

Explainable Navigation System Using Fuzzy Reinforcement Learning

Rolando Bautista-Montesano · Ricardo A. Ramirez-Mendoza · Rogelio Bustamante-Bello


Received: date / Accepted: date

Abstract Explainable outcomes in autonomous navigation have become crucial for drivers, other vehicles, as well as for pedestrians. Creating trustworthy strategies is mandatory for the integration of self-driving cars into quotidian environments. This paper presents the successful implementation of an explainable Fuzzy Deep Reinforcement Learning approach for autonomous vehicles based on the AWS DeepRacerTM platform. A model of the environment is created by transforming crisp values into linguistic variables. A fuzzy inference system is used to define the reward of the vehicle depending on its current state. Guidelines to define the actions and to improve performance of the reinforcement learning agent are given based on the characteristics of the existing hardware. The performance of the models is tested on tracks with distinctive properties using agents with different policies and action spaces, and shows explainable and successful navigation of the agent on diverse scenarios.


Keywords Deep learning · fuzzy inference system · Explainable Artificial Intelligence · Reinforcement Learning · Autonomous Vehicle

1 Introduction

The automotive academic and industrial research and development changed drastically in 2005 when the Defense Advanced Research Projects Agency (DARPA) Grand Challenge was won by Stanford's Stanley [1]. This event opened an enormous range of opportunities in diverse fields such as computer science, electrical, mechanical, transportation and logistics engineering branches. Due to the interdisciplinary nature of the problem, the transfer of technology and knowledge between academic communities and industry has become vital to the development of a solution. Academically developed prototypes of self-driving cars systems were required to fit industry standards in terms of performance, safety, and trust [3]. The autonomous navigation problem seemed to be solved after the DARPA Urban Challenge [2], however, new challenges arose. These challenges comprise the trending research topics: data management, adaptive mobility, cooperative planning, validation, and evaluation. Data uncertainty and reliability represent the biggest challenges for data management. Without reliable data, the representation and modelling of the environment is at stake[4]. The topic of adaptive mobility relates to the introduction of autonomous vehicles to conventional environments. Safety [5], eco-driving [6], perception compensation [7], service orientation [8], and route context are the most important factors to consider before including this type of technology in our daily lives. As driving is a social task, cooperative planning is key for the individual [9],

Rolando Bautista-Montesano* 
orcid.org/0000-0002-0723-5799
E-mail: rolando.bautista@tec.mx

Rogelio Bustamante-Bello
E-mail: rbustama@tec.mx

Ricardo A. Ramirez-Mendoza* 
orcid.org/0000-0002-5122-507X
E-mail: ricardo.ramirez@tec.mx

*Corresponding authors

Tecnologico de Monterrey, Campus Ciudad de Mexico
Calle del Puente 222, Col. Ejidos de Huipulco, Tlalpan,
CDMX, Mexico
Tel.: +52 1 5483 2020

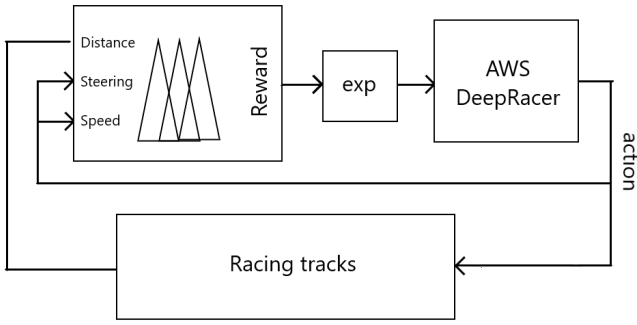


Fig. 1 General diagram of the RL FIS system

collective [10], and shared [11] decision making process. Lastly, the validation and evaluation of transition stability [12], ethics [13], and relaxation [14] of algorithms.

Despite all the mentioned approaches, functional navigation was far from being achieved. Studies like those presented in [15] and in [16] show that highly automated vehicles are far from being accepted from the passenger, driver, and pedestrian point of view. The main reasons are lack of confidence in why, how, when and where a determined action is taken. Human beings feel uncomfortable with the decision process of a vehicle. The purpose of this work is to present a novel human-like (approximate reasoning) explainable autonomous navigation system based on a Fuzzy Inference System (FIS) and Deep Reinforcement Learning for a closed-loop track. The motivation for developing this work is the opportunity for the use of Reinforcement Learning with autonomous vehicles, specifically in highway driving. Explainable and understandable criteria to the users for navigation is achieved in this work. The general diagram of the proposed system is depicted in Figure 1. The paper is organized as follows: Section 3 explains the basics of the employed algorithms. Section 4 describes the simulated environment and hardware in which the training and tests were made. Sections 5 and 6 provide further details of the proposed system, modelling and results. Finally, section 7 concludes this work and proposes future work.

Nomenclature

AI	Artificial Intelligence
AWS	Amazon Web Services
DL	Deep Learning
DNN	Convolutional Neural Network
DNN	Deep Neural Network
FIS	Fuzzy Inference System
MF	Membership Function
ML	Machine Learning

RL	Reinforcement Learning
XAI	Explainable Artificial Intelligence

2 State of the Art

There are three scenarios where autonomous vehicles operate: industrial, urban, and highway environments. A small review of these scenarios is presented. Extensive developments have been made regarding autonomous warehousing [17]. Localization represents the biggest challenge in industrial environments, because it relies on external infrastructure that can be easily damaged, and it is susceptible to noisy measurements. The main problem that urban navigation faces is the enormous quantity of data to be processed in real time. The environment is highly dynamic. Pedestrians, cars, trucks, buses, bicycles, among other elements must be detected, and the properties of their movement must be estimated in almost every direction [18]. The system must determine if an element is relevant to the movement of the vehicle even though the vehicle's movement horizon is short and constantly changing. Lastly, highway traffic flow is limited to one direction or perfectly delimited by a median strip when there are two directions. Average velocity is relatively constant. Elements in the environment are mostly motorized vehicles, and their behavior is less unpredictable than urban driving. All the elements in the environment are more uniform. Thus, the estimation horizon is longer and consistent with the direction the road. The vehicle's dynamics are simplified by reducing the quantity of accelerating and braking moments and by having smaller turn-angles. However, the speed of the car and the other vehicles implies a bigger hazard [19].

Artificial intelligence (AI), machine learning (ML) and deep learning have provided an important breakthrough in autonomous navigation during the last two decades. Their approaches are either cognitive or rational, or rule- or learning-based [20]. Approximate reasoning is the result of integrating cognitive and rule-based learning. These high-level models mirror human processes for solve-algorithms. A fuzzy inference decision system is proposed in [21] to check if the ego vehicle has enough time to change lanes based on an evaluation of the gaps between the surrounding obstacles. The work presented in [22] presents a combination of Hidden Markov Models and Gaussian Mixture Regression to mimic the driver's acceleration and braking model. The models that combine rational and learning-based approaches are heuristic. These are suitable when an optimal solution is not necessary and approximate solutions could be sufficient. Examples include a game-based approach designed for an agent in [23], and a

support vector machine to provide the control points to a Bézier curve-fitting method [24]. On the other hand, the algorithms that are cognitive and learning-based are considered to be human-like. They rely on logic and statistical bases, and provide adaptive reasoning to evolve appropriately to their environment. Examples include risk estimators, which are employed to interpret rational decisions from the scene analysis with a cognitive bias. The main advantage of cognitive and learning-based models is being intuitive by balancing a subjective level of acceptable risk with the objective safety [25]. Logic, an additional approach, is the intersection of rule-based and rational approaches, and is used in constrained and predictable environments. For example, the approach of [26] proposes a binary decision tree that enumerates all possible navigation lanes. The finite states machine proposed in [27] describes two longitudinal and four lateral state transitions with a specific contribution to an emergency stop assistant on highways.

