. ECU I/O system overview

Often, an electronic control unit basically receives those sensor signals, processes all data and with that produces two PWM outputs, one for each inverter or driver.

2.4 Fuzzy Logic

Fuzzy logic is an approach to computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based. Among all the suggested methods for reactive navigation, fuzzy logic controller has been found to be the most attractive. The theory of fuzzy logic systems is inspired by the remarkable human capability to operate on and reason with perception-based information.

Fuzzy logic can be conceptualized as a generalization of classical logic. Modern fuzzy logic was developed by Lotfi Zadeh in the mid-1960s to model those problems in which imprecise data must be used or in which the rules of inference are formulated in a very general way making use of diffused categories. There are not just two alternatives but a whole continuum of truth values for logical propositions [24]. Also, applying a set of rules to determine output based on input values. Figure 11 presents a graphical representation of a conventional set (usual "true" or "false") and fuzzy logic set.

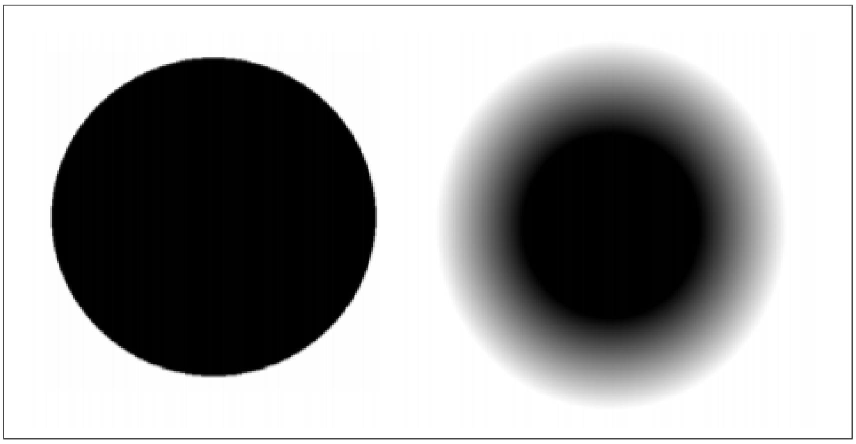

Figure 11. Graphical representation of a conventional set and a fuzzy set

Source: Introduction to fuzzy logic [25]

.

In diffused statements, concepts like “fast” and “slow” are imprecise but easy to interpret in a given context. In some applications, such as expert systems, for example, it is necessary to introduce formal methods capable of dealing with such expressions so that a computer using rigid Boolean logic can still process them. This is what the theory of fuzzy sets and fuzzy logic tries to accomplish.

Figure 12 shows membership functions for the concepts of speed. Based on membership degree, for example, a vehicle could be very slow or slow due to road conditions as surface imperfections or potholes.

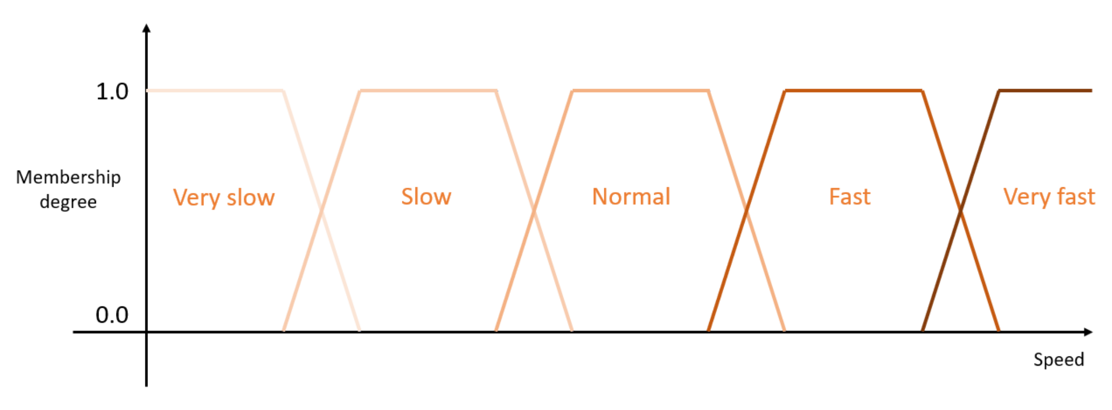

Figure 12. Membership functions for the concepts of speed

In addition, fuzzy logic depends on fuzzy rules which are defined as conditional statements. Basically, rules associate ideas and relate one event to another. The result of the application of a fuzzy rules construction include:

- Choice of process state and control output variable.
- Choice of the rule antecedent and rule consequence.
- Choice of term sets for the process state and control output variable.
- Derivation of the set of rule.

Fuzzy rules also operate using a series of if-then statements. For instance, if X then A, if Y then B, where A and B are all sets of X and Y. As we have defined the fuzzy operators AND, OR and NOT, the premise of a fuzzy rule may well be formed from a combination of fuzzy propositions.

Considering our ED system, fuzzy logic will be implemented as fuzzy accelerator speed reference controller, using input variable sets as steering wheel turning angle and accelerator pedal pressure. By introducing the notion of degree in the verification of a condition, thus enabling a condition to be in a state other than true or false, fuzzy logic provides a very valuable flexibility for reasoning, which makes it possible to consider uncertainties.

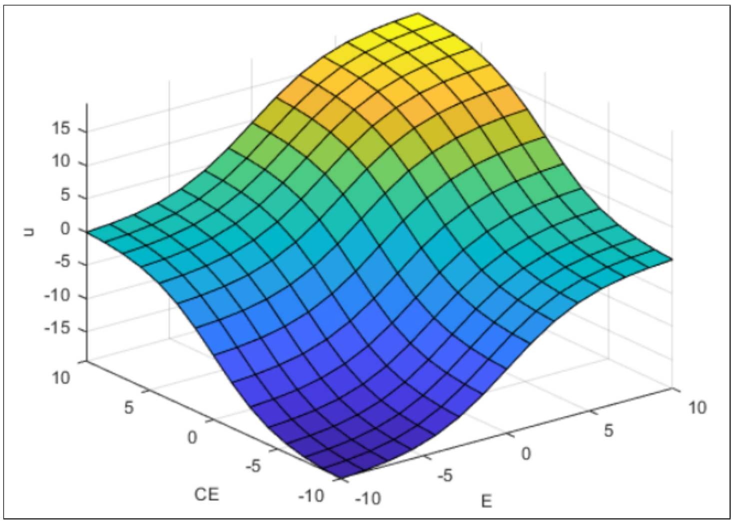

Figure 13. Fuzzy logic surface example

2.4.1 Defuzzification

Defuzzification is the process of producing a quantifiable result in FL and inverse of fuzzification, given fuzzy sets and corresponding membership degrees. It is typically needed in fuzzy control systems as an electronic differential control. Also, defuzzification is interpreting the membership degrees of the fuzzy sets into a specific decision or real value.

A fuzzy controller can use one of several mathematical methods to perform defuzzification: Center of Area , modified Center of Area or Mean of Maximum. Selecting a defuzzification method depends on the context of the design you want to calculate with the fuzzy controller.

2.4.2 Neuro-fuzzy system

Neuro-fuzzy systems were introduced in the thesis of Jyh-Shing Roger Jang in 1992 under the name "Adaptative-Networks-based Fuzzy Inference Systems" (ANFIS) [26]. They use the formalism of neural networks by expressing the structure of a fuzzy system in the form of a multilayer perceptron.

A multilayer perceptron (MLP) is a neural network without cycle. The input layer is given a vector network and the network returns a result vector in the output layer. Between these two layers, the elements of the input vector are weighted by the weights of the connections and mixed in the hidden neurons located in the hidden layer. Figure 14 illustrates an example of a neural network [25].

The fuzzy model implemented in the NFS determines the fuzzy rules format, which forms the fundamental knowledge part structure in a fuzzy inference system. However, the relevance function formats influence degrees of pertinence associated with each variable. In addition, triangular and trapezoidal shapes are the most used for the advantage of being computationally simple.

Space partition of input and output (I / O) variables internally defines fuzzy regions of space, which are related through the fuzzy rules [27]. Thus, division of the output space is usually simpler because it is associated with rules consequences. After doing fuzzy rule set evaluations, real value of the fuzzy rule set NFS is determined by the defuzzification method chosen. Basically, figure 14 describes all steps of fuzzy logic: initial variables, fuzzification, fuzzy rules, defuzzification and output, respectively.

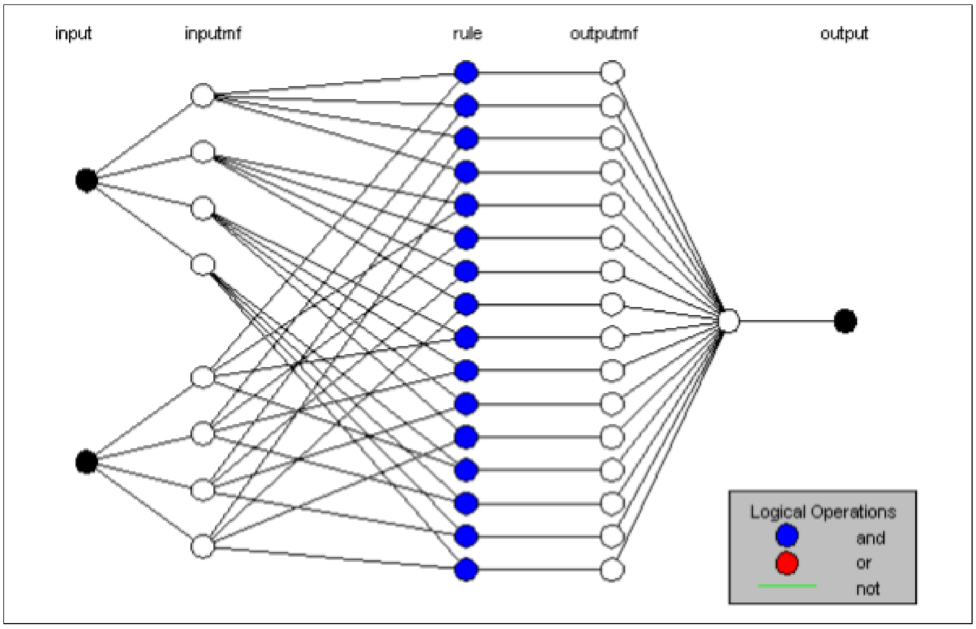

Figure 14. Structure of a neuro-fuzzy system

Source: Introduction to fuzzy logic [25]

.

NFS has the advantages of fuzzy systems, showing huge potential for applications that combine qualitative knowledge with robustness and provides a high-level, fast-computing and user-friendly interface to program, allowing the specialist to focus on functional objectives instead of mathematical details. However, neural-fuzzy system works with a reduced number of entries, due to exponential growth of combinatorial rules. A system with many inputs demands a great computational effort.

2.4.3 MISO Fuzzy control

The majority of process industries are nonlinear, Multi-Input Single-Output (MIMO) systems. The control of these systems is met with a number of difficulties due to process interactions, dead time and process nonlinearities. The difference between MISO systems control and single-input single-output (SISO) systems control is based on an estimation and compensation of the process interaction among each degree of freedom [33].

Fuzzy and neural control, in particular, have had an impact in the control community because of their simplicity to use control knowledge for control problems. The integration of fuzzy logic with neural network techniques has resulted in what is commonly referred to as neuro-fuzzy systems.

2.5 Electric machines for EV

Electric vehicles are powered by electric machines. Different types of EM are designed for automotive traction applications. Some of the requirements to improve traction applications are [11]:

- High instant power
- High efficiency
- High torque at low speed
- Reliability and robustness
- Wide constant torque region
- Efficient regenerative breaking
- Wide speed range region
- Reasonable cost

Many types of electric machines are currently been developed by manufacturers, mainly focused on performance and efficiency. Considering it, we have several EM conceptions which are chosen depending on the application. The most important are three phase electric machines, which are: Induction Motor, PMSM and synchronous brushed motor.

Induction motor: The three phase induction motors are simple in construction, low cost and easy to maintain. They run at a constant speed from no-load to the full load. Also, Tesla is the only large electric cars manufacturer to use induction motors (IM) in their vehicles.

Permanent magnet synchronous motor: In PMSM the rotor permanent magnets (PM) provide the excitation field of the rotor. The stator is essentially similar to induction machines and the applied 3-phase voltage creates a rotating magnetic field. As the induction machine rotor was dragged by the rotating magnetic field, in the synchronous machines the rotor speed synchronizes with the rotating stator field and thus it is the same as synchronous speed [36].

Switched reluctance motor: Switched reluctance motor (SRM) has many attractive characteristics that makes it a good option for EV traction motor. SRM provides high efficiency over wide speed range at a low cost and a simple control [37].

Cost of the electric machine, efficiency/loss and long term return on investment are major concerns in selecting an electric machine for a traction application. Also, permanent magnet machines and induction machines are widely used as traction motors due to their unique advantages [5].

Correlating induction and synchronous machines, we see that the last one is easier to cool, because the rotor generates less heat.

Zeraoulia [11] also made a comparison between electric machines, which one ranks: induction motor the highest, permanent magnet machine the next and switch reluctance machine and DC machine as the least preferred. An additional selection criterion is the market acceptance degree of each motor type, which is closely associated with the comparative availability and cost of its associated power converter technology.

This project focuses on a future real electronic differential application using Permanent magnet synchronous motor (PMSM) which is the one that e-Power UFRGS Racing has selected.

2.5.1 Permanent magnet synchronous motor (PMSM)

Synchronous machines have in their rotors two or more permanent magnets that generate a magnetic field which enters in the stator core (radial, axial or transverse directions) and interacts with flowing currents inside the windings, to produce the torque between rotor and the stator. A permanent-magnet synchronous motor uses permanent magnets embedded in the steel rotor to create a constant magnetic field.

The torque produced by PMSM is generated with two different components, which are PM torque and reluctance torque. The magnitudes of these components are dependent on the rotor structure and different rotor topologies that are shown in figure 15.

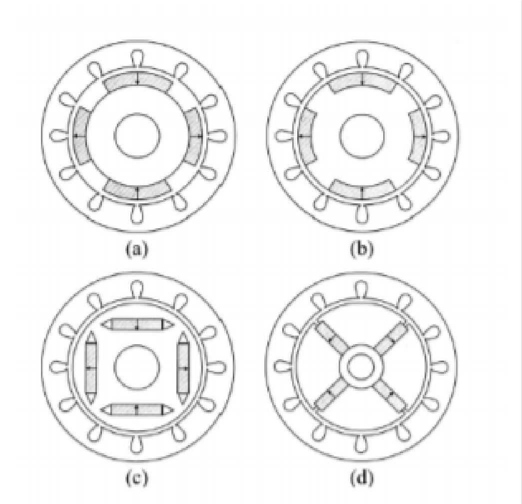

Figure 15. PMSM rotor structures: a) Surface mounted. b) Surface inset. c) Interior radial. d) interior circumferential.

Source: Modern Electric Vehicle Technology [38]

.

Considering the two different torque components, the torque produced is according to torque equation, given by:

$$T = \frac{3}{2} * p[\psi_m I_q - (L_q - L_d) I_q I_d] \tag{15}$$

PMSM torque equation

Where p is number of pair poles, ψ_m stator winding flux linkage due to the PMs, L_q and L_d the q-axis and d-axis stator winding inductances, respectively, and I_d and I_q the d-axis and q-axis currents, respectively [38].

2.6 Inverter and control

Inverter is an electronic device that changes the direct current (DC) to alternative current (AC). It is responsible for transforming batteries accumulated energy - or grid power - in order to control electric machines rotational speed or torque.

In electronic power converters and motors, pulse-width modulation (PWM) is used extensively as a means of powering alternating current (AC) devices with an available direct current (DC) source or for advanced DC/AC conversion. Variation of duty cycle in the PWM signal provides a DC voltage across the load. A specific pattern will appear to the load, as an AC signal, or can control the rotational speed of machines that would otherwise run only at full speed.

The output is controlled by changing the input DC voltage. In most cases, the DC voltage is changed by a controlled rectifier or DC chopper. The rms value of the AC voltage can be changed between 0 and a maximum value, as defined by the square-wave operation. This is a very simple method, in which the switching frequency of the semiconductor elements is equal to the output frequency. Therefore, it is the preferred method used in converters with high-frequency output. Modern inverters use insulated gate bipolar transistors (IGBTs) as the main power control devices. Besides IGBTs, power MOSFETs are also used especially for lower voltages, power ratings, and applications that require high efficiency and high switching frequency [28]. Figure 16 presents a three-phase inverter topology containing IGBT drivers to control PWM output frequency.

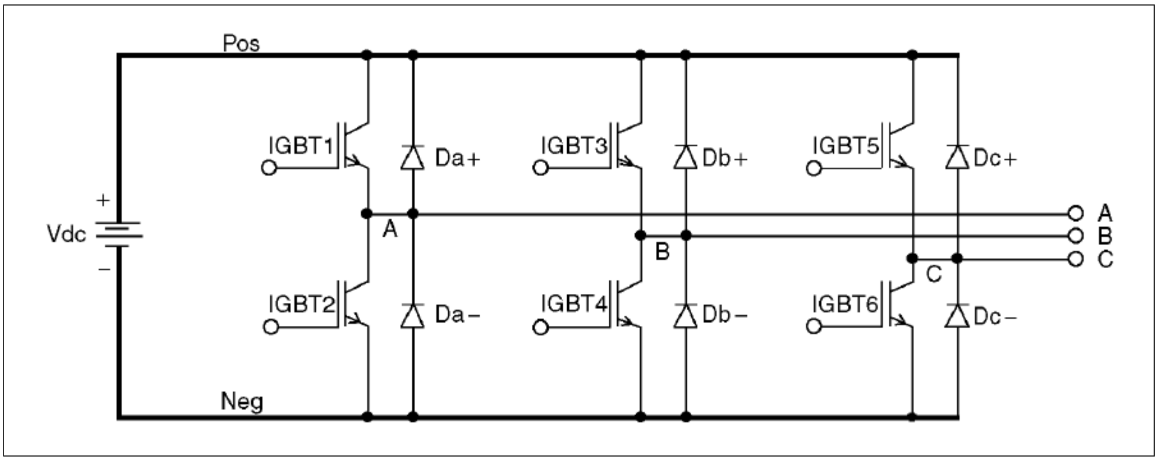

Figure 16. Topology of a three-phase inverter

Usually an inverter consists of:

- Rectifier: AC/DC converter if the power source is AC.
- Power Inverter: composed of controlled semiconductor gates that form a control signal which generate pulses for the load.
- Control: Microcontroller responsible to acquire sensor data, generating appropriated control signals.
- Surge protection device (SPD): protecting device for excessive current and supply electric machines voltages.