Trending research shows that DL[28] and end-to-end learning [29] are the next step in developing intelligent navigation systems because of their logical and statistical bases and their ability to adapt with relative ease to changes in their environment. The work in [30] provides an end-to-end approach for self-driving vehicles where the DL model uses labeled information on proper correct driving. A related work on autonomous robot navigation and feature detection are achieved in [31]. In this work, a convolutional neural network (CNN) is utilized to detect doors and predict their position, enabling robot navigation. Another example is presented in [32]. The approach implements a target-driven visual navigation that avoids the DL's lack of generalization capability to new goals, and its data inefficiency. It achieves fast convergence, generalization across targets and scenes, and is end-to-end trainable. Similar work is reported by [33]. An autoencoder is employed to recognize when a query is novel, and revert it to a safe prior behavior. The deep learning system can be deployed in arbitrary environments and navigate 50% faster than a safe baseline policy in familiar types of environments. Drone racing is a popular application for DL [34], where images are inputted to a CNN to obtain a set of successive positions and a desired speed. This data is then introduced to a local planner to reach a desired goal. Another example is presented in [35]. A CNN is used to handle noisy images and generate a landmark to follow.

An RL approach is presented in [36], where a multi-agent policy relocates agents depending on their position. Their simulation results show that the number of taxis in an urban environment is decreased, improving the traffic flow. The biggest challenges DL faces are

ensuring the model has correctly learned the proper navigation behavior, the limited feedback from physical driving scenarios[37], the complexity and high energy demands of DL models [38], the lack of robustness to adversarial attacks [39], and the lack of transparency and explainability [40].

Modelling human behavior is one of the biggest challenges while automating navigation. Explainable Artificial Intelligence (XAI) is a trending research topic that intends to make AI systems results understandable to humans. Explanation methods are employed to gain insights into the functioning of an AI model. The most popular methods to explain the information content of the model are: *a)* explaining learned representations clarifies what has a given set of neurons learned [41]. *b)* Explaining individual predictions focuses on what a given set of data contributes to make a decision [42]. *c)* Explaining model behavior identifies the prediction strategies of the model [43]. *d)* Explaining with representative examples is used to detect biases in the data and to correct the model [44]. The transportation industry requires a high degree of explainability if autonomous vehicles are going to be used to enhance mobility. XAI systems can help clarify ambiguous situations so that a user can override control or allow the navigation system to keep driving. [45]. The experiment done in [46] demonstrates the importance of providing an explanation before an action has been done to make the user trust the system. A medical perspective on XAI is given by [47]. A vote of confidence is given to AI in aims of being used as a trusted tool to medical staff. Special emphasis is given on providing effective user interfaces. The framework for explainable Deep Neural Networks (DNN) proposed in [48] detects anomalies and provides a prediction coefficient, a textual description of the detected anomaly, and a list of factors to declare it an anomaly in terms of relevance scores.

Few proposals have been made to create explainable systems based on fuzzy logic, DL and Reinforcement Learning (RL). The main reason is difficulty finding solutions that are compatible with the Fuzzy Inference System (FIS) paradigm [49], however, knowledge from the DL model can be easily extracted with rule-based systems. This DL model provides a clear and reliable explanation of the cognitive process of the system [50] as long as the knowledge base remains simple, understandable and non-redundant [51]. XAI with fuzzy modelling is depicted in [52], where the basics and guidelines for a successful integration are provided, as well as proposing a linguistic XAI feedback method. An explainable fuzzy facial expression recognition is suggested in [53]. Their results show a considerable number of false-positives on emotions such as anger and joy

due to poor knowledge base definition, but an 86% of effectiveness on their other tests. In [54] an Unmanned Aerial Vehicle (UAV) employs an three-phased evolving XAI Fuzzy System to justify its deviation from its original path. They compare the performance of a Mamdani against a Sugeno model. The Sugeno model showed better results as it was more comprehensible than the Mamdani one. Fuzzy data conditioning and interpretation with DL is proposed by [55]. The effectiveness of their proposal is verified on raw data with high level of uncertainty applied to image categorization, high-frequency financial data prediction and brain MRI segmentation. [56] use a recurrent fuzzy neural network (RFNN) to identify and control a nonlinear dynamic system. The Lyapunov stability approach is used to iteratively test learning rates. Numerical simulations are done to validate the RFNN on a chaotic system, and to implement an adaptive control of a nonlinear system.

3 Methodology and Theoretical Framework

The employed methodological approach and a quick overview on the basics on DL, Reinforcement Learning (RL) and FIS is covered in this Section. The basic concepts are introduced in order to understand the presented approach. The combination of these algorithms leads to the development that is presented in this work.

3.1 Deep Learning

The term is commonly used when talking about deep artificial neural networks or deep reinforcement learning. This paper uses both approaches. Deep-learning (DL) is a set of multi-layer representations of non-linear hierarchical architectures and methods used to create a highly-abstracted and complex representation of a raw input [57]. It is used for supervised, unsupervised and hybrid learning applications. Supervised learning classifies data according to labeled data. Unsupervised learning creates a high-order correlation between data when no labels are available. Hybrid approaches classified data with the aid of unsupervised networks [58].

3.2 Reinforcement Learning

Reinforcement Learning (RL) is a machine learning (ML) method. Its goal is for an active agent to achieve a specific goal by interacting with the environment and learning the impact of its actions. The learning process is completed by maximizing a reward by an iterative trial and error sequence. The agent explores

the environment through states and its action space to *learn* which actions receive the highest rewards in unexplored states. Over time, the agent *finds* which set of actions lead to long-term rewards. The performance of the agent is limited by how well the reward function has been created. The reward function should tell if the result of an action should be reinforced, avoided or have a neutral response.

The strategy that the agent follows is known as a policy. The states and potential actions of an agent are determined by the environment, as are the rewards, therefore, the policy dictates which action to take to move from one state to another. These processes are also known as exploration and exploitation. RL takes the environment as an input and actions as outputs. The policy is known as the RL model and it is represented by a DNN. The training performance of the DNN is controlled by algorithm-dependent variables called hyperparameters. Their value is set empirically and they depend on the experience of the developer. This DNN takes video images as inputs and returns a discrete set of actions. The set of states that lead from an initial state to a goal state is known as an episode. [59]

Deep reinforcement learning combines neural networks with a reinforcement learning architecture. This way, software-defined agents can explore a virtual environment to order to obtain the highest reward possible. It unites function approximation with target optimization, mapping state-action pairs to expected rewards.

3.3 Fuzzy Inference System

Fuzzy set theory as proposed in [60] has been used to specify how well a measurement fits into a description with flexible boundaries. Zadeh stated in the *principle of incompatibility* that 'The closer one looks at a real-world problem, the fuzzier becomes the solution' [61]. Fuzzy set theory is considered as a generalization of classical set theory. Its strength is the representation of uncertainty with symbols and formal definitions [62]. From this set theory came fuzzy logic, where information modelled with logical expressions is used to trace membership to a given fuzzy set. This process is known as fuzzification. It was developed to emulate human-like logic with linguistic variables and to make decisions with a degree of uncertainty. A linguistic variable is a way of representing crisp values with words through fuzzy sets [63].

Fuzzy reasoning is done with a knowledge base, which is made up of fuzzy rules. The two basic operations found in the knowledge base are the intersection (AND) and union (OR). They are defined by the **min** and **max** operators respectively. Fuzzy rules are statements

that evaluate an antecedent and its implication. Conclusions are reached through *aggregation*. The most common method is to combine the outputs with the **max** operator for each rule into a single fuzzy set. The last step is to obtain a crisp value by defuzzifying with the center of sums method. FISs are widely used because they allow for easy models creation without a high mathematical complexity. Sometimes a system can be considered a black box with a set of inputs and outputs that can be modelled with linguistic variables. [64] Robot navigation approaches have been widely explored by [65]. The applications range from individual behavior and control to mapping and multi-layer integration.

3.3.1 Employed Membership Functions

The Gaussian membership functions (MF) were chosen according to the twelve rules proposed by [66]. The considerations that were given more weight are *representation*, *construction*, and *optimization*. Gaussian, Gaussian combination, S, and Z MFs are model driven, and require less construction and tuning parameters than trapezoidal ones. They are simpler in design because they are easier to represent and optimize, always continuous, and faster for small rule bases.

Gauss Membership Function This MF is used to represent linguistic variables with smooth transitions. They are usually placed in the middle of a set of variables. The mathematical definition of this MF is shown in Equation 1, where x is a value in a defined domain, σ is the standard deviation of the distribution, and c is its mean.

$$f(x; \sigma, c) = \exp - \frac{(x - c)^2}{2\sigma^2} \quad (1)$$

Gaussian Combination MF This other MF is also used to represent linguistic variables with smooth transitions, but those that need to maintain a maximum value for a given interval. They are usually placed in the middle of a set of variables. The mathematical definition of this MF is shown in Equation 2, where x is a value in a defined domain, σ_n is the standard deviation of the distribution, and c_n is the mean of the left or right part of the combined gaussian.

$$f(x; \sigma_1, c_1, \sigma_2, c_2) = \begin{cases} \exp - \frac{(x - c_1)^2}{2\sigma_1^2}, & x \leq c_1 \\ \exp - \frac{(x - c_2)^2}{2\sigma_2^2}, & x \geq c_2 \\ 1, & c_1 \leq x \leq c_2. \end{cases} \quad (2)$$

S MF The S-MF is employed to represent linguistic variables with smooth transitions, but that need to maintain a maximum value for a given interval from a given value to the greatest positive value in the domain. They are usually placed in the rightmost section of a set of variables. The mathematical definition of this MF is shown in Equation 3, where x is a value in a defined domain, a is the shoulder, and b is foot of the MF.

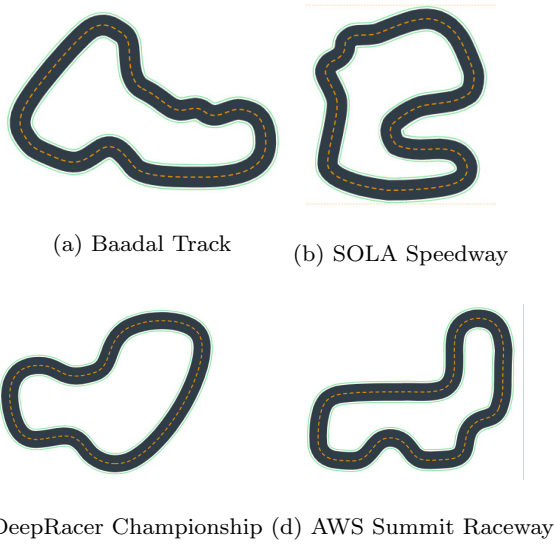
$$f(x; a, b) = \begin{cases} 0, & x \leq a \\ 2\left(\frac{x-a}{b-a}\right)^2, & a \leq x \leq \frac{a+b}{2} \\ 1 - 2\left(\frac{x-b}{b-a}\right)^2, & \frac{a+b}{2} \leq x \leq b \\ 1, & x \geq b. \end{cases} \quad (3)$$

Z MF The Z-MF is employed to represent linguistic variables with smooth transitions, but that need to maintain a maximum value for a given interval from the most negative value to a given value in the domain. They are usually placed in the leftmost section of a set of variables. The mathematical definition of this MF is shown in Equation 4, where x is a value in a defined domain, a is the foot, and b is the shoulder of the MF.

$$f(x; a, b) = \begin{cases} 1, & x \leq a \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2, & a \leq x \leq \frac{a+b}{2} \\ 2\left(\frac{x-b}{b-a}\right)^2, & \frac{a+b}{2} \leq x \leq b \\ 0, & x \geq b. \end{cases} \quad (4)$$

4 Simulation environment

RL requires three basic elements: an agent, the environment and a policy. The DeepRacerTM is the agent, the environment is the different tracks and the policy is defined by the developer. The information that the DeepRacerTM retrieves through sensors defines the environment state. The actions of the agent are the DeepRacer'sTM speed and steering angles, and rewards depend on the performance of the vehicle. Successfully navigating the full track will award the highest reward, while going off-track will be penalized. Amazon Web Services (AWS) created an integrated system to implement RL on autonomous driving applications. It provides a Graphic User Interface (GUI) to train an RL model with a specific reward function, optimization algorithm, environment, hyperparameters and to evaluate its performance on different tracks. [67] The AWS DeepRacerTM trains the RL agent with a Proximal Policy Optimization (PPO) algorithm. It is composed of two networks: an actor and a critic. The actor is in charge of selecting actions given an input and the critic determines the reward.

**Fig. 2** Test tracks**Table 1** Test tracks characteristics

Name	Length	Width	Curves
Baadal track	39m	107cm	7
SOLA Speedway	38m	106cm	7
DeepRacer Championship	23.12m	107cm	7
AWS Summit Raceway	22.57m	91cm	8

4.1 Environment

The selected environment in which the DeepRacer interacts for these research are four tracks: the Baadal Track, SOLA Speedway, the 2020 DeepRacer Championship Track and the AWS Summit Raceway. Figure 2 shows the selected tracks. AWS offers fourteen tracks to train and validate the RL models. The criteria to chose those four tracks are the following. The length of the track should be at least 20m, the width should be close to 1m. At least four curves are needed, two left and two right. All tracks have two lanes divided by a dashed yellow line. The edges of the track are delimited by a red and white or grey marker. The track is delimited by a green area. The topology of the tracks offer an equally valid environment for training and evaluation. The summary of the characteristics of the tracks is shown in Table 1. All the characteristics of the tracks and the DeepRacerTM are included on this section.

4.2 AWS DeepRacerTM

The AWS DeepRacerTM is a simulated and physical 1/18 vehicle with differential drive and Ackermann distribu-

Table 2 AWS DeepRacerTM Parameters

Parameter	Range
all wheels on track	Boolean
distance from center	0 to track width
is left of center	Boolean
speed	0 to max speed
steering angle	$[-30^\circ, 30^\circ]$
track width	0 to 107cm

tion. It has a mounted HD camera and an on-board computer. The on-board computer runs inference models based on RL so that it can drive itself on a track, and has 4GB of RAM and 32 GB of storage. It uses the Robot Operating System (ROS) as middleware to communicate the different input, output and processing nodes of the robot-vehicle. The speed of the DeepRacerTM ranges from 0 to 4 m/s. The developer can set the maximum speed of the vehicle. The granularity of the speed has a minimum of 1 and a maximum of 3 with steps of one. This is, if the granularity is 1 the actions of the vehicle will only be performed at the maximum speed. On the other hand, if a granularity different than 1 is chose, the agent will test the speed on steps of $v_{max}/granularity$ - zero exclusive. The steering angle can range from -30° to 30° . The developer can set the maximum steering angle within the mentioned range. Its granularity ranges from three to seven and follows the same principle of the speed.