Figure 17 shows the typical output of a three-phase inverter during a startup transient into a typical motor load. The upper graph shows the pulse-width modulated waveform between phases A and B, whereas the lower graph shows the currents in all three phases. The motor acts a low-pass filter for the applied PWM voltage and the current assumes the wave-shape of the fundamental modulation signal with very small amounts of switching ripple.

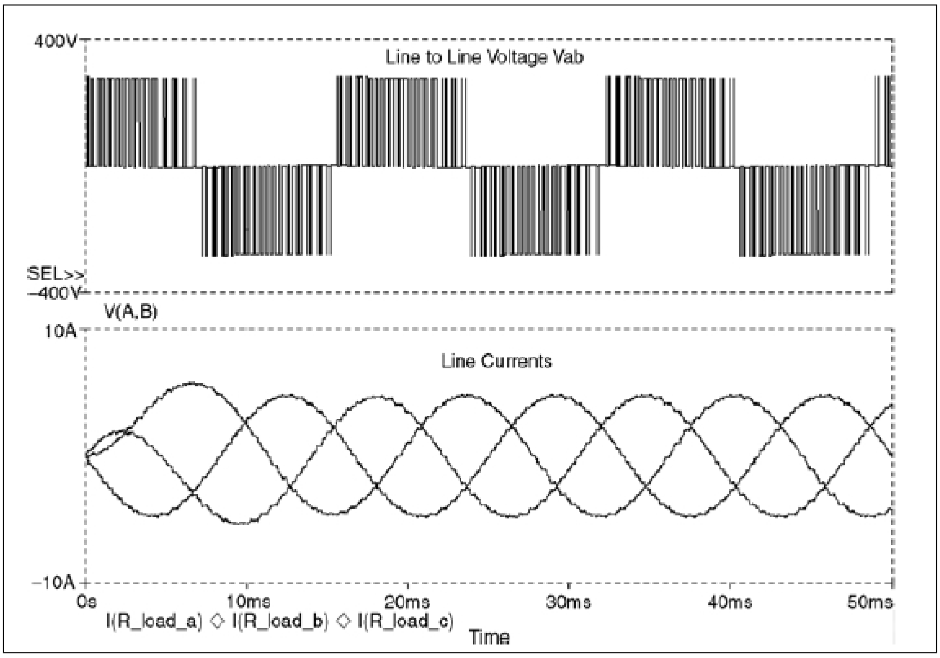

Figure 17. Typical waveforms of inverter voltages and currents

Source: Power electronics: Inverters [28]

.

An inverter allows the control of a electric machine rotational speed, establishing a connection between AC three-phase motor, energy supply and electronic control unit. ED output is formed by two pulse width modulation waves which vary their frequency and grant inverter gates control.

2.7 Sensors application

Electronic differential is a device which controls electric machines rotation speed and provides the required torque for each driving wheel. For that, two sensors were used in order to obtain this data, side by side with control circuits, to collect current car data and send it to an ECU.

The accelerator pedal pressure also can estimate how much pressure the driver is applying to it, sending a signal to the controller. In addition, steering wheel sensor is responsible for collecting SW angle and identify if the car is turning to the left or right.

2.7.1 Accelerator pedal position sensor

The accelerator pedal position sensor is fixed to the electronic accelerator body which has a functional principle that involves conversion of the accelerator pedal movement into a voltage signal. This sensor is responsible for providing information about accelerator pedal position. Also, it is used to estimate driver's applied pressure during driving.

The APP was used in Formula 1 initially by the McLaren team. This system makes use of an electronic assembly (pedal position sensor, motorized throttle body and the ECU) to accelerate the engine thus eliminating the old and problematic throttle cable. In addition, this system guarantees greater driver comfort and is multifunctional. Besides acceleration, it allows the possibility to analyze acceleration pedal position by the electronic central, eliminating additional sensors.

These sensors use potentiometers that work as voltage dividers. Voltage dividers use a resistive element and a wiper arm to "divide" an input voltage (called a reference voltage). They then send this "divided" voltage to a computer, which uses it to adjust the position of the throttle.

Figure 18 helps illustrate the basic principle behind how a voltage divider works. The resistive element, also called a carbon track, is basically a piece of graphite.

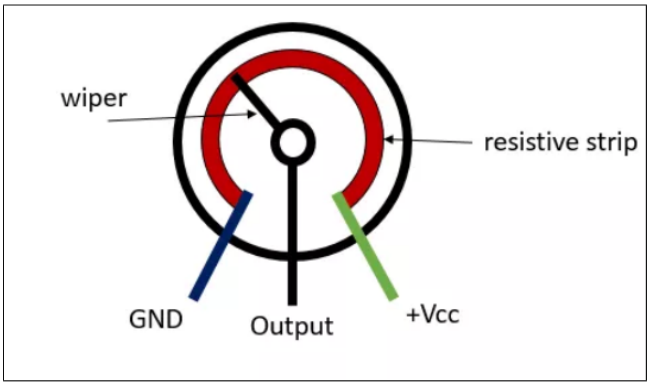

Figure 18. Potentiometer diagram

Moving the arm across the resistive element effectively alters the resistance on either side of the arm (R1 and R2). Moving the wiper clockwise increases R2 and decreases R1 and moving it counterclockwise does the opposite. Where GND is ground, Vcc is reference voltage source and Output is the output voltage. The exact relationship between the output voltage, reference voltage, and position of the wiper arm can be written as an equation:

$$V_{out} = V_{ref} * \frac{R_2}{R_1 + R_2} \tag{16}$$

Furthermore, to keep potentiometer input voltage more stable, several devices that produce reference voltage are usually utilized and thereby decrease the variations in the input and output voltages of the circuit, making signal acquisition more reliable and accurate.

2.7.2 Steering wheel angle sensor

The steering wheel angle sensor is a critical part of the ECU system that measures the steering wheel position angle and rate of turn. A scan tool can be used to obtain this data in degrees. SWAS is located in a sensor cluster in the steering column. Cluster sometimes has more than one steering position sensor for redundancy to confirm data.

Steering wheel angle sensor is usually based on the magneto-resistance principle, that is, the property of a material changes its electrical resistance depending on applied magnetic field. When the steering wheel is turned, it normally rotates a gear and drives two other smaller gears which are provided with magnets. Magneto-resistors then record the angular position of the magnets. So measurement combination makes it possible to calculate the total rotation angle. The reason for multiple sensors is redundancy, accuracy and diagnostics. Multiple sensors are required since the steering angle is critical for the stability control system, and any discrepancy could mean the difference between making it around a corner or hitting another vehicle.

In addition, most sensors are packaged together in a single unit. When combined, the two signals can provide a more accurate reading on the positions of the wheels and how fast the position is changing. The two signal outputs are checked against each other to ensure accuracy. Usually, this sensor is mounted just behind the steering wheel on top of steering column and is a rotating sensor measuring the absolute steering angle.

As the steering wheel is turned, the SWAS produces a signal that usually toggles between 0 and 5 volts as the wheel is turned 360º. It is possible to observe the 0- to 5-volt signal with meters connected to the two SWAS sensors. Most vehicles produce a positive voltage turning right and a negative voltage turning left.

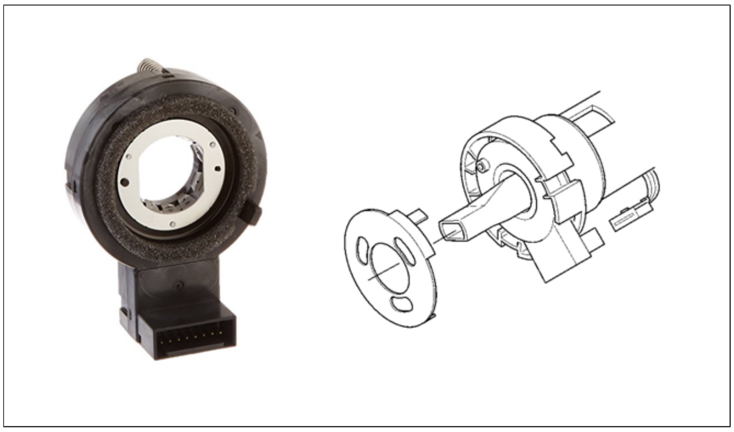

Figure 19. Steering wheel angle sensor

Source: General Motors steering wheel position sensor [31]

.

Precise and reliable detection of the steering angle is an important challenge in the modern vehicle. This data is used not only for Electronic Power Steering (EPS) and Electronic Stability Control (ESC), but it is also necessary for lane keeping assist and other advanced driver assistant systems. These are the first steps toward partial autonomous driving.

2.8 Control applications

Control application is very important to many processes based on a technology by which a process or procedure is performed without human assistance.

Fundamentally, there are two types of control loop; open loop control and closed loop feedback control. In the open loop control, control action is independent of the "process output" (or "controlled process variable"). On the other hand, for closed loop control, the control action is dependent on the process output. A closed loop controller therefore has a feedback loop which ensures the controller exerts a control action to give a process output the same as the "Reference input" or "set point". For this reason, closed loop controllers are also called feedback controllers.

2.8.1 PI Implementation

A proportional-integral controller (PI controller) is a family of controllers. They are the solution of choice when a controller is needed to close the loop and gives the designer a larger number of options. Those options mean that there are more possibilities for changing the dynamics of the system, in a way that helps the designer. A PI controller calculates an "error" value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimize the error by adjusting the process control inputs. It takes the in wheel encoder values as feedback and will check it again and again in closed loop to reduce the error [2].

PI controllers can be viewed as two terms - a proportional term which provides an overall control action proportional to the error signal through the all pass gain factor, and an integral term, reducing steady state errors through low frequency compensation by an integrator.

In electronic differential, PI control will be implemented as an input ED speed control to minimize errors and improve system stability and safety.

2.9 Controller Area Network

Controller Area Network (CAN bus) is a defined serial communications bus originally developed for the automotive industry to replace the complex wiring harness with a two-wire bus. The specification calls for high immunity to electrical interference and the ability to self-diagnose and repair data errors [42].

There are four types of CAN messages: Data Frame, Remote Frame, Error Frame and Overload Frame. The data frame is the standard CAN message, broadcasting data from the transmitter to the other nodes on the bus. A remote frame is broadcast by a transmitter to request data from a specific node. An error frame may be transmitted by any node that detects a bus error. Overload frames are used to introduce additional delay between data or remote frames [43]. Also, many short messages like temperature or RPM are broadcast to the entire network, which provides for data consistency in every node of the system.

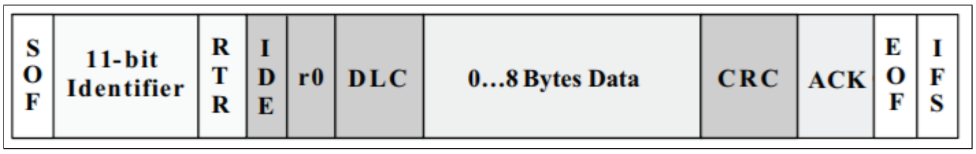

Figure 20. Standard CAN

Source: Introduction to the Controller Area Network (CAN) [42]

.

The meaning of the bit fields of Figure 20 are [42]:

SOF: this bit marks the start of a message, and is used to synchronize the nodes on a bus after being idle.

11-bit identifier: establishes the priority of the message. The lower the binary value, the higher its priority.

RTR: A bit that is dominant when information is required from another node. All nodes receive the request, but the identifier determines the specified node.

IDE: means that a standard CAN identifier with no extension is being transmitted.

r0: reserved bit.

DLC: contains the number of bytes of data being transmitted.

Data: bits of application data.

CRC: contains the checksum (number of bits transmitted) of the preceding application data for error detection.

EOF: marks the end of a CAN message.

A fundamental CAN characteristic shown in Figure 21 is the opposite logic state between the bus, the driver input and receiver output. Normally, a logic-high is associated with a one, and a logic-low is associated with a zero but not so a on CAN [42].

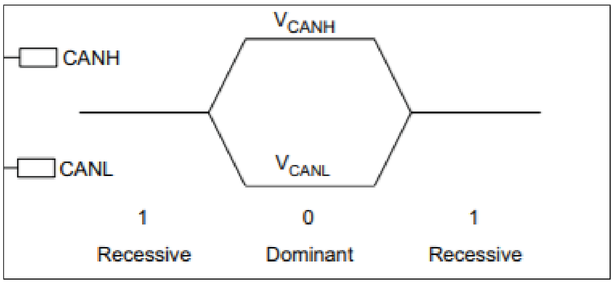

Figure 21. Standard CAN

3 Electronic Differential Development

The usual configuration of electrical or non-electrical vehicles involves only one traction motor driving two wheels, using differential gear. This thesis demonstrates the capacity of work of a rear multi-drive vehicle prototype , controlled independently by an electronic control unit. ED was created mainly to solve the turning issues for electric vehicles.

3.1 Drive-train proposed system

The drive-train proposed system permits the development of an electronic differential to assure that over a straight trajectory the two rear wheels rolls at same velocity and in curve trajectory, it assumes the difference between wheel velocities, satisfying the desired curved path.

Usually, a data communication channel is implemented between the central controller and each drive unit to command wheel speed and wheel torque. ECU controller is also interfaced to an accelerator pedal and a steering wheel input to provide information on drivers intent. In this project we developed a CAN (controlled area network) communication network, the most used in automobile industry.

For our drive-train system, ECU carries out the following basic tasks:

1. Reads the calibrated steering angle sensor and based on this it then calculates the steering wheel angle and also determines which direction the vehicle is moving.
2. Reads the throttle potentiometer voltage and estimates the accelerator pedal position.
3. Using the two readings above, calculates the speed reference applying fuzzy logic control.
4. Adopting a PI controller strategy and based on the above information, estimates the PI desired output speed and sends it to ED unit.
5. Calculates the ratio of the two speeds W_L/W_R using appropriate Ackermann-Jeantaud equations.
6. Applies a separate Pulse Width Modulated (PWM) signal is then applied to each of the two motors in accordance with the required speed ratio.

3.2 Vehicle Model application

According to project objectives, we need to develop a drive train system for Formula SAE e-Power UFRGS vehicle which has some specific characteristics and dimensions presented in table 3:

e-Power UFRGS Racing vehicle dimensions	
Dimension	Value (m)
Distance between front and rear wheel (L)	1.530
Distance between axis gauge (D)	1.300
Wheel radius (W)	0.531

Table 3. UFRGS electric vehicle dimensions

Applying the vehicle dimensions in Ackermann-Jeantand model to calculate each electric car angular wheel speed, we have the following results:

$$\omega_{Wheels} = \omega \cdot \left[\frac{L \pm (\frac{D}{2} \cdot tan(\theta))}{L} \right] \tag{17}$$

Left and right wheels angular speed

Where θ is the steering wheel angle, ω_{Right} and ω_{Left} are the right wheel angular speed and left wheel angular speed, respectively, according to the numerator sign.

Therefore, for electronic differential applications and vehicle dimensions we have:

$$\omega_{Left} = \omega \cdot \left[\frac{1.530 + (0.650 \cdot tan(\theta))}{1.530} \right] \tag{18}$$

Model application left wheel angular speed

$$\omega_{Right} = \omega \cdot \left[\frac{1.530 - (0.650 \cdot tan(\theta))}{1.530} \right] \tag{19}$$

Model application right wheel angular speed

3.2.1 Model simulation

In order to evaluate the performance of this proposed approach, Ackermann-Jeantand model was simulated in MATLAB to verify that the equations were consistent with car dimensions and the respective angular velocities of each wheel. For this, a constant vehicle displacement speed was set by 20 m/s, fluctuating the steering wheel angle in a -30 to 30 degrees range in order to match to FSAE UFRGS race car project.

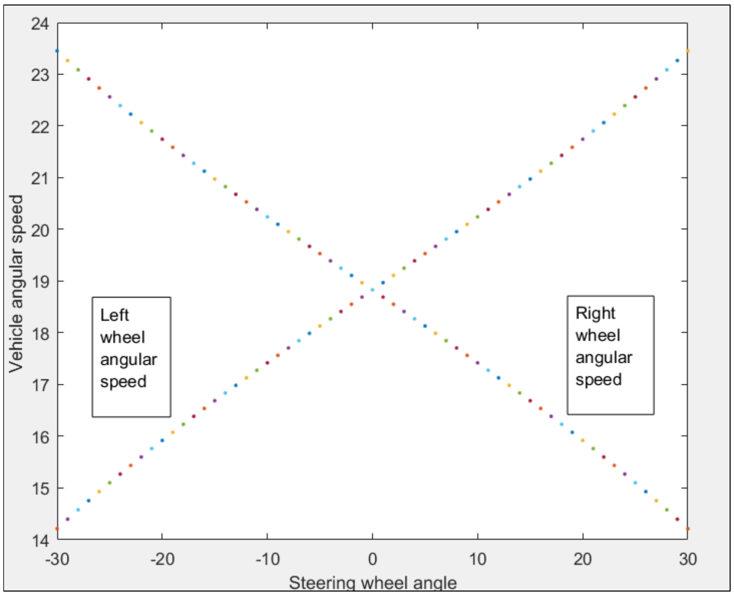

Figure 22. MATLAB wheels angular velocities x steering wheel angle simulation

As expected, the angular velocities of right and left wheels for angles possess the same values but opposite sign, having identical magnitudes as presented in table 4.

SWA (°)	Left Wheel Speed (rad/s)	Right Wheel Speed (rad/s)
-25	15.101	22.563
+25	22.563	15.101
-15	16.688	20.976
+15	20.976	16.688

Table 4. Steering wheel angle vs. Angular wheel speed

3.3 System Overview

The proposed block diagram for electric vehicle drive system delivers two PWM signals to control in-wheel motor rear wheel drives, as illustrated in figure 23.