The actor network of the vehicle can be composed by three or five layers, depending on the complexity of the model and the number of tasks the vehicle performs. It can be equipped with different sensors such as a camera, a stereo camera or a LIDAR. However, only a camera will be used to train and validate the models. The camera has a frame rate of 15 frames per second and the image is downsized to a 160x120 grey-scale resolution before it is inputted to the PPO algorithm.

AWS provides 23 parameters that supply information about the environment and the DeepRacerTM. However, only the ones that were used by the FIS are described in Table 2. Boolean parameters are converted into integers to avoid in-code decision structures. All other values are normalized to $[0,1]$ or to a $[-1,1]$ range. More information on this can be found in Section 5.1

5 Fuzzy Inference Reward System

The developed FIS recreates the expertise of a driver. The purpose of the FIS is to drive in the middle of a given track, while trying to maximize speed and minimize the number of required steering movements. The FIS receives three input variables: speed, steering an-

gle, and distance to the center of the track, and the system outputs the reward of the DL system. All inputs are normalized to a $[0,1]$ or $[-1,1]$ range. As mentioned in subsections 4.1, and 4.2, information on the track and the vehicle can be retrieved from parameters provided by AWS. The system developed in this project uses the Mamdani's direct method. First, the models of the environment and the vehicle are represented by a set of linguistic variables, which are composed of diverse MFs. These variables are then introduced into a Multiple Input Single Output (MISO) knowledge base. The inference engine is composed of a set of rules that will define the linguistic reward of the DL agent. Finally, the defuzzification process provides a numeric value for the reward.

5.1 Modelling

5.1.1 Speed

The AWS DeepRacer'sTM speed ranges from zero to four meters per second as mentioned in section 4.2. The value was first normalized to a $[0,1]$ range using the equation 5, where v_n is the normalized velocity, v the measured velocity, and V_{max} the maximum speed of the car.

$$v_n = \frac{v}{V_{max}} \quad (5)$$

This input was fuzzified into two MFs: *slow* and *fast*. Initially a *Medium* MF was proposed. However, the complexity of the system knowledge grew too much. Simulation results show that *Slow* and *Fast* are enough. S- and Z- MFs were chosen because of their smooth transitions. The *Slow* MF uses equation 4, Where $a = 0.04167$ and $b = 0.375$. The *Fast* MF uses equation 3, Where $a = 0.204$ and $b = 0.8108$. Figure 3 shows the S- and Z- MFs for the normalized speed.

5.1.2 Steering

The AWS DeepRacer'sTM steering angles are in the $[-30^\circ, 30^\circ]$ range. The input value was first normalized to $[-1,1]$ using equation 6, where θ_n is the normalized steering angle, θ the measured steering angle, and $\theta_{max} = 30^\circ$.

$$\theta_n = \frac{\theta}{\theta_{max}} \quad (6)$$

This input was fuzzified into three MFs: *Left*, *Center*, and *Right*. *Left* uses a Z-, *Right* a S- and *Center*

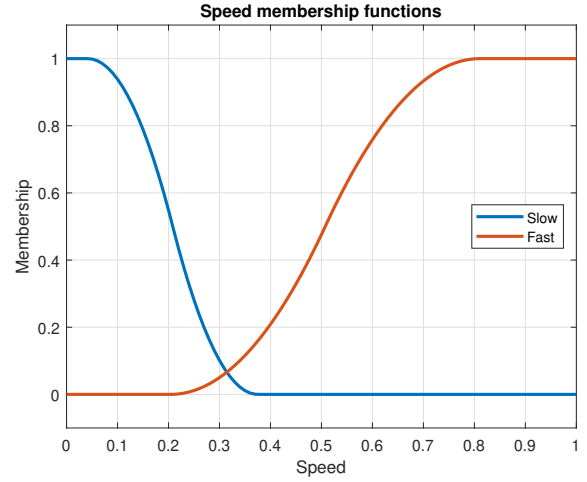


Fig. 3 Speed MFs

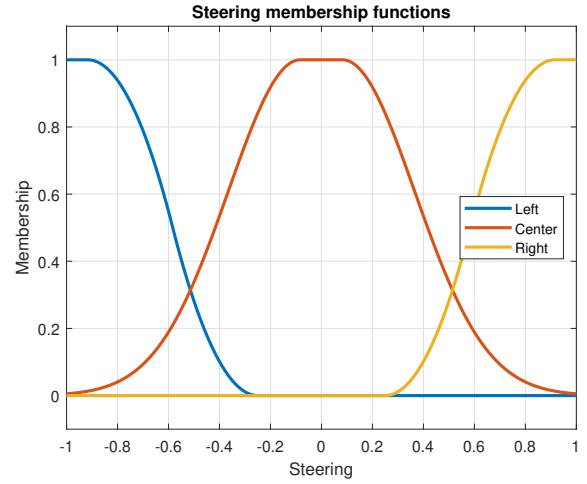


Fig. 4 Steering MFs

a Gauss Combination- MF. The S- and Z-MFs were again chosen because of their smooth transitions. The Gauss combination was used to make the system immune to small variations or disturbances while driving straight ahead. The *Left* MF uses equation 4, Where $a = -0.9167$ and $b = -0.25$. *Center* uses equation 2, where $\sigma_1 = 0.2831$, $c_1 = -0.08333$, $\sigma_1 = 0.2831$, and $c_1 = 0.08333$. The *Right* MF uses equation 3, Where $a = 0.25$ and $b = 0.9167$. Figure 4 shows the S-, Z-, and Gauss combination- MFs for the normalized steering.

5.1.3 Distance to center

AWS provides information regarding the position of the DeepRacerTM with respect to the central part of the track. It is a value in the $[0, \frac{trackwidth}{2}]$ range. It does not offer information whereas the vehicle is in the left of in the right lane. Therefore, the *is left of center* parameter becomes relevant to know the exact location of

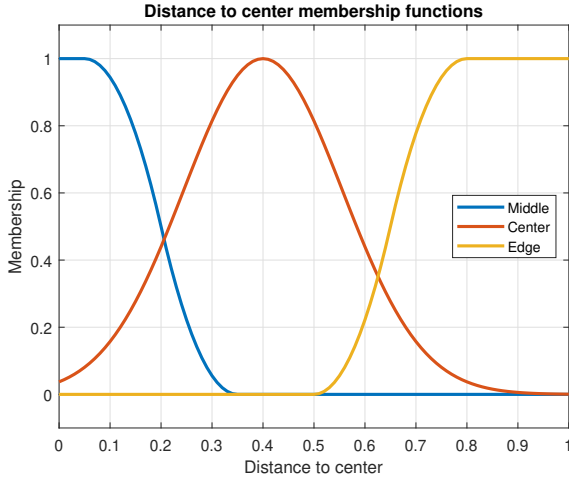


Fig. 5 Distance to Center MFs

the DeepRacerTM. The distance to center value is normalized to a $[0,1]$ range using equation 7, where d_n is the normalized distance, d the measured distance, and the *track width* depends on the raceway the vehicle is in, as shown in Table 1.

$$d_n = \frac{d}{\text{track}/2} \quad (7)$$

The distance to the center was fuzzified into three MFs: *Middle*, *Center*, and *Edge*. *Middle* uses a Z-, *Edge* a S- and *Center* a Gauss- MF. The three MFs were again chosen because of their smooth transitions. Initially the *Center* and *Edge* MFs were equally distributed on the lane. However, the vehicle drove off track in multiple DL models. Shifting those two functions to the *Middle* increased the performance of the FIS. The *Middle* MF uses equation 4, Where $a = 0.05$ and $b = 0.35$. *Center* uses equation 1, where $\sigma = 0.156$ and $c = 0.4$. The *Edge* MF uses equation 3, Where $a = 0.5$ and $b = 0.8$. Figure 5 shows the S-, Z-, and Gauss- MFs for the normalized distance to center.

5.1.4 Reward

Reward is the output linguistic variable. It has three possible MFs: *Low*, *Medium*, and *High*. The reward is in the $[-1,1]$ range. The *Low* MF uses equation 4, Where $a = -0.938$ and $b = -0.02381$. *Middle* uses equation 1, where $\sigma = 0.2224$ and $c = 0.0$. The *High* MF uses equation 3, Where $a = 0.0238$ and $b = 0.938$. Figure 6 shows the S-, Z-, and Gauss- MFs for the reward. After the defuzzification process, the reward crisp value is used as an exponent of e .

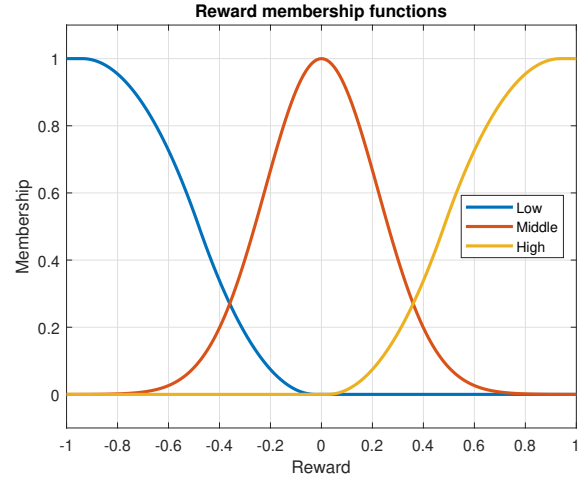


Fig. 6 Reward MFs

5.2 Rules

The knowledge base the FIS has, is designed to navigate towards the middle of a track. The system receives the highest reward when keeping an equal distance from the edges of a raceway. It allows the agent to drive in the middle of the right or left lanes, however, the reward would be lower. Consequently, the system receives the lowest reward when driving at the edges of the raceway. The rules encourage low to moderate speed when the agent is in a curve, and moderate to high speed when in a straight path. All rules have the same weight in the decision making process. The first five rules are indistinct if the DeepRacerTM is located in the middle of the track. Rules 5 to 11 change depending on the lane the vehicle is in. The *AND* operator is used to connect linguistic variables. They will now be enumerated:

1. IF *Steering* is *Center* AND *Distance to center* is *Middle* THEN *Reward* is *High*
2. IF *Steering* is *Left* AND *Distance to center* is *Middle* THEN *Reward* is *Middle*
3. IF *Steering* is *Right* AND *Distance to center* is *Middle* THEN *Reward* is *Middle*
4. IF *Steering* is *Center* AND *Distance to center* is *Center* THEN *Reward* is *Middle*
5. IF *Steering* is *Center* AND *Distance to center* is *Edge* THEN *Reward* is *Low*
6. (a) *Right lane*: IF *Steering* is *Right* AND *Distance to center* is *Center* THEN *Reward* is *Low*
 (b) *Left lane*: IF *Steering* is *Left* AND *Distance to center* is *Center* THEN *Reward* is *Low*
7. (a) *Right lane*: IF *Speed* is *Slow* AND *Steering* is *Left* AND *Distance to center* is *Center* THEN *Reward* is *High*

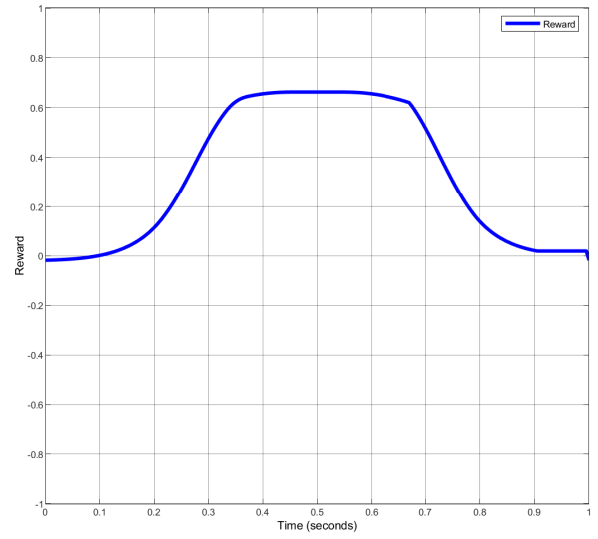
- (b) *Leftt lane: IF Speed is Slow AND Steering is Right AND Distance to center is Center THEN Reward is High*
- 8. (a) *Right lane: IF Speed is Fast AND Steering is Left AND Distance to center is Center THEN Reward is Middle*
- (b) *Leftt lane: IF Speed is Fast AND Steering is Right AND Distance to center is Center THEN Reward is Middle*
- 9. (a) *Right lane: IF Steering is Right AND Distance to center is Edge THEN Reward is Low*
- (b) *Left lane: IF Steering is Left AND Distance to center is Edge THEN Reward is Low*
- 10. (a) *Right lane: IF Speed is Slow AND Steering is Left AND Distance to center is Edge THEN Reward is Middle*
- (b) *Leftt lane: IF Speed is Slow AND Steering is Right AND Distance to center is Edge THEN Reward is Middle*
- 11. (a) *Right lane: IF Speed is Fast AND Steering is Left AND Distance to center is Edge THEN Reward is High*
- (b) *Leftt lane: IF Speed is Fast AND Steering is Right AND Distance to center is Edge THEN Reward is High*

5.3 FIS validation

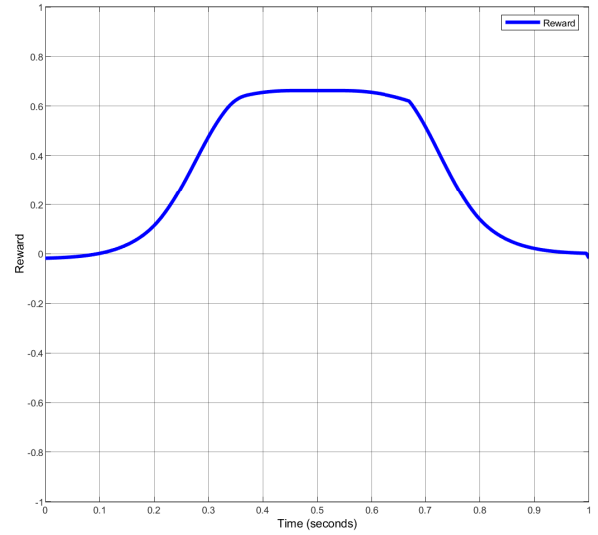
The **max** operator was used as the aggregation method. The defuzzification process was completed by computing the centroid. Before the FIS was implemented in the AWS DeepRacer™, it was tested in MATLAB Simulink™. The tests program was based on the one proposed in the MATLAB™ documentation. The *speed*, *steering angle*, and *distance to center* variables were simulated with signal generators and constants. A 1Hz saw-tooth wave with an amplitude of 2 simulated the leftmost to rightmost normalized steering. The combinations were input into the FIS and a Scope was used to check the response of the system. Constant values. The results of the tests will be shown from the most simple to the most complex. The vehicle is assumed to be located in the left lane.