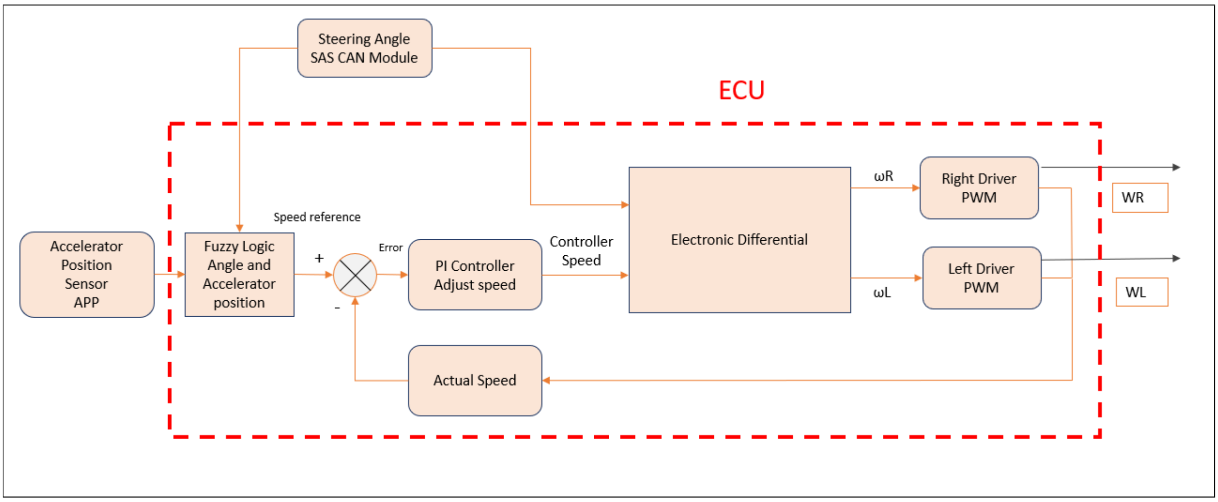

Figure 23. System blocks diagram overview

This system is responsible for generating the desired output speed for each motor drive. A fuzzy controller is applied in order to manage the accelerator speed reference through the accelerator pedal position and steering wheel angle. With that, proportional-integral (PI) controller produces an output speed for electronic differential based on accelerator speed reference and current vehicle speed. Furthermore, ED system balances the differential speed between wheels and thus, the microcontroller sets two PWM signals that correspond to right and left wheels output speed.

The wheels' speeds are the same when steering wheel angle is 0 degrees, so vehicle is traveling in a straight path. Therefore, turning or not, left and right speed average is the prototype current velocity.

3.4 Electronic control unit

Electronic control unit (ECU) controls many electrical system or subsystems in a vehicle. A microcontroller, designed for embedded applications, is in charge of retrieving sensor data, processing information, generating the requested signals for our vehicle. It is comprised of a fuzzy controller, PI controller, electronic differential system (ED system) and a communication system (CAN module).

Selected device microcontroller was a Microchip - ATMEGA328 due to its capacity of processing, its ability to change parameters with ease and its intuitive programming language. It has many digital and analog inputs/outputs which allows further project evolution as other vehicle sensors are attached to CAN bus. Figure 24 displays all available pins on the controller development board.

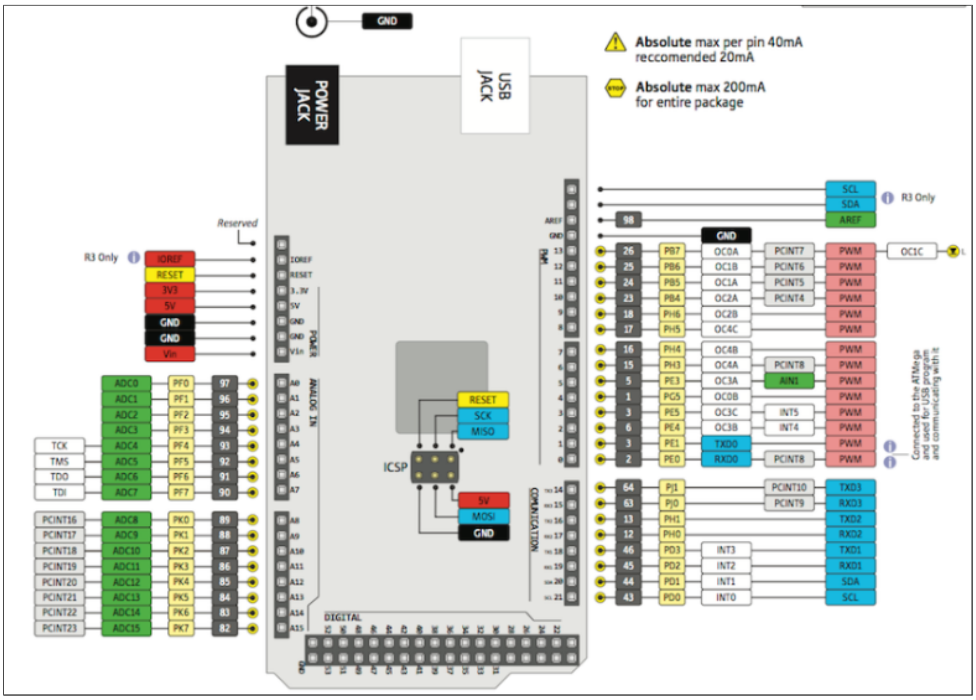

Figure 24. Controller board pinout

For the electronic differential system, its sensors and communication, the following pinout was used to connect the analog and digital outputs, in order to offer a better wire arrangement. The pinout diagram is presented in table 5.

Pinout	Port	Function
A1	Analog	APP sensor signal
GND1	Analog	Board and drivers ground
D8	Digital	Driver module IN1
D7	Digital	Driver module IN2
D6	Digital	APP Fuzzy speed reference
D45	Digital	PWM left motor driver
D46	Digital	PWM right motor driver

Table 5. Board pinout

Pinout	Port	Function
GND2	Analog	CAN module ground
PWR1	Analog	CAN module power supply
D2	Digital	CAN module interrupt
D53	Digital	CAN module CS
D52	Digital	CAN module SCK
D51	Digital	CAN module SI
D50	Digital	CAN module SO

Table 6. Communication board - module pinout

3.4.1 Software

Software that controls the electronic central unit was designed to be user friendly, simplifying the development and changing of parameters, electronic differential tuning and accelerator pedal fuzzy control according to pilot or team preferences.

Software programming is divided in three major blocks: variable and pins definitions, system setup and main functioning. The first one is comprised of some variables, necessary for electronic differential and communication settings while the second block consists of setup and variable functions.

The main block is a loop function, that is to say, it works continually when the device is working, communicating with the accelerator pedal position sensor and steering wheel angle sensor, generating and managing the vehicle functions and electric machine drivers.

Each motor driver requires three signals to work properly: PWM that controls the motor speed based on pulse width, and two digital port signals to manage EM rotation direction.

3.4.2 PI Controller

A PI controller was implemented, in order to do speed adjustments for the electronic differential. Since this type of controller works according to its pre-set parameters, they have been set up to increase the system efficiency.

This tuning is done by applying the difference between current car speed and the speed reference, provided by fuzzy controller, combining the accelerator pedal position and steering wheel angle. Electronic differential PI-controlled system is less responsive to real (non-noise) and relatively fast alterations in state, considering P and I parameters.

One of the techniques for determining a PI controller's parameters is based on the closed loop system characteristics. This analyzes the system in frequency domain. The controller is connected to the process (ED) with integral parameter set as zero and proportional gain is increased until the system starts to oscillate. After it, when the P parameter is optimal, I parameter is increased until the output of the control loop has stable and consistent oscillations, to eliminate steady-state error. This technique was used to regulate controller parameters, set in order to manage the input speed of the electronic differential.

PI parameters, found through tests described above, are proportional P = 1.25 and integer I = 1.0. Figure 25 shows the PI controller blocks diagram and its parameters.

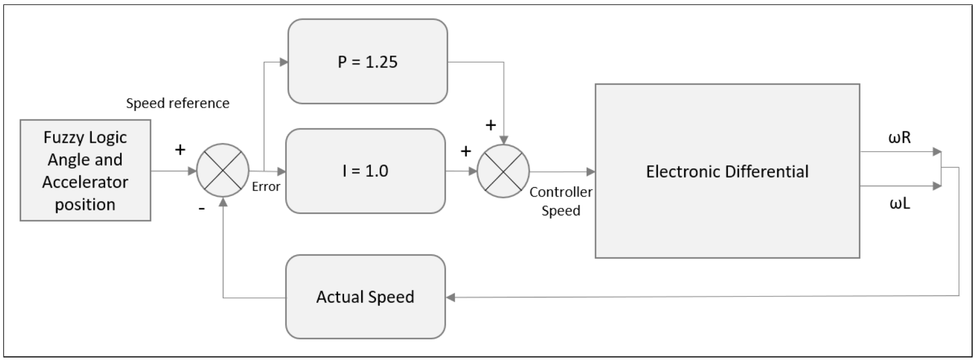

Figure 25. PI controller and parameters

3.4.3 In-System Communication Protocol

A vehicle communication system is very important to allow devices to share data, facilitating communication with each other, without causing any overload to ECU. The in-system communication protocol is the controller area network (CAN), commonly used in the automobile industry due to high reliability, long distance range and average communication speed.

This network was implemented in order to bring flexibility when we add or exclude car sensors. That brings many advantages since the controller area network also has low signal attenuation.

For this electric system, CAN module selected was MCP2515, it has two acceptance masks and six acceptance filters that are used to filter out unwanted messages and interfaces with microcontroller via an industry standard Serial Peripheral Interface (SPI).

The module handles all functions for receiving and transmitting messages to CAN bus, by first loading the appropriate message buffer and control registers, while transmission is started using control bits via SPI interface.

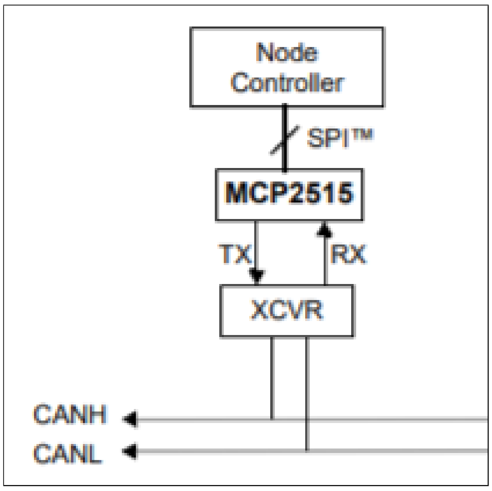

Figure 26. CAN system implementation

ECU communicates with CAN module via SPI interface logic which presents four wires: CS, SCK, SI and SO. CS has the chip selected input pin for SPI interface, SCK is the clock output for SPI while SI is the data input pin and SO, the data output pin. Also, ECU needs an interrupt (INT) output to share information with module, as established in MCU software programming. Prototype MCP2515 CAN module and its connections are presented in figure 27.

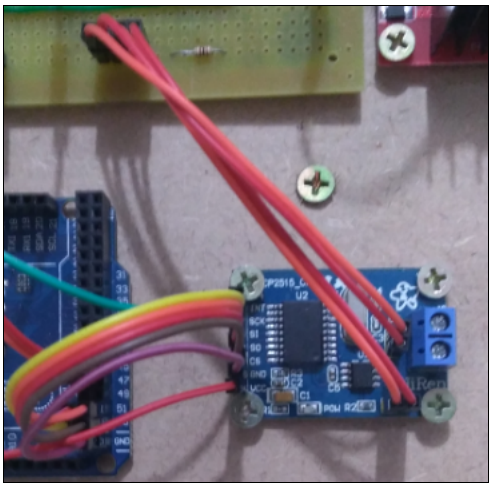

Figure 27. Prototype MCP2515 CAN module

3.5 Fuzzy accelerator application method

In order to slightly cause an electronic stability control effect, fuzzy method was applied to control the accelerator pedal reference speed for electronic differential. Typically, the electric machine's speed control is directly proportional to the accelerator pedal response.

With the fuzzy control, we put the accelerator pedal sensor and steering wheel angle at the steering column to fit: when the steering wheel angle is high, fuzzy control causes a slight decrease in the reference speed in order to reduce the chance of sliding on the track as well as provide a softer driving. Fuzzy accelerator block diagram is exemplified in the figure 28.

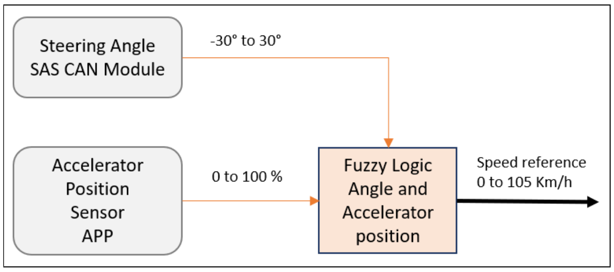

Figure 28. Fuzzy accelerator blocks diagram

3.5.1 Fuzzy accelerator rules

The fuzzy rules associate ideas and relate one event to another. Fuzzy logic control rules have been defined with the objective of leaving an accelerator reference speed more proportional to accelerator pedal position when applied small angles to steering wheel, while for angles more distant from zero, the controller implements a slight decrease into output reference speed.

For that, seven rules were created for accelerator pedal position and five rules for steering wheel torsion position. The control requires a combination of all the possibilities of these rules, thus totaling 35 rules in order to achieve this goal.

Considering accelerator pedal position input, the membership functions are: Zero, VeryLow, Low, Medium, High, VeryHigh and MAX. They are presented in figure 29.

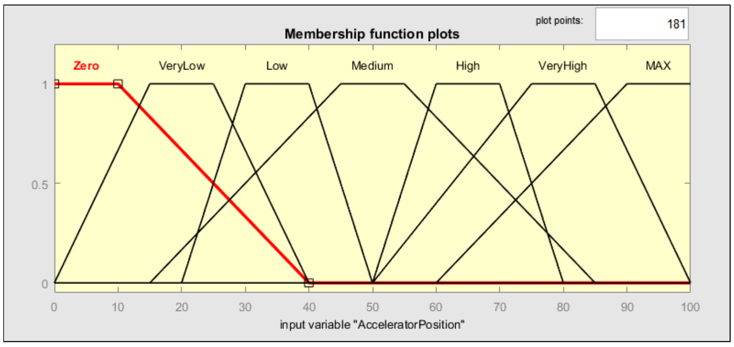

Figure 29. Input membership function of APP

For the other input rules, according to steering wheel turning angle, the following are considered: HighLeft, LowLeft, Zero, LowRight and HighRight. These membership functions are displayed in figure 30.

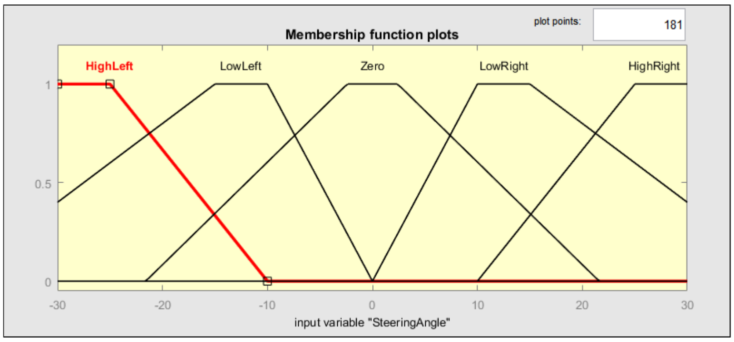

Figure 30. Input membership function of SWA

Taking into account that output speed reference is an input variable to electronic differential, also seven rules are created: Zero, VeryLow, Low, Medium, High, VeryHigh and MAX.

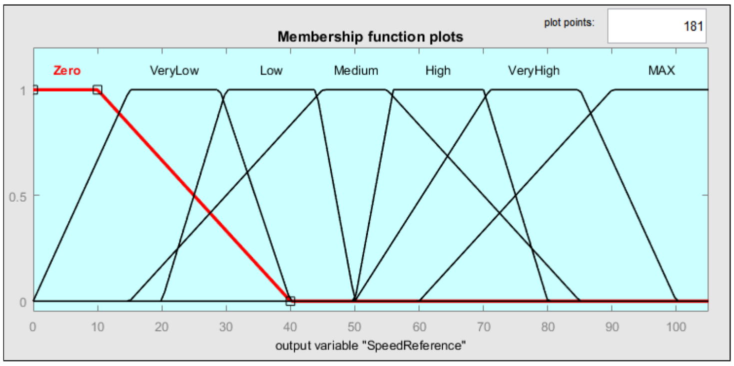

Figure 31. Output membership function of speed reference

3.5.2 Accelerator fuzzy-controlled

Fuzzy control was set to work between steering wheel angles from -30 to 30 degrees (steering wheel angle after steering column transmission) and the accelerator pedal position was defined at 0 to 100 percent. In this way, it will produce a reference speed that will vary from 0 to 105 km / h. Accelerator pedal position x speed reference surface are represented in figure 32.

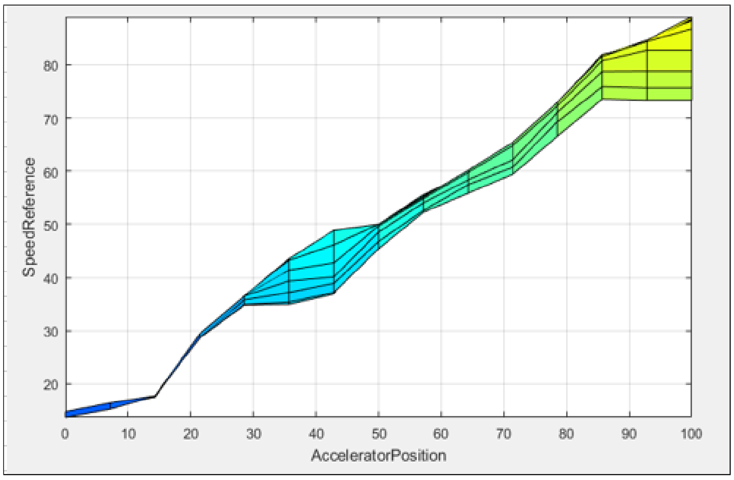

Figure 32. Fuzzy accelerator: Accelerator position x speed reference

Verifying the fuzzy logic control, a simulation was carried out. All accelerator pedal positions and all steering wheel steering angles were implemented in the test. Simulations results are shown in figure 33.

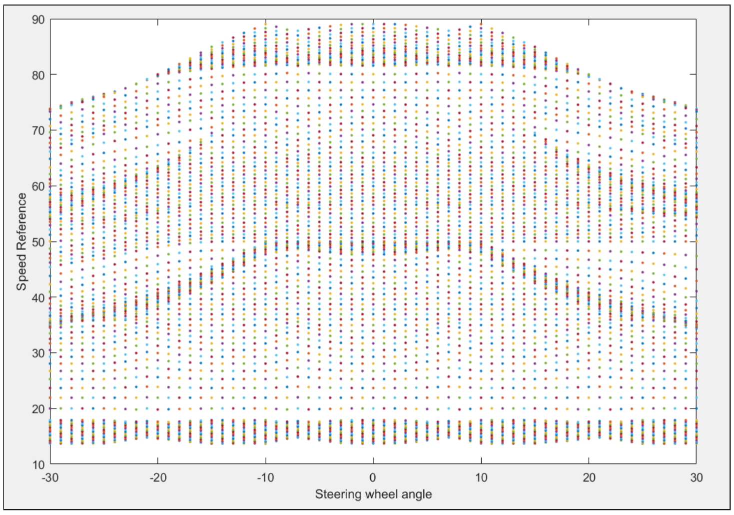

Figure 33. Fuzzy accelerator: Speed reference x Steering wheel angle

3.5.3 Fuzzy accelerator applied surface

After applying the rules and defining parameters, a multi-input single-output (MISO) system generates a surface where it is possible to verify system behavior for every association, between steering wheel turning and accelerator pedal position. Figure 34 shows the surface, where yellow and blue, respectively, represent the highest and lowest velocities.