5.3.1 Test 1

The first test, depicted in Figure 7, was conducted under the following conditions. The vehicle is at the middle of the track, the speed is constant, and the vehicle is steering from leftmost to rightmost. When the steering is *Left* $t = 0$ until approximately $t = 0.2$ the reward is *Middle*, as the value in the graph is close to zero.



(a) Speed is Slow



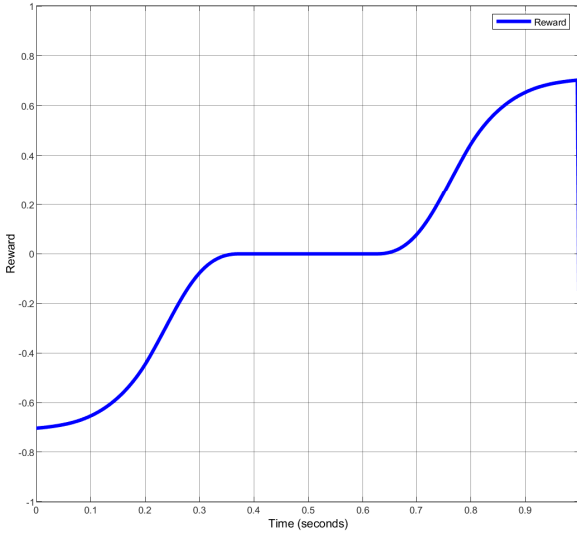
(b) Speed is Fast

Fig. 7 Reward - Vehicle at the center of the track with left to right steering

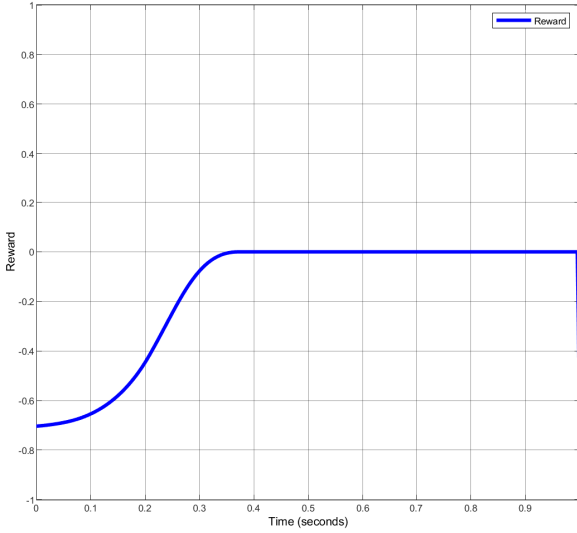
A similar behavior is found starting from $t = 0.8$ until $t = 1.0$ when the steering is *Right*. In contrast, the interval from $t = 0.2$ until $t = 0.8$ shows an increasing reward as the steering angle gets closer to 0° . A similar behavior can be found in Figures 7 a and b, however the *Speed* of the vehicle varies a-*Slow* and b-*Fast*.

5.3.2 Test 2

The second test, depicted in Figure 8, was conducted under the following conditions. The vehicle is at the middle of the lane, the speed is constant, and the vehicle is steering from leftmost to rightmost. Sub-figures a and



(a) Speed is Slow



(b) Speed is Fast

Fig. 8 Reward - Vehicle at the center of the lane with left to right steering

b behave similarly in the $t = 0$ until approximately $t = 0.7$ interval. During this interval, the steering angle is *Left* $t = 0$ until approximately $t = 0.2$ the reward is *Low*. This would make the vehicle drive off track, thus the value in the graph is close to -1. Staying in the same steering position would award a *Middle* reward, as shown in the $t = 0.3$ until $t = 0.7$ interval. During the final interval the variation in behavior depends on the *Speed*. When the vehicle is going *Slow*, it will tend towards the middle of the track with a *High* reward, whereas to attempt this movement while driving would *Fast* compromise the stability of the vehicle.

5.3.3 Test 3

The third and final test was conducted under the following conditions. The vehicle is at the outer edge of the lane, the speed is constant, and the vehicle is steering from leftmost to rightmost. Figure 9 shows the results of Test 3. Sub-figures a and b behave similarly in the $t = 0$ until approximately $t = 0.7$ interval. When the steering angle is *Left* or *Middle* there is a high probability that the vehicle will drive off track, thus the value in the graph is close to -1. During the last interval the difference in behavior depends on the *Speed*. When the vehicle is going *Slow*, it will move towards the middle of the track with a *Middle* reward, attempting to move slowly into the middle. In contrast, driving on a *Fast Speed* would make the agent drastically veer towards the middle.

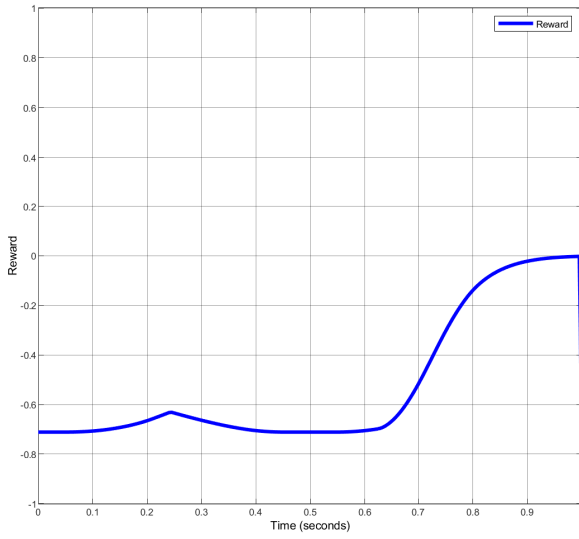
6 Deep Reinforcement Learning

After validation of the reward FIS, the next step is to implementation in the AWS DeepRacerTM. The reward function the agent uses is coded in Python 3.7. Only the *math* library is imported as the AWS platform does not support *numpy* or *scipy*. The MFs presented in Section 3.3 were adapted into functions aiming to be as time- and computational cost- efficient as possible. Built-in functions such as *lists*, *append*, *max*, and *sum* are intensively used.

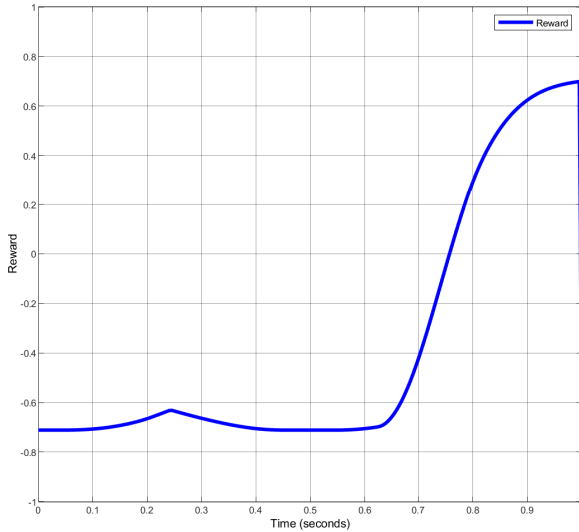
Several models were built, however only the most successful are presented in this paper. This section includes details on the DeepRacerTM action space, their curated hyperparameters, the training track and the performance of each model on different validation tracks. The validations tests of each model on every track were run 50 times. The reported performance is the average of such tests.

6.1 Model 1

This model demonstrates the effectiveness of the reward FIS. Its main goal is to successfully navigate through an entire track, regardless of its speed. The DeepRacerTM used to train this model has the following characteristics: (a) $v_{max} = 1m/s$, (b) speed granularity of 3, (c) $\theta_{max} = 30$, (d) steering granularity of three. The DNN topology has three hidden layers because of the simplicity of the task. It was trained for 175 iterations during 2 hours in the SOLA Speedway. The hyperparameters chosen were the default values in the AWS GUI. They can be consulted in the corresponding column of Table 3.



(a) Speed is Slow



(b) Speed is Fast

Fig. 9 Reward - Vehicle at the outer edge of the lane with left to right steering

The agent's cumulative reward, depicted in Figure 10a, started low. It maintained such behavior for almost the first two epochs of the training process. A similar behavior is shown by the average training completion, Figure 10b, and the average evaluation completion, Figure 10c. This performance can be explained using the exploration stage of the RL algorithm. The exploitation stage took over for the next two epochs, where the autonomous vehicle completed in average almost half of the track during the evaluation phase, regardless of the low average training completion. Starting from here there is a sustained, almost exponential, increment

on the reward. This is due to the augmenting completion average in the training and evaluation stages. The model results in a high reward and a close to a hundred percent evaluation phase completion.

The Baadal track and the SOLA Speedway were completed in similar times, due to their length (~ 1 min.). Consequently, the Championship Track and the Summit Raceway had a duration of half of the other two (~ 30 sec.). The completion times of Model 1 can be checked in the corresponding column of Table 4.

6.2 Model 2

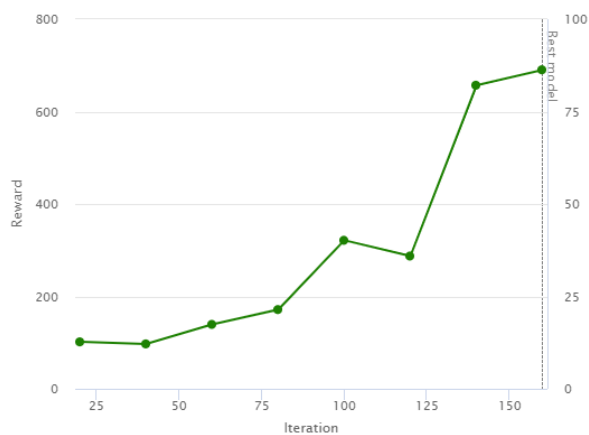
Once the effectiveness of the FIS was demonstrated in Model 1, the goal of Model 2 is to increase the speed of the vehicle while still completing all the training and validation tracks. The DeepRacerTM used to train this model has the following characteristics: (a) $v_{max} = 2m/s$, (b) speed granularity of 3, (c) $\theta_{max} = 30$, (d) steering granularity of five. The DNN topology has still three hidden layers, although the speed has been doubled. A speed of 2m/s is considered conservative. It was trained for 460 iterations in the SOLA Speedway. The hyperparameters chosen again were the default in the AWS GUI. They can be consulted in the corresponding column of Table 3.

The agent's cumulative reward, depicted in Figure 11a, started low. It maintained similar behavior for almost the first two epochs of the training process. A similar behavior is shown by the average training completion, Figure 11b, and the average evaluation completion, Figure 11c. This performance is typical of the exploration stage of the RL algorithm. The evaluation average completion held a continuous and sustained growth. The slope between epochs is gentle, however it was enough to generate a valid policy. This policy awarded increasing rewards on every epoch. Although the evaluation average completion did not exceed 50%, the model performed well on the validation tests on other tracks.

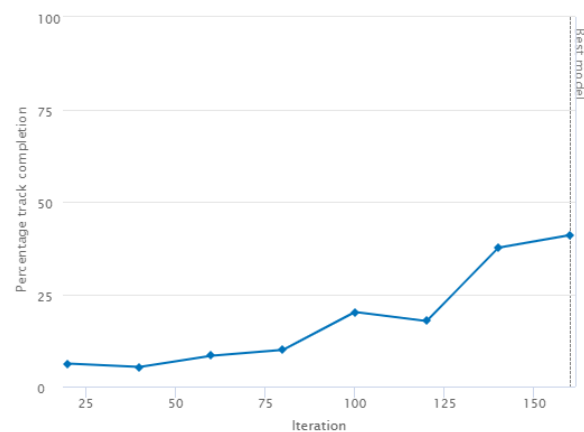
The Baadal track and the SOLA Speedway were completed in similar times due to their length (~ 37 sec). Consequently, the Championship Track and the Summit Raceway were close to the half of the times of the other two (~ 20 sec). The completion times of Model 2 can be checked in the corresponding column of Table 4.

6.3 Model 3

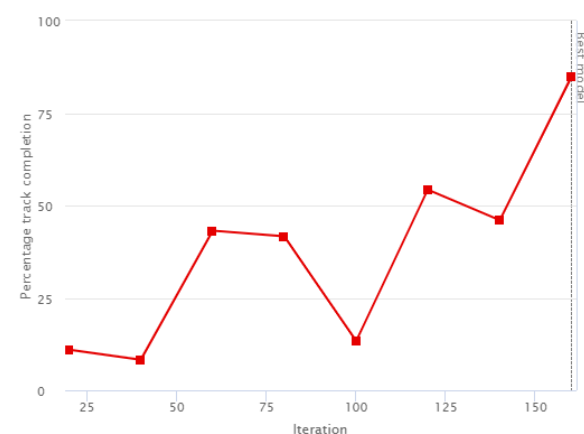
Model 2 behaved as expected. The challenge for Model 3 is to achieve successful navigation at a higher speed.