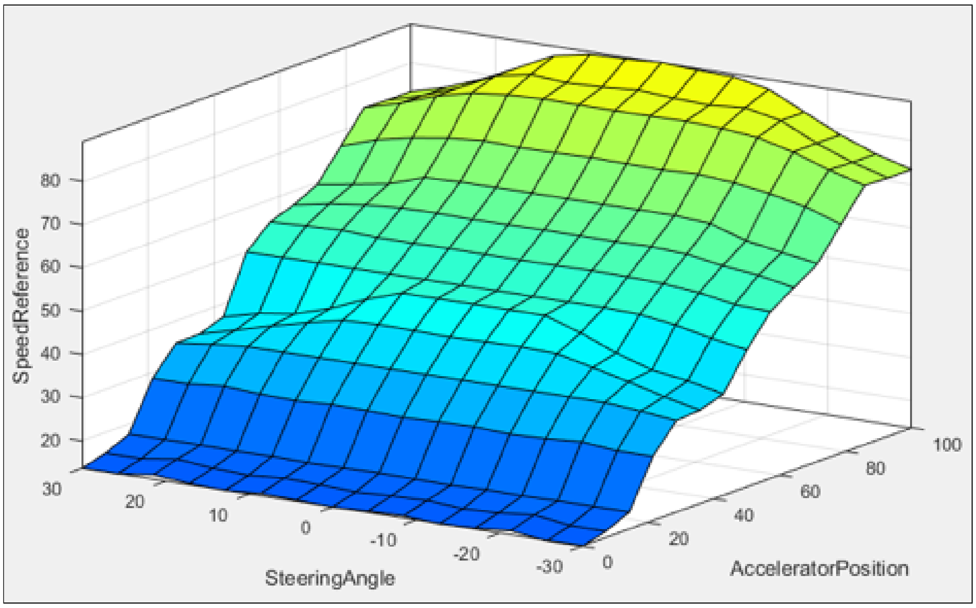

Figure 34. Fuzzy accelerator surface simulation

3.5.4 Accelerator conditioning circuit

For proper control of the electronic differential system, a conditioning circuit was implemented for accelerator pedal position sensor, thus the system was left more reliable and with less noise. APP sensor is controlled by three connections: 0-5V direct current supply, output signal for microcontroller and ground. If the output signal is noisy, the sensor behaves unsteadily, making it impossible to use for precise operations. In addition, noise is reduced, implementing capacitors between signals. Also, for power supply, a voltage regulator 7805, shown in figure 35, was used to guarantee a stable 5V voltage.

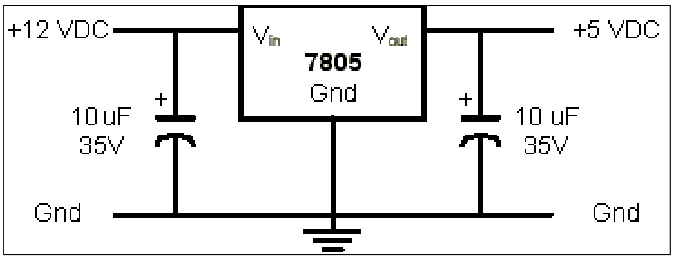

Figure 35. Voltage regulator used in APP sensor power supply

Source: 7805 Regulator [46]

.

3.6 SWA sensor

Steering wheel angle sensor is responsible for giving a feedback about steering column position. The vehicle drive has been designed with the interest of turning the steering wheel about -130 to 130 degrees. The equations were calculated in order to apply the angular wheel position, so, steering column has a transmission ratio software-programmed.

Sensor is positioned directly below steering wheel, connected to a steering column, in charge of approximating the prototype drive to a real vehicle part, rotating ± 150 degrees. SWA sensor communicates with ECU via CAN module and it has 0.1° resolution. The device sends a CAN message with the measurement data every 10 ms.

SWAS has four connections: 10 volts power supply, ground, CAN L and CAN H. These are the communications with ECU and module and the calibration for zero position is adjustable through CAN command.

3.7 Data acquisition method

To obtain the data of an electronic differential system it is necessary to have a sensors and calibration devices, aligned to filters that attenuate the noise present in the system, besides an experimental methodology that brings more reliable results.

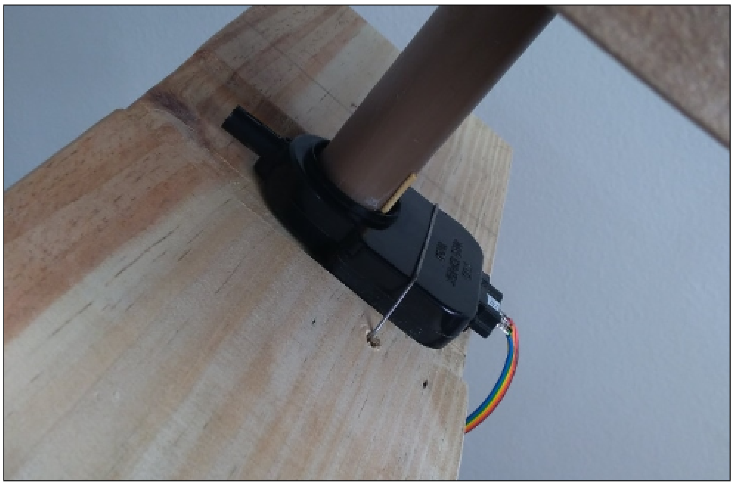

Figure 36. SWA sensor positioning

3.7.1 Calibration

Calibration is the comparison of measurement values delivered by a device under testing with those of a calibration standard of known accuracy. For system validation tests, electronic differential prototype must assume that the DC electric machines are rotating at the same speed when the same acceleration is applied to both motors and steering wheel angle is equal to zero, that is to say, the vehicle is going in a straight path.

For the angular velocities speed calibration a stroboscopic light was used (producing regular flashes of light to stop the appearance of rotating motion) allowing to measure the rotational speed.

First, to calibrate left and right electric motors rotational speed, steering wheel angle was set to zero degrees and a minimum acceleration was applied on the pedal so as to let the motors spin. This point was considered zero acceleration, then three measurements of the rotational speed of each machine were completed. According to results, adjustments in the PWM pulse that controls the drivers are made and the motor speeds were set the same. Second calibration was applying maximum acceleration in the APP and also putting the steering angle equal to zero. Thus, we have the maximum speed of each motor. Then, they were calibrated so as to set the same maximum speed of rotation in both electric machines. In this way, we ensure that the maximum and minimum speeds are the same and they are calibrated for electronic differential tests. Stroboscopic light prototype calibration is presented in figure 37.

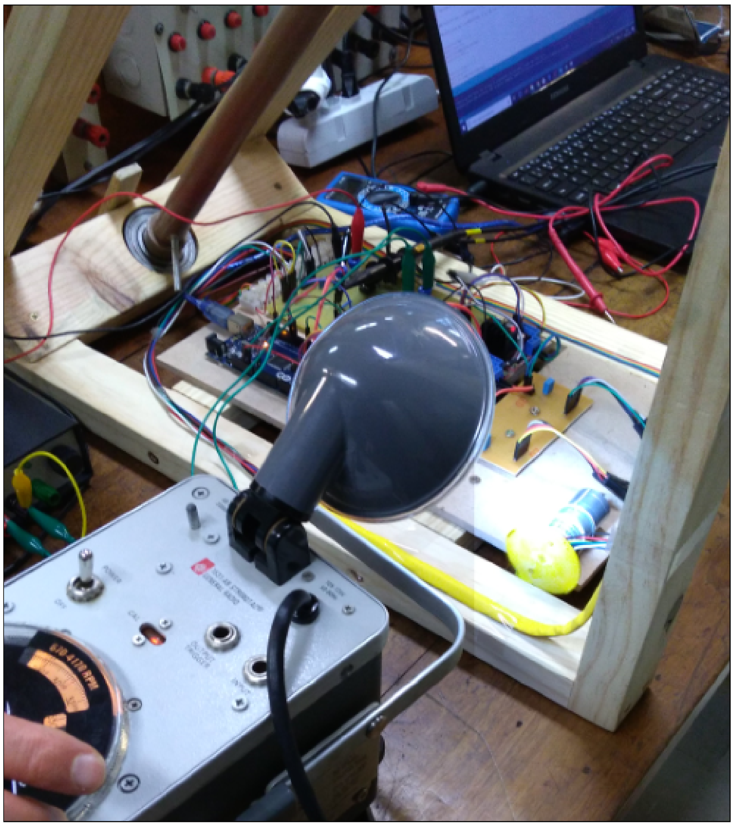

Figure 37. Stroboscopic light calibration

3.7.2 Procedure

Tests of Fuzzy accelerator, rotational speed of left and right electric motor, steering wheel angle and validation of the electronic differential system were performed. For every data acquisition, three repetitions were done to reduce measurement errors and the result of this arithmetic mean was used. Arithmetic mean formula is presented below:

$$\overline{x} = \frac{x_1 + x_2 + x_3}{3} \tag{20}$$

With the obtained results, graphs were generated applying the arithmetic values in order to compare with previously performed simulations, using they for the electronic differential prototype system validation.

3.7.3 Noise reduction

Several devices, both analog and digital, have traits that make them susceptible to noise. There are many techniques to reduce noise for data acquisition, such as first order filters and shielded cables. The prototype uses an auxiliary board in order to reduce noise with conditioning circuits.

Auxiliary board has three buffers combined with capacitors treating PWM left and right microcontroller signals and accelerator pedal position sensor signal. The board was built with the intention to make the system more stable, attenuate noise and make connections easier to identify. Figure 38 provides the schematic circuit of the auxiliary board.

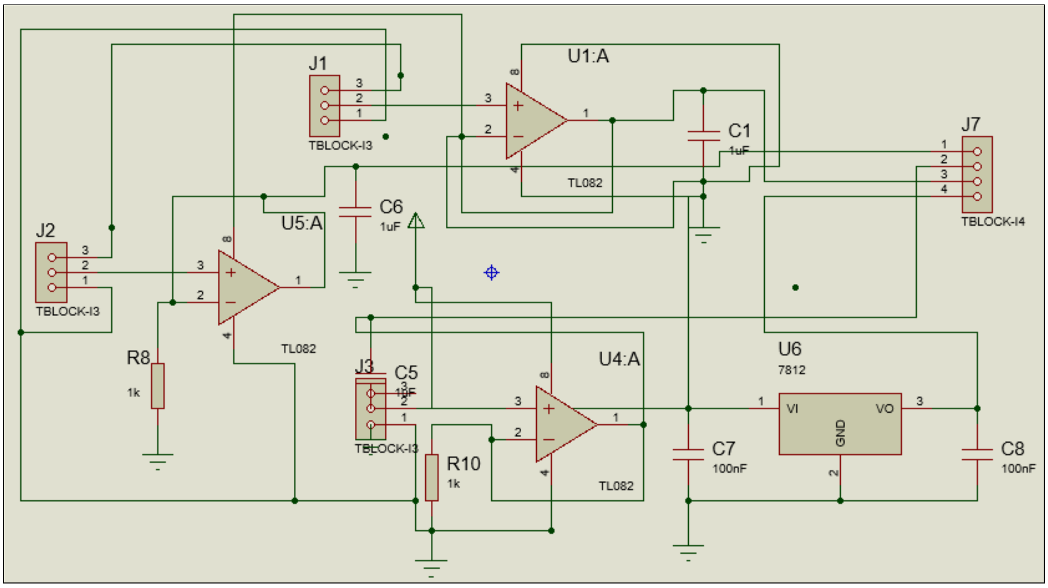

Figure 38. Noise reduction schematic circuit

3.7.4 Vehicle drive-train prototype

To validate the electronic differential system, a prototype was assembled. It contains a built in wooden structure to fix the sensors and make them easy to operate. The structure has a base, which has the electric machines and electronic circuits. Located above it is a steering column, attached to the General Motors (GM) Trailblazer steering wheel sensor C68049XF, which is then connected to a steering wheel. The accelerator pedal, which contains the potentiometer position sensor, 6QE721502, is the same as that used in the Volkswagen (VW) Gol G5.

Two 6V DC electric machines were used for their low cost and similar operation, from ECU point of view, to motors. To actuate the EM, two drivers were applied which allows the direction of rotation but also of its speed, using the two PWM left and right signals generated by the electronic control unit. Thus, prototype is composed of: DC motors, two H bridge drivers, microcontroller, CAN module, auxiliary board and connections. Figure 39 shows the device blocks diagram overview.

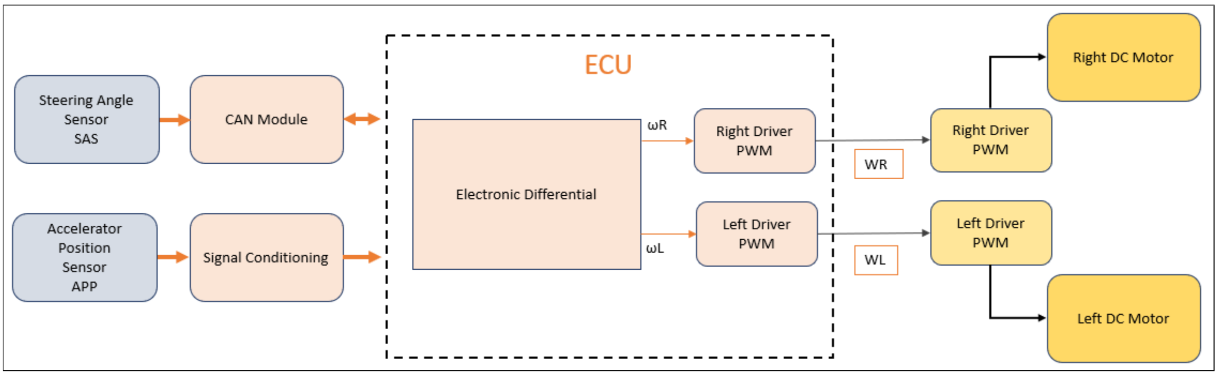

Figure 39. Device blocks diagram overview

A prototype overview, containing sensors, ECU, electric machines and wooden structure is presented in figure 40.

Figure 40. Prototype overview

Prototype frontal view in figure 41 exposes:

1. Accelerator pedal position sensor
2. Steering wheel angle sensor
3. Electronic control unit
4. Auxiliary board and drivers
5. DC electric machines

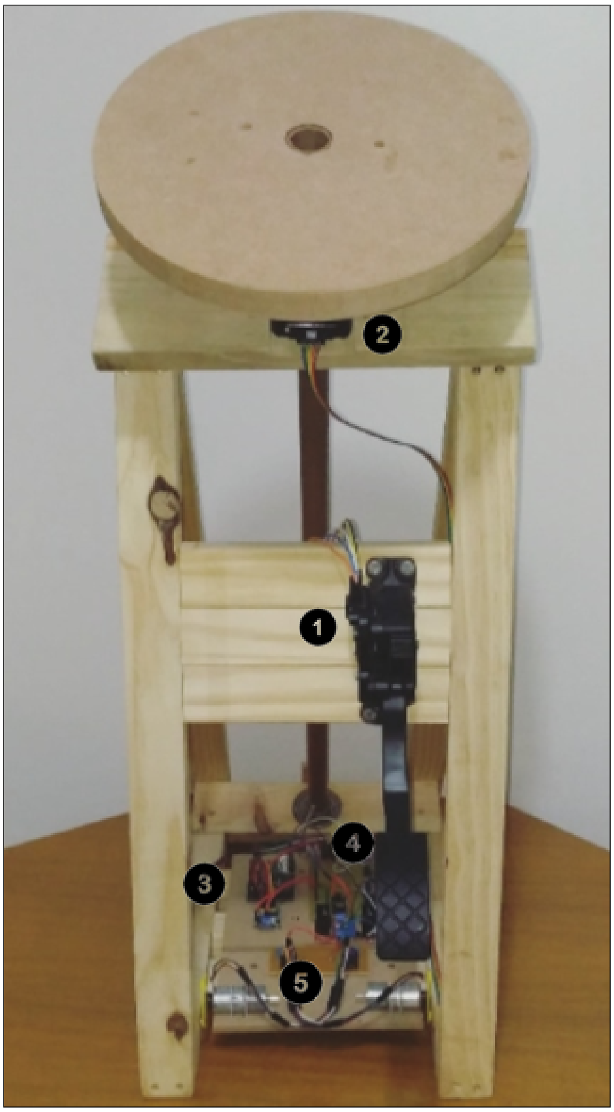

Figure 41. Prototype frontal view

System top view in figure 42 shows:

1. Left DC electric machine
2. Right DC electric machine
3. Electronic control unit
4. Auxiliary conditioning board
5. Left PWM bridge driver
6. Right PWM bridge driver
7. Power supply

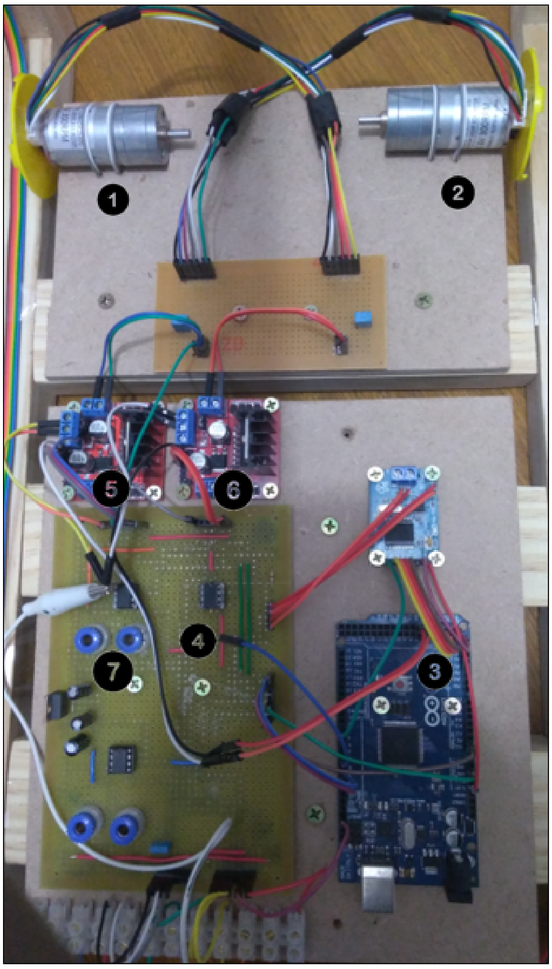

Figure 42. Prototype frontal view

A prototype electrical circuit overview, containing sensors, ECU, electric machines, voltage regulators, buffers and filters is presented in figure 43.

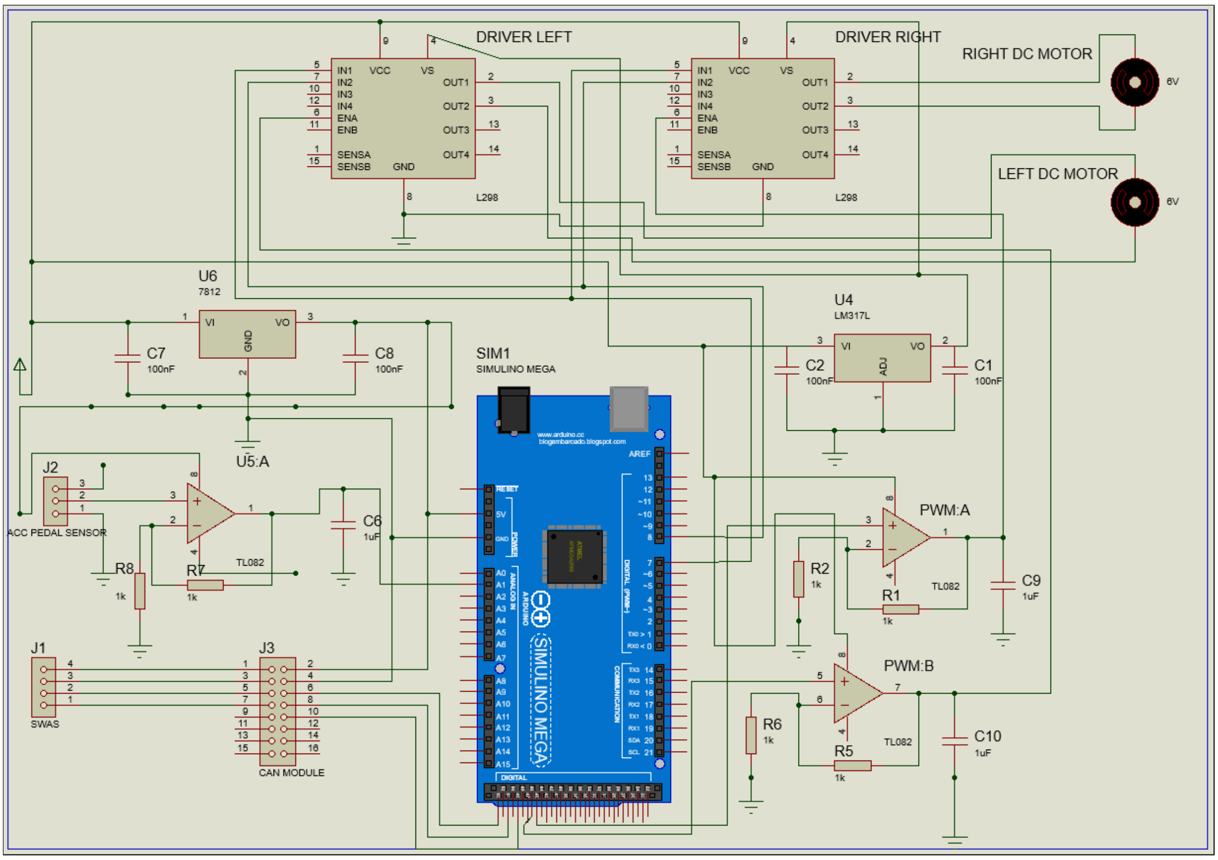

Figure 43. Prototype electrical circuit overview

4 Results

This chapter will present the experimental method results proposed in this work. It will initially discuss the obtained results through the fuzzy accelerator control method, as well as the PI controller input reference speed, comparing the collected experimental data with software simulations. Then, the control of the electronic differential will be analyzed according to vehicle current speed and the reference speed provided by the fuzzy accelerator. Finally, the results obtained in the electronic differential prototype will be presented and compared with the results seen in the simulations.

4.1 Fuzzy-controlled accelerator

Considering experimental tests, firstly, fuzzy accelerator control system was characterized measuring the output of accelerator position pedal sensor for different positions. Thereby, the top and bottom sensor values has been defined as: 0 to 100% acceleration that represents a minimum value of 0.78 V and a maximum of 4.47 V. The sensor response curve, demonstrated a linear variation between the maximum and minimum points, as expected for a linear potentiometer sensor. Figure 44 presents the accelerator pedal position sensor response for position variations.

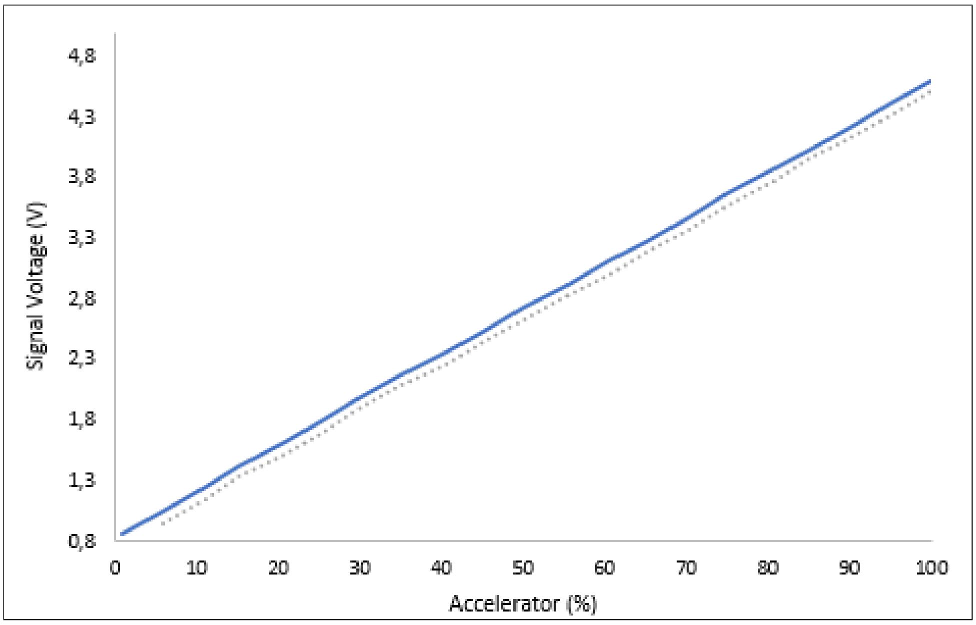

Figure 44. Accelerator pedal position sensor output

Later, the fuzzy control system tests were carried out, based on the accelerator sensor input and steering wheel angle: SWA was set at 0 degrees and several accelerations were applied at 5 percent intervals, to check the control system response exclusively in relation to electronic accelerator signal.

When a small angle is set, the fuzzy accelerator offers a more direct response, because the probability of losing vehicle control is lower, thus, stability control fuzzy system presents a low impact. Considering the output fuzzy accelerator speed reference acceleration values represent a minimum of 0.19V or 0 Km/h and also a maximum value of 1.87V or 104.9 Km/h.

The obtained results are very similar to those presented in the device simulations, corroborating for the validation of fuzzy accelerator control system. Figure 45 displays the fuzzy accelerator speed reference results, when steering wheel angle was set at zero degrees.

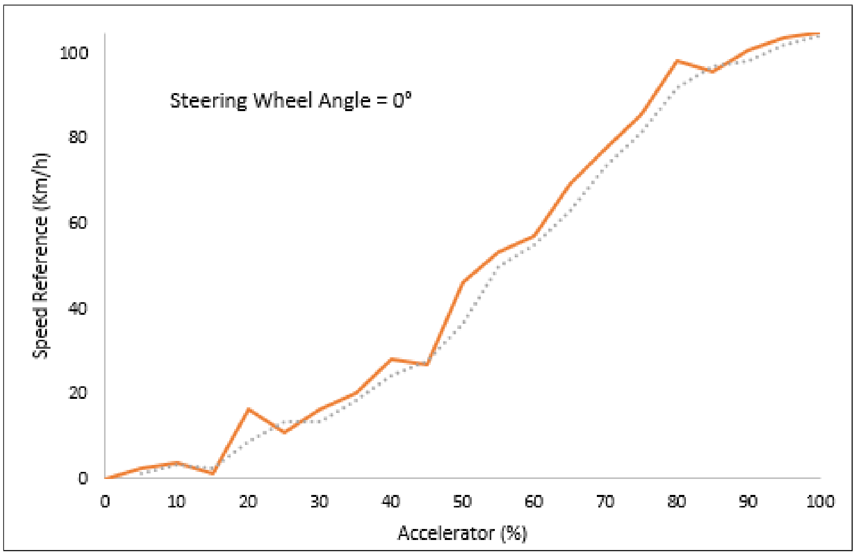

Figure 45. Fuzzy accelerator output: SWA 0°

In the second experimental test, steering angle was set at +25 degrees and accelerations were applied starting at the minimum value, increasing gradually by 5 percent. For a larger angle, according to the simulations performed, the fuzzy control stability acts on the system in order to decrease the reference speed.

According to the output fuzzy accelerator speed reference, in this case, values represent a minimum of 0.19 V or 0 Km/h and also a maximum value of 1.70V or 94.37 Km/h. Figure 46 displays the fuzzy accelerator speed reference results for a +25°SWA.

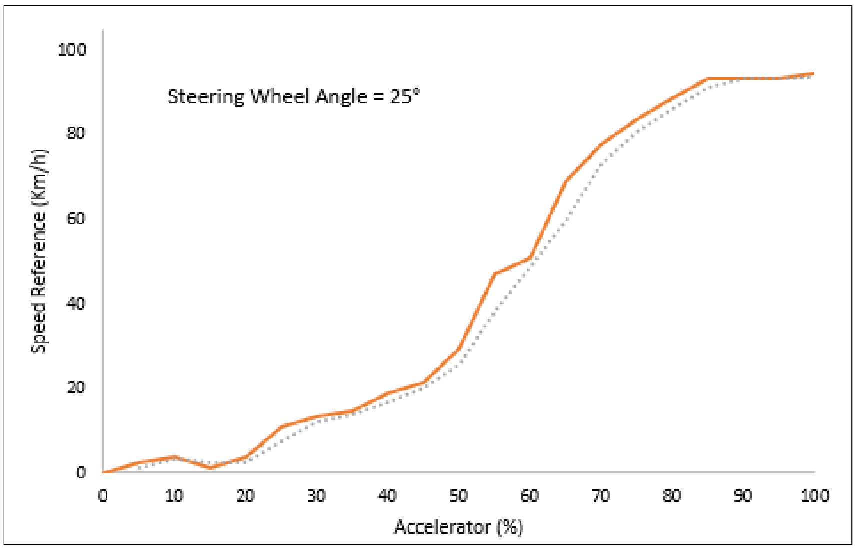

Figure 46. Fuzzy accelerator output: SWA 25°

4.2 Speed Controller

PI controller output speed is responsible for electronic differential speed input. According to the simulation, when the steering angle is set at zero degrees, this velocity should act in a more linear way, as can be seen in the figure 47.

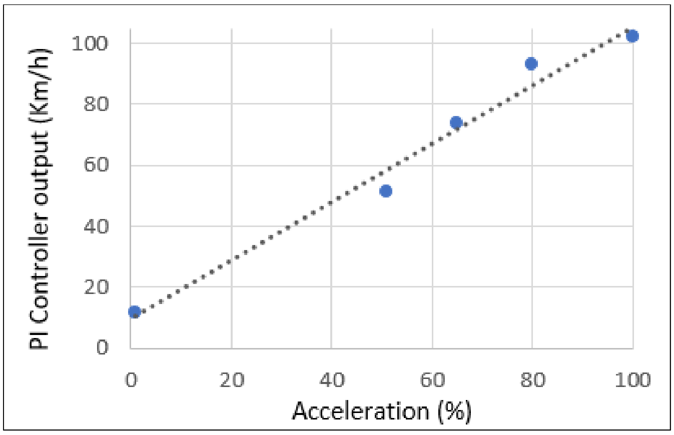

Figure 47. PI Controller output speed

4.2.1 Pulse Width Modulation

The management of the prototype's speed control is done by two PWM signals generated by the electronic control unit (ECU). Drivers that control engine speeds have three inputs, without considering power supply. Two pins to indicate the motor rotational direction and a PWM pin that controls the speed via pulse width. In this case, when the right and left pulse widths are the same, the motors should theoretically rotate at the same angular velocity. Typically, because of internal factors such as parts assembly and power supply, the motors are not identical, needing to undergo a calibration to leave them at the same angular velocity via software adjustments, when the steering wheel angle is set at zero degrees.

Firstly, a constant pressure was applied on the accelerator pedal, so that the acceleration is equal to 65 percent. Then, the steering wheel was placed at +2 degrees on prototype. In this case, the signals have almost the same pulse width, that is, the electric machines rotates at a very similar angular speed. The figure 48 shows the pulses width that are generated by the electronic control unit. Yellow and blue signals represent the pulses for the left and right motors' control, respectively.

Figure 48. Left and right PWM: +2° steering wheel angle

A second pulse width test was accomplished: a steering wheel angle of -22.9 degrees was applied, that is to say, vehicle was turning to right. Same acceleration of 65 percent was maintained to compare with the previous case. Observing the test, it is noticed that the pulse sent to the left driver has a width greater than the right one , that is, left electric machine rotates at an angular speed greater than the right EM, which appears to be consistent with the carried out simulations. The figure 49 represents the left and right pulses.

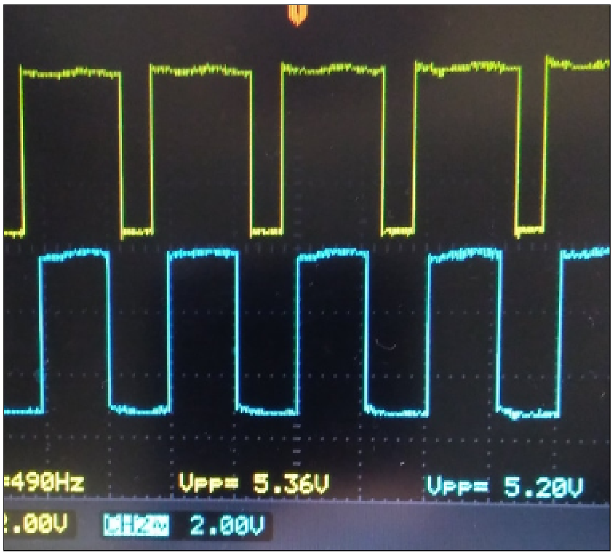

Figure 49. Left and right PWM: -22.9° steering wheel angle

In the final test for this check, the position of accelerator pedal is kept at 65 percent and a steering wheel angle of +10 degrees is applied. In this way, the prototype would be turning to the left, that is, the left PWM has a smaller pulse width than the right one. This situation is confirmed in figure 50.

Figure 50. Left and right PWM: +10.1° steering wheel angle

4.2.2 Electronic differential effect

In order to validate the electronic differential system and confirm the differential effect applied to it in-wheels electric machines, tests were executed to analyze the speed of rotation of each EM, for different situations. The analyses performed after the device calibration and some circumstances may have influenced the measurement performed such as: stroboscopic light position, precision and parallax error.

A constant acceleration equal to 65 percent of the APP was employed to compare the rotational speeds of the two electric machines, modifying the corresponding steering wheel angle by a range of -5.2 to 5.2 degrees, divided by approximately 1.1 degrees. Figure 51 and table 7 present the results obtained in the test.

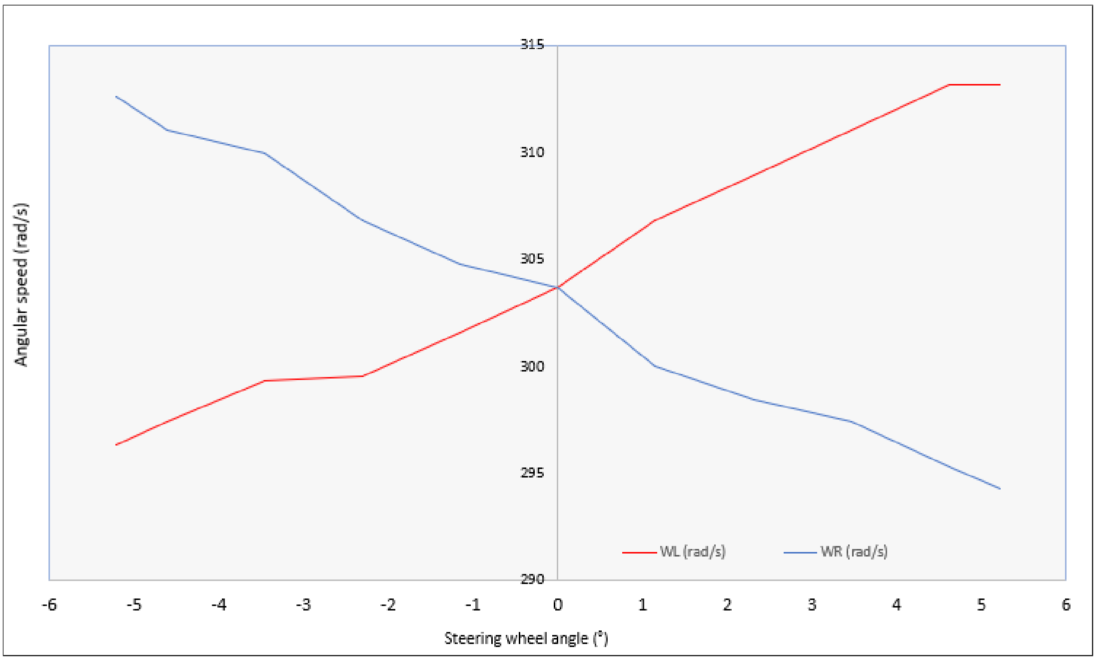

Figure 51. Experimental Left and right wheels angular speed

The results corroborate with the simulation done for the same case, showing that when the steering wheel angle is zero, the two EM rotational speeds are the same. In addition, for complementary angles, that is, with the same magnitude but opposite signals, the results are very close when the speeds of the two wheels are evaluated.

SWA (°)	WL Speed (rad/s)	WR Speed (rad/s)
-5.2	296.4	312.6
-4.6	297.4	311.0
-3.4	299.2	309.9
-2.3	299.5	306.8
-1.1	301.6	304.7
0.0	303.7	303.7
+1.1	306.8	300.0
+2.3	308.9	298.5
+3.4	311.1	297.4
+4.6	313.1	295.3
+5.2	313.1	294.3

Table 7. Experimental steering Wheel angle vs. Angular wheel speed

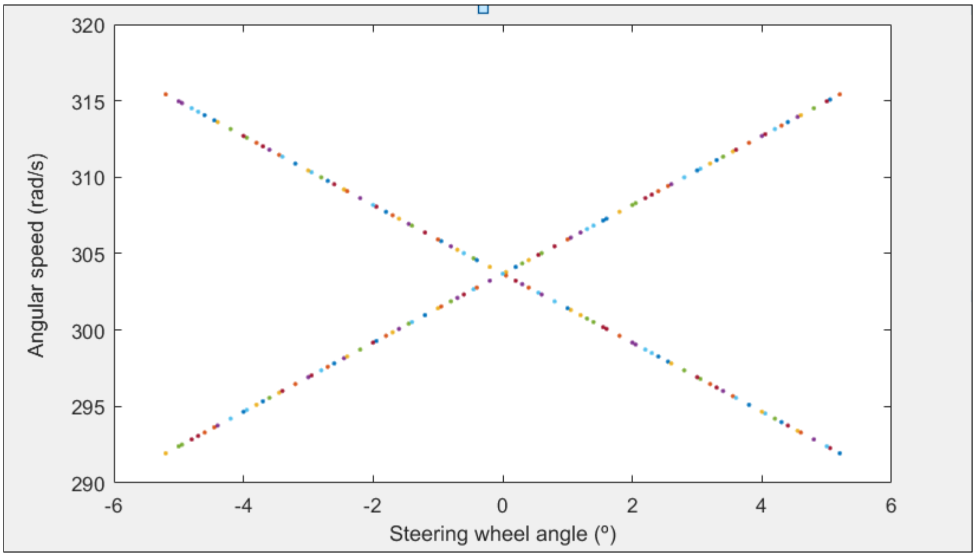

Figure 52. Left and right wheels angular speed simulation

To PWM signal average voltage estimation, a constant acceleration equal to 30 percent of the APP was employed to compare the rotational speeds of the two electric machines, modifying the corresponding steering wheel angle by a range of -20.0 to 20.0 degrees. It shows that the output average voltage has a consistent variation of wheels speed ratio, considering the steering wheel angle and the electronic differential effect. Figure 53 and table 8 presents the left (yellow line) and right (green line) PWM output average voltage.

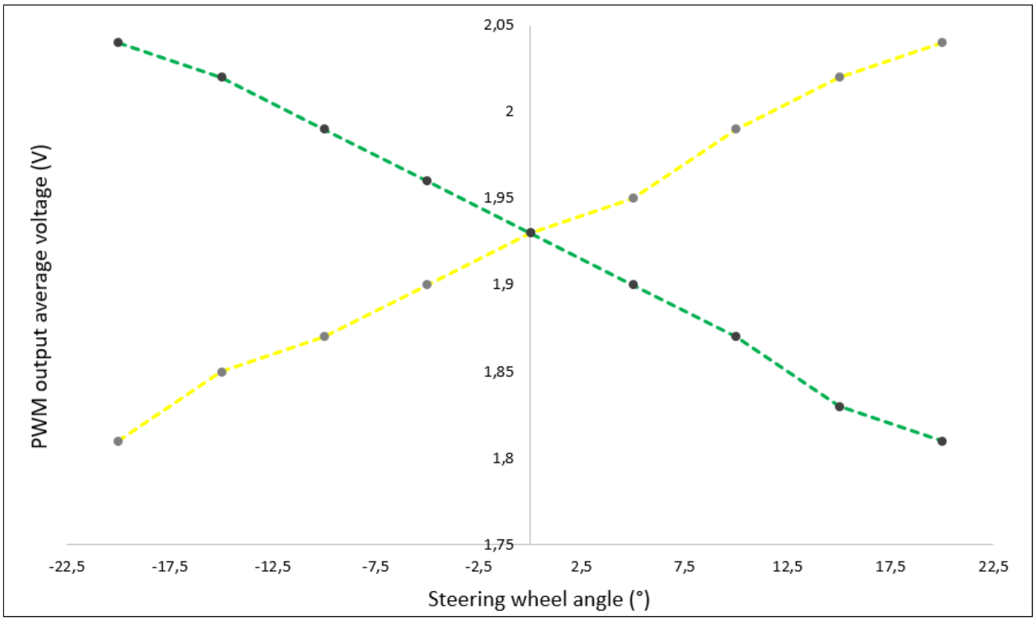

Figure 53. Left and right PWM output average voltage

SWA (°)	PWML Voltage (V)	PWMR Voltage (V)
-20.0	1.81	2.04
-15.0	1.85	2.01
-10.0	1.87	1.99
-5.0	1.91	1.96
0.0	1.93	1.93
+5.0	1.96	1.90
+10.0	1.98	1.87
+15.0	2.01	1.84
+20.0	2.04	1.81

Table 8. Left and right PWM output average voltage

To better illustrate the velocity variations, the figure 54 demonstrates different cases of angular velocities. At first, with the angle set at zero degrees, the electric machines are accelerated until reaching a speed of 303.7 rad/s. Then, in the first case, turn the steering wheel to the right by changing the angular velocities so that the angular velocity of the left engine is greater than the right one. After, similar tests are performed, always with the steering wheel returning to the origin, demonstrating the differential effect of angular velocities. Considering the obtained data, we can confirm that the system is compatible with the Ackermann-Jeantand equations for vehicle modeling.

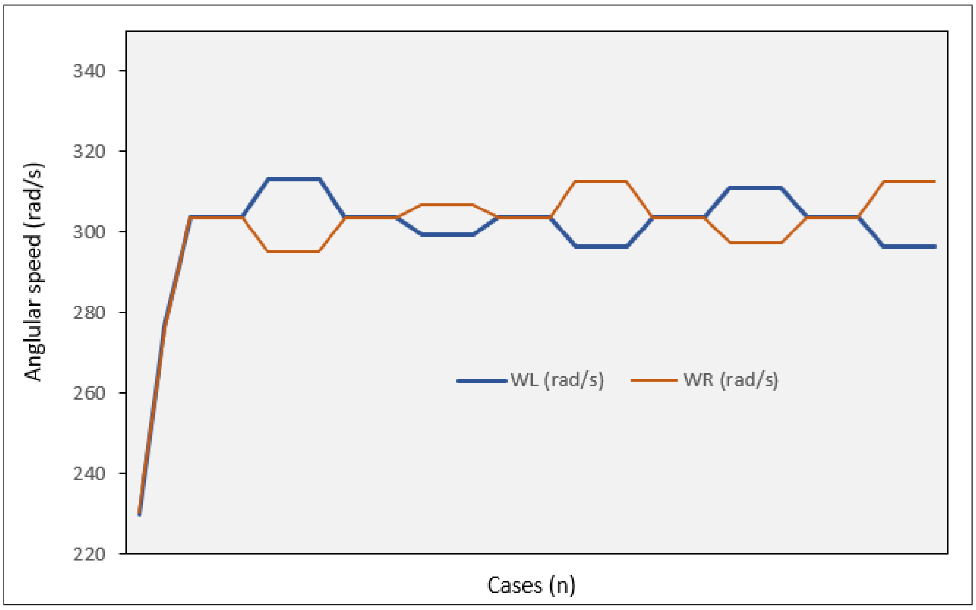

Figure 54. Left and right wheels angular speed comparison

5 Conclusions

This project has described the design and implementation of an electronic differential for an electric multi-drive vehicle. Simulation results show that the structure permits and ensures static and dynamic performance for electronic differential. The electronic differential controls the drive wheel speeds with good accuracy on both flat and curved paths. The absence of the need for a power transmission system is also an advantage.

Initially, it was proposed to use a speed control system of two electric machines, responsible for the propulsion of an electric vehicle, commanded by two PWM signals generated by an electronic central unit in charge for the management and treatment of signals received from the sensors.

To validate the system, after calibration, we must assume that the DC electric machines are rotating at same speed when the same acceleration is applied to both motors and steering wheel angle is equal to zero, that is to say, vehicle is going in a straight path. Also, tests have been carried out: steering wheel angle sensor position, accelerator pedal position, PWM pulse width comparison for electric machines speed control and electric machines rotational speed.

Considering fuzzy accelerator results, when a small angle is set (near 0 degrees), it offers a more direct response, because the probability of losing vehicle control is minimized. Thus, stability control fuzzy system presents a low impact.

In order to validate the electronic differential system and confirm the differential effect it applied to in-wheels electric machines, several tests were executed to analyze the speed of rotation of each EM, for different situations. Analyzing the obtained data, it can be seen that the prototype presents a differential effect, compatible with the Ackermann-Jeantand equations for vehicle modeling and the results present an expressive similarity according to simulations, validating the system, thus enabling the possibility for electric machine control in multi-drive vehicles.

6 Discussion and further work

This section presents some proposed improvements and work development. These challenges include both aspects of software development and features to implement an electronic differential in an electric vehicle.

Finding other ways to improve vehicle stability, the study of yaw rate should be considered because it measures the rotation rate of the car. In other words, the sensor determines how far off-axis a car is in a turn. This information is then fed into the microcontroller that compares the data with angular wheel speed, SWA and accelerator position, and, if the system senses too much yaw, it activates the electronic stability control. Also, a lateral acceleration sensor could measure the g-force from a turn and send that information to the ECU. This information, along with the fuzzy accelerator, can bring about a significant improvement in stability.

For the transmission of sensor signals, shielded cables can be used to reduce noise caused by other circuits or external factors, causing differences in the information obtained by the microcontroller.

Concerning the data acquisition, a study of the uncertainties and other acquisition methods could been done in order to reduce the error, performing a better calibration of the electric machines speed of rotation, fundamental for the system. To measure the vehicle real speed, when the system is implemented, it is possible to place four magnetic encoders to measure the speed of each wheel, creating a redundancy with the speed measurement provided by the microcontroller, having a more robust system with less failure.

Further study of the electronic differential speed controller can be done by analyzing the PI controller and changing its topology to PID, IP or PD. In this way, it is possible to verify which is the best control method to reach the device reference speed, enhancing the electronic differential system.

References

[1] L.ZHAI AND S. DONG, *Electronic differential speed steering control for four in-wheel motors independent drive vehicle.* Proc. World Congress Intelligence Control Automation, no 1, 780 to 783, 2011.

[2] SEN, P. C., *Electric motor drives and control-past, present, and future. IEEE Transactions on Industrial Electronics, 37(6), 562–575. doi:10.1109/41.103462, 1990.*

[3] RAYMOND A. SERWAY, JOHN W. JEWETT, *Physics for Scientists and Engineers, Volume 1: Physics, Physics, 34, 2011.*

[4] WRIGHT, M. T., *"The Antikythera Mechanism reconsidered". Interdisciplinary science reviews. 32 (1), 2007.*

[5] SANDUN, S.K., *Electric Machine Differential For Vehicle Traction Control And Stability Control, Purdue University, 13, 2013.*

[6] DEJUN YIN, DANFENG SHAN AND JIA-SHENG HU, *A Study on the Control Performance of Electronic Differential System for Four-Wheel Drive Electric Vehicles, 3, 2017.*

[7] M. W. CHOI, J. S. PARK, B. S. LEE, AND M. H. LEE, *The performance of independent wheels steering vehicle (4WS) applied Ackerman geometry, IEEE International Conference on Control, Automation and Systems (ICCAS), 197 to 202, 2008.*

[8] MERVE YđLDđRđM, EYYÜP ÖKSÜZTEPE, BURAK TANYERI, AND HASAN KÜRÜM,*Design of Electronic Differential System for An Electric Vehicle with In-Wheel Motor, 2016 IEEE Power and Energy Conference at Illinois (PECI), 2, 2016.*

[9] WILLIAM F. MILLIKEN, DOUGLAS L.MILLIKEN, *Race car vehicle dynamics, Automobiles, Racing-Dynamics, 1L243M55, 623, 1995.*

[10] DAVID KENNEDY, *"Open Differential vs. Spool - Differential Differences". Four-Wheeler Network, 12, 2004.*

[11] ZERAOULIA, M., BENBOUZID, M. E. H., DIALLO, D. *Electric Motor Drive Selection Issues for HEV Propulsion Systems: A Comparative Study. IEEE Transactions on Vehicular Technology, 55(6), doi:10.1109/TVT.2006.878719, 1756–1764, 2006*

[12] A. RAVI, S. PALANI *Slip Rate Estimation for Vehicle Stability Enhancement using Fuzzy based Electronic Differential controller, Journal of Engineering Science and Technology Vol. 10, No.11, School of Engineering, Taylor's University, 1497 to 1507, 2015.*

[13] RYAN SPRAGUE, *An Analysis of Current Battery Technology and Electric Vehicles, Journal of Undergraduate Research 5, 1, 2015.*

[14] ZHENPO WANG, WENLIANG ZHANG, *Battery System Matching and Design for a Formula Student Electric Racecar, ITEC Asia-Pacific, 1569889247, 2014.*

[15] SYNTHESIS PARTNERS, LLC, *Technology and Market Intelligence: Hybrid Vehicle Power Inverters Cost Analysis, Prepared for the Department of Energy by Synthesis Partners, LLC, Contract Number: DE-DT0002121, 11, 2011.*

[16] DILLARD, TED, *Rare Look Inside A Tesla Model S Battery Pack". InsideEVs. Retrieved 2014-09-23, 2014.*

[17] JURGEN BUEKERS, MIRJA VAN HOLDERBEKE, JOHAN BIERKENS, LUC INT PANIS, *Health and environmental benefits related to electric vehicle introduction in EU countries, Transportation Research Part D 33 (2014) 26–38, 2014.*

[18] UNICAMP E-TEAM MEDIA, *http://www.unicamperacing.com.br/web/unicamp-e-racing–media.html, 2017.*

[19] DAVID B. RICHARDSON, *Electric vehicles and the electric grid: A review of modeling approaches, Impacts, and renewable energy integration, Renewable and Sustainable Energy, Review 19, 247 to 254, 2013.*

[20] PUTRUS G. A. SUWANAPINGKARL P. JOHNSTON D. BENTLEY E. C. NARAYANA M. *Impact of Electric Vehicles on Power Distribution Networks, DOI: 10.1109/VPPC.2009.5289760, 1 to 2, 2009.*

[21] SHISHENG HUANG, BRI-MATHIAS S. HODGE, FARZAD TAHERIPOUR, JOSEPH F. PEKNY, GINTARAS V. REKLAITIS AND WALLACE E. TYNER, *The Effects of Electricity Price on PHEV Competitiveness. GPRI Digital Library. Paper 6, 2010.*

[22] A. R. HOLDWAY; A. R. WILLIAMS; OLIVER R. INDERWILDI; DAVID A. KING, *Indirect emissions from electric vehicles: emissions from electricity generation, Energy and Environmental Science, doi:10.1039/C0EE00031K, 3, 2010.*

[23] MUHAMMAD H. RASHID, *Power Electronics Handbook (Fourth Edition), ISBN: 978-0-12-811407-0, 2007.*

[24] RAUL ROJAS, *Neural Networks - A Systematic Introduction, Springer-Verlag, 502, 350 illustrations, 1996.*

[25] FRANCK DERNONCOURT, *Introduction to fuzzy logic, Massachusetts Institute of Technology, January, 19, 2013.*

[26] ROGER JANG, *Neuro-Fuzzy modeling: Architecture, Analysis and Application, PhD thesis, University of California, Berkeley, 18, 1992.*

[27] ELEAZAR SANCHES, *Controle por aprendizado acelerado e Neuro-fuzzy de sistemas Servo - hidráulicos de alta frequência, 0721406/CA - PUC-RIO, 2009.*

[28] MICHAEL GIESSELMANN, ATTILA KARPATI, *Power Electronics: Converters, Applications, and Design - John Wiley, New York, 1989, 104 to 121, 2009.*

[29] HONEYWELL INC., *Applications of Magnetic Position Sensors, Application Note, AN211, 1, 2015.*

[30] AVAGO TECHNOLOGIES,*Motion Control Encoders in Electrical Motor Systems, Design Guide, 3, 2007.*

[31] GENERAL MOTORS, *https://www.gmpartsdirect.com/oem-parts/gm-position-sensor-26104070, GM, 2010.*

[32] KADA HARTANI, MOHAMED BOURAHLA, YAHIA MILOUD, M. SEKOUR, *Electronic Differential with Direct Torque Fuzzy Control for Vehicle Propulsion System, Turk J Elec Eng and Comp Sci, Vol.17, No.1, 1, 2009.*

[33] SEEMA CHOPRA, R. MITRA, VIJAY KUMAR *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:1, No:9, 2007.*

[34] B. GASBAOUI, C. ABDELKADER, L. ADELLAH *Multi-input multi-output fuzzy logic controller for utility electric vehicle, Faculty of Sciences and Technology, Department of Electrical Engineering Bechar University, DOI 10.2478/v10171-011-0023-6, 2011.*

[35] HEIKKI LAITINEN *Improving electric vehicle energy efficiency with two-speed gearbox, Aalto University, School of engineering, 25, 2017.*

[36] HUGHES, DRURY, *Electric Motors and Drives: Fundamentals, Types and Applications, 284 to 286, 2013.*

[37] Yimin Ehsani et al., *Modern Electric, Hybrid Electric and Fuel Cell Vehicles, 204, 2005.*

[38] C. C. CHAN AND K. T. CHAU, *Modern Electric Vehicle Technology, University of Oxford, Electrical Engineering, 2001.*

[39] TOYOTA MOTOR CORPORATION, *Toyota Prius Electric vehicle motor, 2007.*

[40] YOICHI HORI ET AL, *Optimal Slip Ratio Estimator for Traction Control System of Electric Vehicle based on Fuzzy Inference, January, Electrical Engineering in Japan, DOI: 10.1002/eej.1033, 56 to 63, 2000.*

[41] FUJIMOTO H., *Electric vehicles wirelessly charge as they drive, 2017.*

[42] STEVE CORRIGAN, TEXAS INSTRUMENT, *Introduction to the Controller Area Network (CAN), Application Report SLOA101B, 2017.*

[43] J. A. COOK, J. S. FREUDENBERG, *Controller Area Network (CAN), EECS 461, Fall 2008.*

[44] VAN ZANTEN, *Stability Control, Automotive Electronics Handbook, pp. 17.1 – 17.33, 1999.*

[45] AZEDDINE DRAOU, *Electronic differential speed control for two in-wheels motors drive vehicle, 4th International Conference on Power Engineering, Energy and Electrical Drives, 2013.*

[46] KEC SEMICONDUCTOR, *KIA7805AP, Bipolar linear integrated circuit, 2010.*

A

ECU Software

```
/*

ELECTRONIC CONTROL UNIT (ECU)

ELECTRONIC DIFFERENTIAL CONTROL - e-Power UFRGS Racing Team v1.0

2018 - Formula SAE Electric Vehicle

Author: Carlos BOHM AGUSTI
Date: 09/08/2018

 */

//========= LIBRARIES ==================

#include <math.h> // Math Library

//.......... Fuzzy Libraries ..............
#include <FuzzyRule.h>
#include <FuzzyComposition.h>
#include <Fuzzy.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzyOutput.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzySet.h>
#include <FuzzyRuleAntecedent.h>
//.........................................
```

```
//................ CAN Communication Libraries.....................
#include <mcp_can.h> //CAN library
#include <SPI.h>   // peripherals library
//.....................................................................
//================== PIN DEFINITIONS ===============================
// DC Motor 1
// Digital Pins

#define Encoder_C1 3
#define Encoder_C2 4
#define IN1       8
#define IN2       7

//Analog Pins
#define Pin_ADC   A1
#define Encoder_PPR 9
byte Encoder_C1Last;
int duracao;
boolean Direcao;

// PWM Motor Outputs
int Output_Motor_Left = 11; // PWM Motor 1
int Output_Motor_Right = 12; // PWM Motor 2

// DC Motor 2
#define Encoder_C1_2 10
#define Encoder_C2_2 5
//...............CAN DEFINITIONS.....................
#define CAN0_INT 2                          // Set INT to pin 2
MCP_CAN CAN0(53);                           // Set CS to pin 10
//=====================================================================
//================== VARIABLE DECLARATIONS =========================

 //.............. PID CONTROL .............
```

```
//double Setpoint, Input, Output;
double WtotalPID;
double VSpeedRef,VSpeedActual, WSpeedED, WSpeedError; //Setpoint, Input, Output
double WSpeedError1;
double Vtotalkmh;
double Setpoint;
double Input;
// Read vehicle actual speed
  double VSA;
  double speedset;
  double Outputteste;
float Wtotal;
float Vtotal;
//....................................PID
    LIBRARY.........................................
class PID{
public:
  double error;
  double sample;
  double lastSample;
  double kP, kI, kD;
  double P, I, D;
  double pid;
  double setPoint;
  long lastProcess;

  PID(double _kP, double _kI, double _kD){
    kP = _kP;
    kI = _kI;
    kD = _kD;
  }

  void addNewSample(double _sample){
    sample = _sample;
```

```
  }

  void setSetPoint(double _setPoint){
    setPoint = _setPoint;
  }

  double process(){
    // Implementacao PID
    error = setPoint - sample;

    float deltaTime = (millis() - lastProcess) / 1000.0;
    lastProcess = millis();

    //P
    P = error * kP;

    //I
    I = I + (error * kI) * deltaTime;

    //D
    D = (lastSample - sample) * kD / deltaTime;
    lastSample = sample;

    // Soma tudo
    pid = P + I + D;

    return pid;
  }
};

PID meuPid(1.25, 1.0, 0); // KP KI KD // best found

//.............. FUZZY LOGIC - SPEED REFERENCE
    .......................................
```

```
// Instantiating an object of library
Fuzzy* fuzzy = new Fuzzy();
//...... Steering Wheel Sensor..............

FuzzySet* HighLeft = new FuzzySet(-40, -30, -25, -10);
FuzzySet* LowLeft = new FuzzySet(-40, -15,-10, 0);
FuzzySet* ZeroSW = new FuzzySet(-21.6, -2.4, 2.4, 21.6); // Angle Zero Steering
    Wheel angle sensor
FuzzySet* LowRight = new FuzzySet(0, 10, 15, 40);
FuzzySet* HighRight = new FuzzySet(10, 25, 30, 40);

//...... Accelerator Position Sensor ........

FuzzySet* ZeroAPP = new FuzzySet(-10, 0, 10, 40); // Position Zero Accelerator
FuzzySet* VeryLow = new FuzzySet(0, 15, 25, 40);
FuzzySet* Low = new FuzzySet(20, 30, 40, 50);
FuzzySet* Medium = new FuzzySet(15, 45, 55, 85);
FuzzySet* High = new FuzzySet(50, 60, 70, 80);
FuzzySet* VeryHigh = new FuzzySet(50, 75, 85, 100);
FuzzySet* MAX = new FuzzySet(60, 90, 100, 110);
//...............................................................................
//.......................VEHICLE DIMENSIONS AND ACKERMANN'S
    VARIABLES..................

float L = 1.530; // Distance between front and rear wheels
float D = 1.300; // Distance between axis gauge - bitola
float WheelRadius = 0.531; // Wheel Radius in meters
float Sensor_SW_Output; // Steering Wheel angle sensor output
float Sensor_APP_Output; // Accelerator pedal position sensor output
float Sensor_ActualSpeed_Output; // Actual vehicle speed sensor output
float Sensor_ActualSpeed_Output_Right; // Actual vehicle speed sensor output
    right motor
```

```
float Sensor_ActualSpeed_Output_Left; // Actual vehicle speed sensor output
    left motor
float Sensor_ActualSpeed_Output_rads;
float PID_Speed_Output; // PID Reference speed for electronic differential
float SpeedReference_Fuzzy;
float V_Speed; // Vehicle speed in m/s
float W_Speed; // Vehicle speed in rad/s
float SW_Angle_Rad; // Steering wheel angle in radians
float R; // Vehicle rotational radius
float SW_Angle_Tangent; // Steering wheel angle tangent in radians
float Steering_Wheel_Angle_Fuzzy; // steering wheel after transmission SW ->
    rotation angle
int WSpeed_Left; // Angular velocity left rear wheel
int WSpeed_Right; // Angular velocity right rear wheel
int PWM_Output_Left_map; // PWM Output Left mapping
int PWM_Output_Right_map; // PWM Output right mapping
//.....................................................................................
//............................ACCELERATOR PEDAL POSITION VARIABLES
    INITIALISATION.................
//Conectar saida do TL082 na porta analogica A1 do Arduino.
//Conectar ground do Arduino.
//Potenciometro 1: 0.78 a 4.64V, Alimentacao: fio amarelo; Aterramento: fio
    vermelho; Sinal: fio branco;
//Potenciometro 2: 0.39 a 2.31V, Alimentacao: fio preto; Aterramento: fio azul;
    Sinal: fio laranja;

//Potenciometro 1:
const int potenciometro1_Accelerator = A1; // Pino A1 do arduino.
int valorLido = 0;
float minimo_Accelerator = 161;
float maximo_Accelerator = 962;
float percentual_Accelerator;

//Potenciometro 2:
```

```
/*const int potenciometro2 = A2; // pino A2 do arduino.
int valorLido2 = 0;
float minimo2 = 82;
float maximo2 = 479;
float percentual2;
*/
//..................... Variables ............................................
enum RotationState // DC Motor rotation state
{
  kClock,
  kCounterClock,
  kStop
};
volatile enum RotationState rotation_state_Left;
volatile enum RotationState rotation_state_Right;
volatile bool c2_state;
volatile bool interrupt_flag_left; // Interruption
volatile bool interrupt_flag_right; // Interruption
volatile bool c3_state;
uint16_t potentiometer_value;
// Encoder 1
//uint8_t pwm_value_left;
//uint32_t last_time_left;
//uint32_t time_interval_left;
//uint16_t motor_rpm_left;
uint8_t pwm_value_left;
uint32_t last_time_left;
uint32_t time_interval_left;
uint16_t motor_rpm_left;
// Encoder 2
uint16_t motor_rpm_right;
uint32_t last_time_right;
uint32_t time_interval_right;
uint8_t pwm_value_right;
```

```
//........................................................................................
// ELECTRIC MACHINES ROTATIONAL DIRECTION

void Clockwise()
{
  digitalWrite(IN1, HIGH);
 digitalWrite(IN2, LOW);
}
void CounterClockwise()
{
  digitalWrite(IN1, LOW);
 digitalWrite(IN2, HIGH);
}

//.........................................................................................
//..................................STEERING WHEEL ANGLE - CAN
    VARIABLES..................
char StringOne[5],StringTwo[5];
float passo=0.0125, angulo_real;
long unsigned int rxId;
unsigned char len = 0;
unsigned char rxBuf[8];
char msgString[128];                    // Array to store serial string
long int a;
float anguloinv;
//...........................................................................................

void setup() {
  //...........PINS AND INITIAL VALUES SETTINGS................................
  //  Motor 1
   pinMode(Encoder_C1, INPUT);
   pinMode(Encoder_C2, INPUT);
   pinMode(Pin_ADC, INPUT);
```

```
   pinMode(IN1, OUTPUT);

   pinMode(IN2, OUTPUT);
   pinMode(Output_Motor_Right, OUTPUT);
   Serial.begin(115200); // Serial Initialisation


// Variables initialisation values
   potentiometer_value = 0;
   pwm_value_left = 0;
   time_interval_left = 0;
   motor_rpm_left = 0;
   c2_state = 0;
   rotation_state_Left = kStop;
   interrupt_flag_left = 0;
   last_time_left = micros();


//  Motor 2
  pinMode(Encoder_C1_2, INPUT);
  pinMode(Encoder_C2_2, INPUT);
  pinMode(Output_Motor_Left, OUTPUT);
  motor_rpm_right = 0;
  interrupt_flag_right = 0;
  c2_state = 0;
  rotation_state_Right = kStop;
//.....................................................................
//...........................STEERING WHEEL CAN SETTINGS.............

// Initialize MCP2515 running at 8MHz with a baudrate of 500kb/s and the masks
    and filters disabled.
  if (CAN0.begin(MCP_ANY, CAN_500KBPS, MCP_8MHZ) == CAN_OK)
    Serial.println("MCP2515 Initialized Successfully!");
  else
```

```
    Serial.println("Error Initializing MCP2515...");

  CAN0.setMode(MCP_NORMAL);                // Set operation mode to normal so
      the MCP2515 sends acks to received data.
  pinMode(CAN0_INT, INPUT);                       // Configuring pin for /INT
      input
  Serial.println("MCP2515 Library Receive Example...");

//.....................................................................
//............... FUZZY LOGIC - SPEED REFERENCE ..........................
//...... Steering Wheel Sensor Input...............

  FuzzyInput* angle = new FuzzyInput(1);
  angle->addFuzzySet(HighLeft);
  angle->addFuzzySet(LowLeft);
  angle->addFuzzySet(ZeroSW);
  angle->addFuzzySet(LowRight);
  angle->addFuzzySet(HighRight);

  fuzzy->addFuzzyInput(angle);

//...... Accelerator Position Sensor Input.........
  FuzzyInput* AcceleratorPosition = new FuzzyInput(2);
  AcceleratorPosition->addFuzzySet(ZeroAPP);
  AcceleratorPosition->addFuzzySet(VeryLow);
  AcceleratorPosition->addFuzzySet(Low);
  AcceleratorPosition->addFuzzySet(Medium);
  AcceleratorPosition->addFuzzySet(High);
  AcceleratorPosition->addFuzzySet(VeryHigh);
  AcceleratorPosition->addFuzzySet(MAX);

  fuzzy->addFuzzyInput(AcceleratorPosition);
```

```
//...... Speed Reference Output .........
 FuzzyOutput* SpeedReference = new FuzzyOutput(1);

 /*1*/ FuzzySet* ZeroSR = new FuzzySet(-10, 0, 10, 40);
  SpeedReference->addFuzzySet(ZeroSR);
 /*2*/ FuzzySet* VeryLowSR = new FuzzySet(0, 15, 29, 40);
  SpeedReference->addFuzzySet(VeryLowSR);
 /*3*/FuzzySet* LowSR = new FuzzySet(20, 30, 44, 50);
  SpeedReference->addFuzzySet(LowSR);
 /*4*/ FuzzySet* MediumSR = new FuzzySet(15, 45, 55, 85);
  SpeedReference->addFuzzySet(MediumSR);
 /*5*/ FuzzySet* HighSR = new FuzzySet(50, 56, 70, 80);
  SpeedReference->addFuzzySet(HighSR);
 /*6*/ FuzzySet* VeryHighSR = new FuzzySet(50, 71, 85, 100);
  SpeedReference->addFuzzySet(VeryHighSR);
 /*7*/ FuzzySet* MAXSR = new FuzzySet(60, 90, 105, 150);
  SpeedReference->addFuzzySet(MAXSR);

  fuzzy->addFuzzyOutput(SpeedReference);
//.............................................................................
//.............. FUZZY RULES SETUP - SPEED REFERENCE .........................
// Building Fuzzy Rules

 /*1*/ FuzzyRuleAntecedent* AcceleratorPositionZeroAPPAndangleHighLeft = new
    FuzzyRuleAntecedent();
       AcceleratorPositionZeroAPPAndangleHighLeft->joinWithAND(ZeroAPP,
          HighLeft);

         FuzzyRuleConsequent* thenSpeedReferenceZeroSR = new
            FuzzyRuleConsequent();
  thenSpeedReferenceZeroSR->addOutput(ZeroSR);

  FuzzyRule* fuzzyRule1 = new FuzzyRule(1,
     AcceleratorPositionZeroAPPAndangleHighLeft, thenSpeedReferenceZeroSR);
```

```
  fuzzy->addFuzzyRule(fuzzyRule1);

/*2*/ FuzzyRuleAntecedent* AcceleratorPositionZeroAPPAndangleZeroSW = new
   FuzzyRuleAntecedent();
       AcceleratorPositionZeroAPPAndangleZeroSW->joinWithAND(ZeroAPP, ZeroSW);

        FuzzyRule* fuzzyRule2 = new FuzzyRule(2,
            AcceleratorPositionZeroAPPAndangleZeroSW, thenSpeedReferenceZeroSR);
  fuzzy->addFuzzyRule(fuzzyRule2);

/*3*/ FuzzyRuleAntecedent* AcceleratorPositionZeroAPPAndangleHighRight = new
   FuzzyRuleAntecedent();
       AcceleratorPositionZeroAPPAndangleHighRight->joinWithAND(ZeroAPP,
           HighRight);

       FuzzyRule* fuzzyRule3 = new FuzzyRule(3,
           AcceleratorPositionZeroAPPAndangleHighRight,
           thenSpeedReferenceZeroSR);
       fuzzy->addFuzzyRule(fuzzyRule3);

/*4*/ FuzzyRuleAntecedent* AcceleratorPositionZeroAPPAndangleLowLeft = new
   FuzzyRuleAntecedent();
       AcceleratorPositionZeroAPPAndangleLowLeft->joinWithAND(ZeroAPP, LowLeft);

       FuzzyRule* fuzzyRule4 = new FuzzyRule(4,
           AcceleratorPositionZeroAPPAndangleLowLeft, thenSpeedReferenceZeroSR);
       fuzzy->addFuzzyRule(fuzzyRule4);

/*5*/ FuzzyRuleAntecedent* AcceleratorPositionZeroAPPAndangleLowRight = new
   FuzzyRuleAntecedent();
       AcceleratorPositionZeroAPPAndangleLowRight->joinWithAND(ZeroAPP,
           LowRight);
```

```
       FuzzyRule* fuzzyRule5 = new FuzzyRule(5,
           AcceleratorPositionZeroAPPAndangleLowRight,
           thenSpeedReferenceZeroSR);
       fuzzy->addFuzzyRule(fuzzyRule5);
//........

/*6*/ FuzzyRuleAntecedent* AcceleratorPositionVeryLowAndangleHighLeft = new
   FuzzyRuleAntecedent();
       AcceleratorPositionVeryLowAndangleHighLeft->joinWithAND(VeryLow,
           HighLeft);

        FuzzyRuleConsequent* thenSpeedReferenceVeryLowSR = new
            FuzzyRuleConsequent();
       thenSpeedReferenceVeryLowSR->addOutput(VeryLowSR);

       FuzzyRule* fuzzyRule6 = new FuzzyRule(6,
           AcceleratorPositionVeryLowAndangleHighLeft,
           thenSpeedReferenceVeryLowSR);
       fuzzy->addFuzzyRule(fuzzyRule6);

/*7*/ FuzzyRuleAntecedent* AcceleratorPositionVeryLowAndangleZeroSW = new
   FuzzyRuleAntecedent();
       AcceleratorPositionVeryLowAndangleZeroSW->joinWithAND(VeryLow, ZeroSW);

        FuzzyRule* fuzzyRule7 = new FuzzyRule(7,
            AcceleratorPositionVeryLowAndangleZeroSW,
            thenSpeedReferenceVeryLowSR);
       fuzzy->addFuzzyRule(fuzzyRule7);

/*8*/ FuzzyRuleAntecedent* AcceleratorPositionVeryLowAndangleHighRight = new
   FuzzyRuleAntecedent();
       AcceleratorPositionVeryLowAndangleHighRight->joinWithAND(VeryLow,
           HighRight);
```

```
        FuzzyRule* fuzzyRule8 = new FuzzyRule(8,
            AcceleratorPositionVeryLowAndangleHighRight,
            thenSpeedReferenceVeryLowSR);
        fuzzy->addFuzzyRule(fuzzyRule8);

/*9*/ FuzzyRuleAntecedent* AcceleratorPositionVeryLowAndangleLowLeft = new
    FuzzyRuleAntecedent();
        AcceleratorPositionVeryLowAndangleLowLeft->joinWithAND(VeryLow, LowLeft);

 FuzzyRule* fuzzyRule9 = new FuzzyRule(9,
     AcceleratorPositionVeryLowAndangleLowLeft, thenSpeedReferenceVeryLowSR);
        fuzzy->addFuzzyRule(fuzzyRule9);

/*10*/ FuzzyRuleAntecedent* AcceleratorPositionVeryLowAndangleLowRight = new
    FuzzyRuleAntecedent();
        AcceleratorPositionVeryLowAndangleLowRight->joinWithAND(VeryLow,
            LowRight);

 FuzzyRule* fuzzyRule10 = new FuzzyRule(10,
     AcceleratorPositionVeryLowAndangleLowRight, thenSpeedReferenceVeryLowSR);
        fuzzy->addFuzzyRule(fuzzyRule10);

/*11*/ FuzzyRuleAntecedent* AcceleratorPositionLowAndangleHighLeft = new
    FuzzyRuleAntecedent();
        AcceleratorPositionLowAndangleHighLeft->joinWithAND(Low, HighLeft);

        FuzzyRule* fuzzyRule11 = new FuzzyRule(11,
            AcceleratorPositionLowAndangleHighLeft, thenSpeedReferenceVeryLowSR);
        fuzzy->addFuzzyRule(fuzzyRule11);

/*12*/ FuzzyRuleAntecedent* AcceleratorPositionLowAndangleHighRight = new
    FuzzyRuleAntecedent();
        AcceleratorPositionLowAndangleHighRight->joinWithAND(Low, HighRight);
```

```
       FuzzyRule* fuzzyRule12 = new FuzzyRule(12,
          AcceleratorPositionLowAndangleHighRight,
          thenSpeedReferenceVeryLowSR);
       fuzzy->addFuzzyRule(fuzzyRule12);


/*13*/ FuzzyRuleAntecedent* AcceleratorPositionLowAndangleZeroSW = new
    FuzzyRuleAntecedent();
       AcceleratorPositionLowAndangleZeroSW->joinWithAND(Low, ZeroSW);

        FuzzyRuleConsequent* thenSpeedReferenceLowSR = new
           FuzzyRuleConsequent();
       thenSpeedReferenceLowSR->addOutput(LowSR);

       FuzzyRule* fuzzyRule13 = new FuzzyRule(13,
          AcceleratorPositionLowAndangleZeroSW, thenSpeedReferenceLowSR);
       fuzzy->addFuzzyRule(fuzzyRule13);

/*14*/ FuzzyRuleAntecedent* AcceleratorPositionLowAndangleLowLeft = new
    FuzzyRuleAntecedent();
       AcceleratorPositionLowAndangleLowLeft->joinWithAND(Low, LowLeft);

       FuzzyRule* fuzzyRule14 = new FuzzyRule(14,
          AcceleratorPositionLowAndangleLowLeft, thenSpeedReferenceLowSR);
       fuzzy->addFuzzyRule(fuzzyRule14);

/*15*/ FuzzyRuleAntecedent* AcceleratorPositionLowAndangleLowRight = new
    FuzzyRuleAntecedent();
       AcceleratorPositionLowAndangleLowRight->joinWithAND(Low, LowRight);

 FuzzyRule* fuzzyRule15 = new FuzzyRule(15,
    AcceleratorPositionLowAndangleLowRight, thenSpeedReferenceLowSR);
       fuzzy->addFuzzyRule(fuzzyRule15);
```

```
/*16*/ FuzzyRuleAntecedent* AcceleratorPositionMediumAndangleHighLeft = new
    FuzzyRuleAntecedent();
        AcceleratorPositionMediumAndangleHighLeft->joinWithAND(Medium, HighLeft);

         FuzzyRule* fuzzyRule16 = new FuzzyRule(16,
             AcceleratorPositionMediumAndangleHighLeft, thenSpeedReferenceLowSR);
        fuzzy->addFuzzyRule(fuzzyRule16);

/*17*/ FuzzyRuleAntecedent* AcceleratorPositionMediumAndangleHighRight = new
    FuzzyRuleAntecedent();
        AcceleratorPositionMediumAndangleHighRight->joinWithAND(Medium,
            HighRight);

         FuzzyRule* fuzzyRule17 = new FuzzyRule(17,
             AcceleratorPositionMediumAndangleHighRight,
             thenSpeedReferenceLowSR);
        fuzzy->addFuzzyRule(fuzzyRule17);

/*18*/ FuzzyRuleAntecedent* AcceleratorPositionMediumAndangleZeroSW = new
    FuzzyRuleAntecedent();
        AcceleratorPositionMediumAndangleZeroSW->joinWithAND(Medium, ZeroSW);

 FuzzyRuleConsequent* thenSpeedReferenceMediumSR = new FuzzyRuleConsequent();
        thenSpeedReferenceMediumSR->addOutput(MediumSR);

        FuzzyRule* fuzzyRule18 = new FuzzyRule(18,
            AcceleratorPositionMediumAndangleZeroSW, thenSpeedReferenceMediumSR);
        fuzzy->addFuzzyRule(fuzzyRule18);

/*19*/ FuzzyRuleAntecedent* AcceleratorPositionMediumAndangleLowLeft = new
    FuzzyRuleAntecedent();
        AcceleratorPositionMediumAndangleLowLeft->joinWithAND(Medium, LowLeft);
```

```
        FuzzyRule* fuzzyRule19 = new FuzzyRule(19,
            AcceleratorPositionMediumAndangleLowLeft,
            thenSpeedReferenceMediumSR);
        fuzzy->addFuzzyRule(fuzzyRule19);

/*20*/ FuzzyRuleAntecedent* AcceleratorPositionMediumAndangleLowRight = new
    FuzzyRuleAntecedent();
        AcceleratorPositionMediumAndangleLowRight->joinWithAND(Medium, LowRight);

        FuzzyRule* fuzzyRule20 = new FuzzyRule(20,
            AcceleratorPositionMediumAndangleLowRight,
            thenSpeedReferenceMediumSR);
        fuzzy->addFuzzyRule(fuzzyRule20);

/*21*/ FuzzyRuleAntecedent* AcceleratorPositionHighAndangleHighLeft = new
    FuzzyRuleAntecedent();
        AcceleratorPositionHighAndangleHighLeft->joinWithAND(High, HighLeft);

        FuzzyRule* fuzzyRule21 = new FuzzyRule(21,
            AcceleratorPositionHighAndangleHighLeft, thenSpeedReferenceMediumSR);
        fuzzy->addFuzzyRule(fuzzyRule21);

/*22*/ FuzzyRuleAntecedent* AcceleratorPositionHighAndangleHighRight = new
    FuzzyRuleAntecedent();
        AcceleratorPositionHighAndangleHighRight->joinWithAND(High, HighRight);

        FuzzyRule* fuzzyRule22 = new FuzzyRule(22,
            AcceleratorPositionHighAndangleHighRight,
            thenSpeedReferenceMediumSR);
        fuzzy->addFuzzyRule(fuzzyRule22);

/*23*/ FuzzyRuleAntecedent* AcceleratorPositionHighAndangleZeroSW = new
    FuzzyRuleAntecedent();
        AcceleratorPositionHighAndangleZeroSW->joinWithAND(High, ZeroSW);
```

```
 FuzzyRuleConsequent* thenSpeedReferenceHighSR = new FuzzyRuleConsequent();
       thenSpeedReferenceHighSR->addOutput(HighSR);

       FuzzyRule* fuzzyRule23 = new FuzzyRule(23,
          AcceleratorPositionHighAndangleZeroSW, thenSpeedReferenceHighSR);
       fuzzy->addFuzzyRule(fuzzyRule23);

/*24*/ FuzzyRuleAntecedent* AcceleratorPositionHighAndangleLowLeft = new
   FuzzyRuleAntecedent();
       AcceleratorPositionHighAndangleLowLeft->joinWithAND(High, LowLeft);

FuzzyRule* fuzzyRule24 = new FuzzyRule(24,
   AcceleratorPositionHighAndangleLowLeft, thenSpeedReferenceHighSR);
       fuzzy->addFuzzyRule(fuzzyRule24);

/*25*/ FuzzyRuleAntecedent* AcceleratorPositionHighAndangleLowRight = new
   FuzzyRuleAntecedent();
       AcceleratorPositionHighAndangleLowRight->joinWithAND(High, LowRight);

FuzzyRule* fuzzyRule25 = new FuzzyRule(25,
   AcceleratorPositionHighAndangleLowRight, thenSpeedReferenceHighSR);
       fuzzy->addFuzzyRule(fuzzyRule25);

/*26*/ FuzzyRuleAntecedent* AcceleratorPositionVeryHighAndangleHighLeft = new
   FuzzyRuleAntecedent();
       AcceleratorPositionVeryHighAndangleHighLeft->joinWithAND(VeryHigh,
          HighLeft);

FuzzyRule* fuzzyRule26 = new FuzzyRule(26,
   AcceleratorPositionVeryHighAndangleHighLeft, thenSpeedReferenceHighSR);
       fuzzy->addFuzzyRule(fuzzyRule26);
```

```
/*27*/ FuzzyRuleAntecedent* AcceleratorPositionVeryHighAndangleHighRight = new
    FuzzyRuleAntecedent();
        AcceleratorPositionVeryHighAndangleHighRight->joinWithAND(VeryHigh,
            HighRight);

        FuzzyRule* fuzzyRule27 = new FuzzyRule(27,
            AcceleratorPositionVeryHighAndangleHighRight,
            thenSpeedReferenceHighSR);
        fuzzy->addFuzzyRule(fuzzyRule27);

/*28*/ FuzzyRuleAntecedent* AcceleratorPositionVeryHighAndangleZeroSW = new
    FuzzyRuleAntecedent();
        AcceleratorPositionVeryHighAndangleZeroSW->joinWithAND(VeryHigh, ZeroSW);

         FuzzyRuleConsequent* thenSpeedReferenceVeryHighSR = new
             FuzzyRuleConsequent();
        thenSpeedReferenceVeryHighSR->addOutput(VeryHighSR);

        FuzzyRule* fuzzyRule28 = new FuzzyRule(28,
            AcceleratorPositionVeryHighAndangleZeroSW,
            thenSpeedReferenceVeryHighSR);
        fuzzy->addFuzzyRule(fuzzyRule28);

/*29*/ FuzzyRuleAntecedent* AcceleratorPositionVeryHighAndangleLowLeft = new
    FuzzyRuleAntecedent();
        AcceleratorPositionVeryHighAndangleLowLeft->joinWithAND(VeryHigh,
            LowLeft);

        FuzzyRule* fuzzyRule29 = new FuzzyRule(29,
            AcceleratorPositionVeryHighAndangleLowLeft,
            thenSpeedReferenceVeryHighSR);
        fuzzy->addFuzzyRule(fuzzyRule29);
```

```
/*30*/ FuzzyRuleAntecedent* AcceleratorPositionVeryHighAndangleLowRight = new
   FuzzyRuleAntecedent();
       AcceleratorPositionVeryHighAndangleLowRight->joinWithAND(VeryHigh,
          LowRight);

       FuzzyRule* fuzzyRule30 = new FuzzyRule(30,
          AcceleratorPositionVeryHighAndangleLowRight,
          thenSpeedReferenceVeryHighSR);
       fuzzy->addFuzzyRule(fuzzyRule30);

/*31*/ FuzzyRuleAntecedent* AcceleratorPositionMAXAndangleHighLeft = new
   FuzzyRuleAntecedent();
       AcceleratorPositionMAXAndangleHighLeft->joinWithAND(MAX, HighLeft);

       FuzzyRule* fuzzyRule31 = new FuzzyRule(31,
          AcceleratorPositionMAXAndangleHighLeft, thenSpeedReferenceHighSR);
       fuzzy->addFuzzyRule(fuzzyRule31);

/*32*/ FuzzyRuleAntecedent* AcceleratorPositionMAXAndangleHighRight = new
   FuzzyRuleAntecedent();
       AcceleratorPositionMAXAndangleHighRight->joinWithAND(MAX, HighRight);

       FuzzyRule* fuzzyRule32 = new FuzzyRule(32,
          AcceleratorPositionMAXAndangleHighRight, thenSpeedReferenceHighSR);
       fuzzy->addFuzzyRule(fuzzyRule32);

/*33*/ FuzzyRuleAntecedent* AcceleratorPositionMAXAndangleZeroSW = new
   FuzzyRuleAntecedent();
       AcceleratorPositionMAXAndangleZeroSW->joinWithAND(MAX, ZeroSW);

              FuzzyRuleConsequent* thenSpeedReferenceMAXSR = new
                 FuzzyRuleConsequent();
       thenSpeedReferenceMAXSR->addOutput(MAXSR);
```

```
       FuzzyRule* fuzzyRule33 = new FuzzyRule(33,
           AcceleratorPositionMAXAndangleZeroSW, thenSpeedReferenceMAXSR);
       fuzzy->addFuzzyRule(fuzzyRule33);

/*34*/ FuzzyRuleAntecedent* AcceleratorPositionMAXAndangleLowLeft = new
    FuzzyRuleAntecedent();
       AcceleratorPositionMAXAndangleLowLeft->joinWithAND(MAX, LowLeft);

       FuzzyRule* fuzzyRule34 = new FuzzyRule(34,
           AcceleratorPositionMAXAndangleLowLeft, thenSpeedReferenceMAXSR);
       fuzzy->addFuzzyRule(fuzzyRule34);

/*35*/ FuzzyRuleAntecedent* AcceleratorPositionMAXAndangleLowRight = new
    FuzzyRuleAntecedent();
       AcceleratorPositionMAXAndangleLowRight->joinWithAND(MAX, LowRight);

FuzzyRule* fuzzyRule35 = new FuzzyRule(35,
    AcceleratorPositionMAXAndangleLowRight, thenSpeedReferenceMAXSR);

       fuzzy->addFuzzyRule(fuzzyRule35);
    //........................................................................................

}

void loop()
{
//______________________SENSORS DATA ACQUISITION______________________

//......................ACCELERATOR POSITION SENSOR....................

  valorLido = analogRead(potenciometro1_Accelerator);

    //Serial.println(valorLido);
```

```
  percentual_Accelerator = ((valorLido - minimo_Accelerator) /
      (maximo_Accelerator - minimo_Accelerator)) * 100;

//......................................................................
//....................STEERING WHEEL ANGLE SENSOR....................
if (!digitalRead(CAN0_INT))                    // If CAN0_INT pin is low, read
    receive buffer
  {
    CAN0.readMsgBuf(&rxId, &len, rxBuf);  // Read data: len = data length, buf =
        data byte(s)

    if ((rxId & 0x80000000) == 0x80000000) // Determine if ID is standard (11
        bits) or extended (29 bits)
      sprintf(msgString, "Extended ID: 0x%.8lX DLC: %1d Data:", (rxId &
          0x1FFFFFFF), len);
    else
      sprintf(msgString, "Standard ID: 0x%.3lX  DLC: %1d Data:", rxId, len);

    //Serial.print(msgString);

    if ((rxId & 0x40000000) == 0x40000000) { // Determine if message is a
        remote request frame.
      sprintf(msgString, " REMOTE REQUEST FRAME");
     // Serial.print(msgString);
    } else {
      for (byte i = 0; i < len; i++) {
        sprintf(msgString, " 0x%.2X", rxBuf[i]);
        // Serial.print(msgString);
      }
    }
    sprintf(StringOne, "%.2X", rxBuf[5]); //dados necessarios para conversao
    sprintf(StringTwo, "%.2X", rxBuf[6]);
    strcat(StringOne,StringTwo);
    a= strtol(StringOne,NULL,16);
```

```
    if(a>=0&&a<=14399)
    {
      anguloinv=passo*a*(-1); //angulo em graus esquerda
    }else
      anguloinv=(65535-a)*passo; // angulo em graus direita

    //Serial.println(angulo_real,4);
  }

angulo_real = (-1)*anguloinv; // angulo real

//.....................................................................
//....................ACTUAL SPEED - MOTOR ENCODERS...................
//Motor 1...........Left Motor.................................
  Clockwise(); // Motor rotation direction
  //potentiometer_value = analogRead(Pin_ADC); // para teste
Sensor_ActualSpeed_Output = Vtotalkmh;
  //.....................................................................
//............... FUZZY - SPEED REFERENCE - INPUTS AND OUTPUTS ..........................
  //Steering_Wheel_Angle_Fuzzy = ((angulo_real)*(30/130)); // Transmission between steering wheel angle and rotation angle.
  Steering_Wheel_Angle_Fuzzy = angulo_real;

  //SWAS
  fuzzy->setInput(1, Steering_Wheel_Angle_Fuzzy); // Steering Wheel Angle - Range -30 - 30 degrees fuzzy

  //ACC
  fuzzy->setInput(2, percentual_Accelerator); // Accelerator Pedal Position // range 0 - 100% fuzzy

  fuzzy->fuzzify();
```

```
  SpeedReference_Fuzzy = fuzzy->defuzzify(1); // SpeedReference to PID after
      fuzzy logics -> WSpeedReference


analogWrite(6, VSpeedRef);

//........................................................................................
//.............PID
    CONTROL..............................................................

  VSpeedRef = SpeedReference_Fuzzy; // Speed reference in km/h Setpoint

  VSpeedActual = Sensor_ActualSpeed_Output; // Actual speed in km/h Input

 // WSpeedError = abs(WSpeedRef - WSpeedActual); // Error in km/h Distance away
     from setpoint

Setpoint = VSpeedRef;

Input = VSpeedActual;

// Read actual speed
 VSA = VSpeedActual;

  // Manda pro objeto PID!
  meuPid.addNewSample(VSA);

 speedset = VSpeedRef; // setpoint sample

  meuPid.setSetPoint(Setpoint);

//Outputteste = (meuPid.process() + Vtotal);
```

```
Outputteste = meuPid.process() + Vtotal;

PID_Speed_Output = Outputteste;

//.....................................................................
//......................ELECTRONIC DIFFERENTIAL VALUES UPDATE...........

V_Speed = ((PID_Speed_Output)/3.6); // Vehicle speed in m/s

W_Speed = (V_Speed/WheelRadius); // Vehicle speed in rad/s

SW_Angle_Rad = (Steering_Wheel_Angle_Fuzzy*0.0174533); // Steering wheel angle
    in radians 1 grau = 0,0174533 rad

SW_Angle_Tangent = tan(SW_Angle_Rad); // Steering wheel angle tangent in radius

R = (L/(SW_Angle_Tangent)); // Vehicle rotational radius


WSpeed_Left = (W_Speed*((L+((D/2)*SW_Angle_Tangent)))/L); // Angular velocity
    left rear wheel

WSpeed_Right = (W_Speed*((L-((D/2)*SW_Angle_Tangent)))/L); // Angular velocity
    right rear wheel

Vtotal = WheelRadius*Wtotal; // Velocidade em m/s

Vtotalkmh = (3.6 * Vtotal); // Velocidade em Km/h

//..................................................................................
//......................ELECTRONIC DIFFERENTIAL
    OUTPUT..............................
//Remapping Speed values for PWM Outputs 1 and 2
// WSpeed_Max =
```

```
// WSpeed_Min = 0 rad/s
PWM_Output_Left_map = map(WSpeed_Left, 5, 54, 80, 200); // ajuste da velocidade
    dos motores
PWM_Output_Right_map = map(WSpeed_Right, 5, 54, 100, 235); // ajuste



// Left Output pin 46
  analogWrite(46, PWM_Output_Left_map);
  analogWrite(4, PWM_Output_Left_map);

// Right Output pin 45
  analogWrite(45, PWM_Output_Right_map);
    analogWrite(5, PWM_Output_Right_map);



Wtotal = ((WSpeed_Left+WSpeed_Right)/2); // actual vehicle speed

 duracao = 0;
}
```