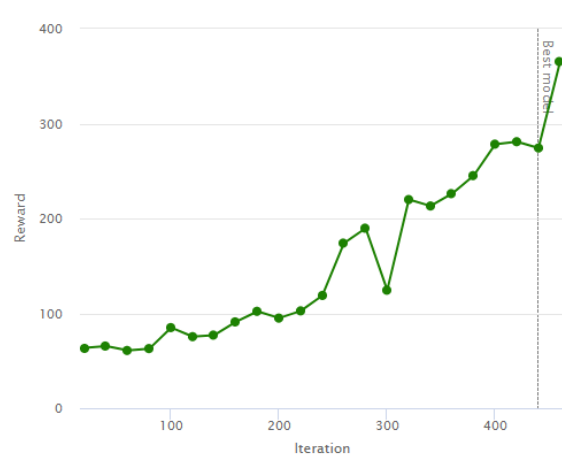
(a) Reward



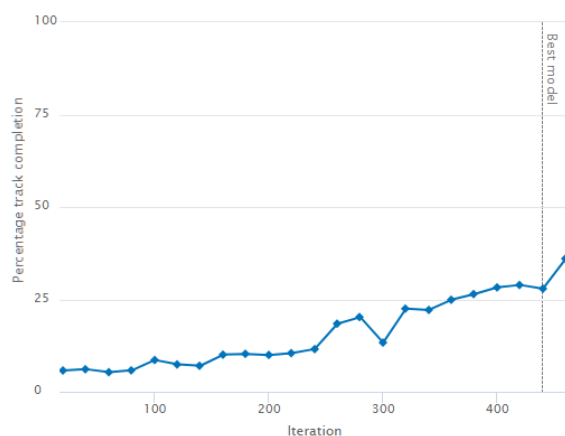
(b) Training Average Completion



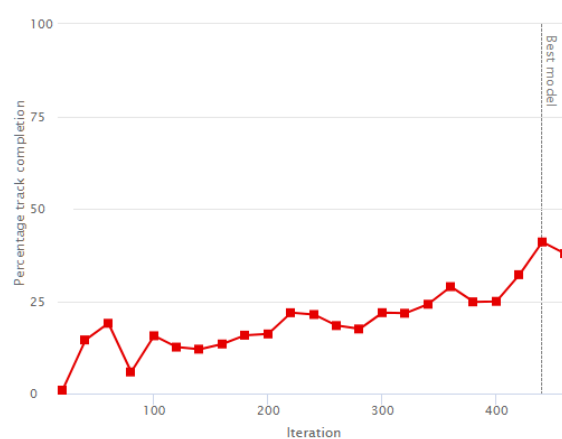
(c) Evaluation Average Completion



(a) Reward



(b) Training Average Completion



(c) Evaluation Average Completion

Fig. 10 Model 1 performance**Fig. 11** Model 2 performance

Due to the increase in the speed, extra considerations were made with respect to the characteristics of the vehicle. The DeepRacerTM used to train this model has the following characteristics: (a) $v_{max} = 3m/s$, (b) speed granularity of 3, (c) $\theta_{max} = 30$, (d) steering granularity of five. The DNN topology has again three hidden layers, although the speed has been increased. A speed of 3m/s is no longer considered conservative. However, the granularity of the steering angle is reduced. A reduced action space assists in the convergence of the RL training. It was trained for 1740 iterations on the SOLA Speedway and Baadal Track. The hyperparameters chosen were modified from the default in the AWS GUI. They can be consulted in the corresponding column of Table 3.

The model was originally trained in the SOLA Speedway. Validation tests on that track performed as expected, nevertheless, its performance was very poor in the other tracks. It failed to complete a lap. Strategies including retraining on the same track and modifying the hyperparameters were attempted with little success in improving its performance. Extending the training to the Baadal Track was the best solution. The graphs shown in Figure 12 show the results of the extended training on the Baadal Track. The agent's cumulative reward, depicted in Figure 12a is not representative as it stayed in a semi-static boundary. Nevertheless, the original training showed continuous growth. The model struggled to converge into a valid policy. The training and evaluation average completions during the first training grew on every epoch, which contrasts with the graphs on Figure 12b and c. It should be mentioned that even though the evaluation average completion does not grow steadily, it shows an overall growth. This model completed almost 100% of the training track.

The Baadal track and the SOLA Speedway were completed in similar times due to their length (~ 30 sec). Consequently, the Championship Track and the Summit Raceway were close to the half of the times of the other two (~ 19 sec). The completion times of Model 3 can be checked in the corresponding column of Table 4.

6.4 Model 4

Model 3 accomplished all its goals. The challenge was now to create a model that could use the DeepRacer'sTM maximum speed ($4m/s$). Initially, the same considerations were used as model 3, however, several models were unsuccessful. Combinations of the action space of the vehicle were tested, and the best solution was determined to be reduction of the granularity of the

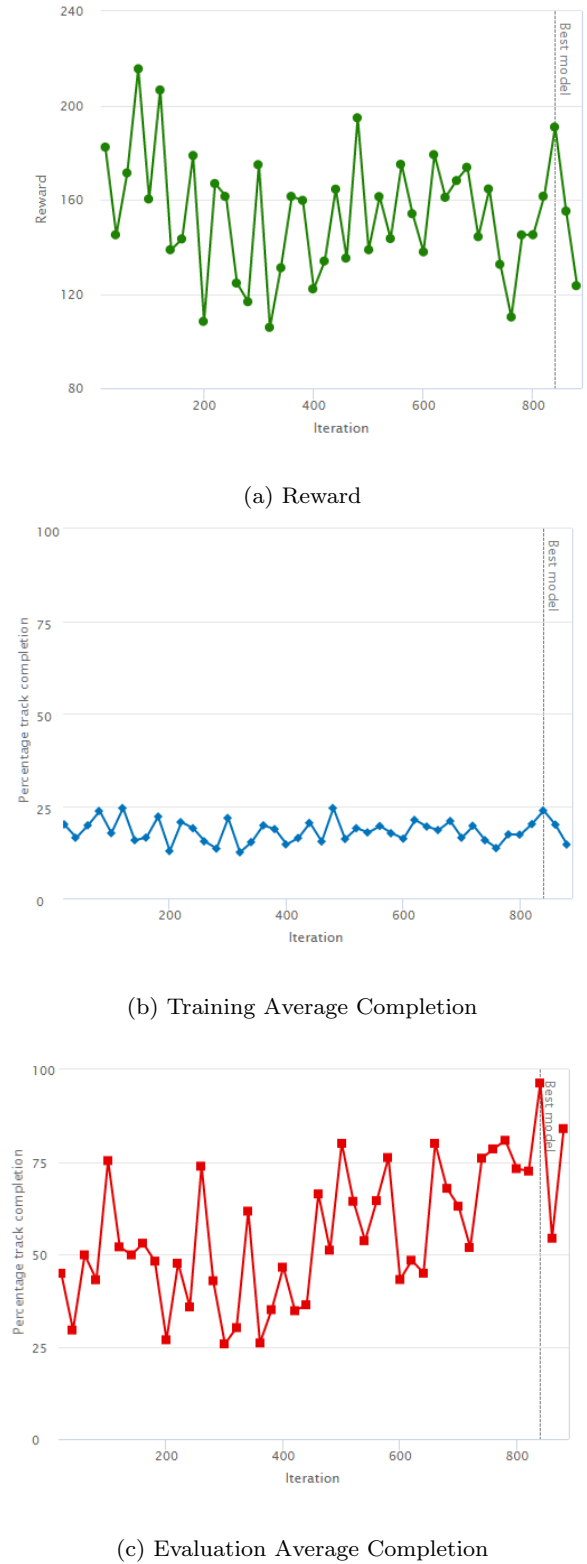


Fig. 12 Model 3 performance

steering angle. The DeepRacerTM used for the training of this model has the following characteristics: (a) $v_{max} = 4m/s$, (b) speed granularity of 3, (c) $\theta_{max} = 30$, (d) steering granularity of three. The DNN topology again has three hidden layers, although the speed has been increased. A speed of 4m/s is no longer considered conservative. However, the granularity of the steering angle is reduced. A reduced action space assists in the convergence of the RL training. It was trained for 4200 iterations on the Baadal Track. The hyperparameters chosen were modified from the default in the AWS GUI. They can be consulted in the corresponding column of Table 3.

Before the 2000th iteration the agent’s cumulative reward, depicted in Figure 13a, had a very low but sustained growth. Similar behaviors were spotted in the training completion average, 11b, and the evaluation completion average 11c. After the 2000th iteration there was a continuous and steep growth on the evaluation average. It seemed that it would converge rapidly to a successful policy. Nevertheless, the exploration phase found a local minimum, causing the model to decrease its performance. The training phase was going to be concluded around the 3000th iteration because of poor completion average, however after becoming aware of the improvement in performance, it was allowed for the completion of the training. The evaluation average completion reached a maximum of 80%.

The Baadal track and the SOLA Speedway were completed in similar times due to their length (~ 26 sec). Consequently, the Championship Track and the Summit Raceway were close to the half of the times of the other two (~ 16 sec). The completion times of Model 4 can be checked in the corresponding column of Table 4.

6.5 Model performance comparison

The presented models demonstrate the correct implementation of a FIS with RL. All of them completed the training and validation tracks. The hyperparameters used to train each RL model have a great impact not only on the policy but on the time the training lasts. Models 1 and 2 were trained using the default settings. Relatively few iterations were needed until they produced valid results. On the other hand, Model 3 has a higher value of entropy and discount factor. This means that it would explore the environment more broadly in search of a better policy in the current and future states. The training graphs in Figures 10, 11, 12, and 13 demonstrate the importance of the exploration phase of an RL algorithm. Although they only complete short segments of the track, they provide

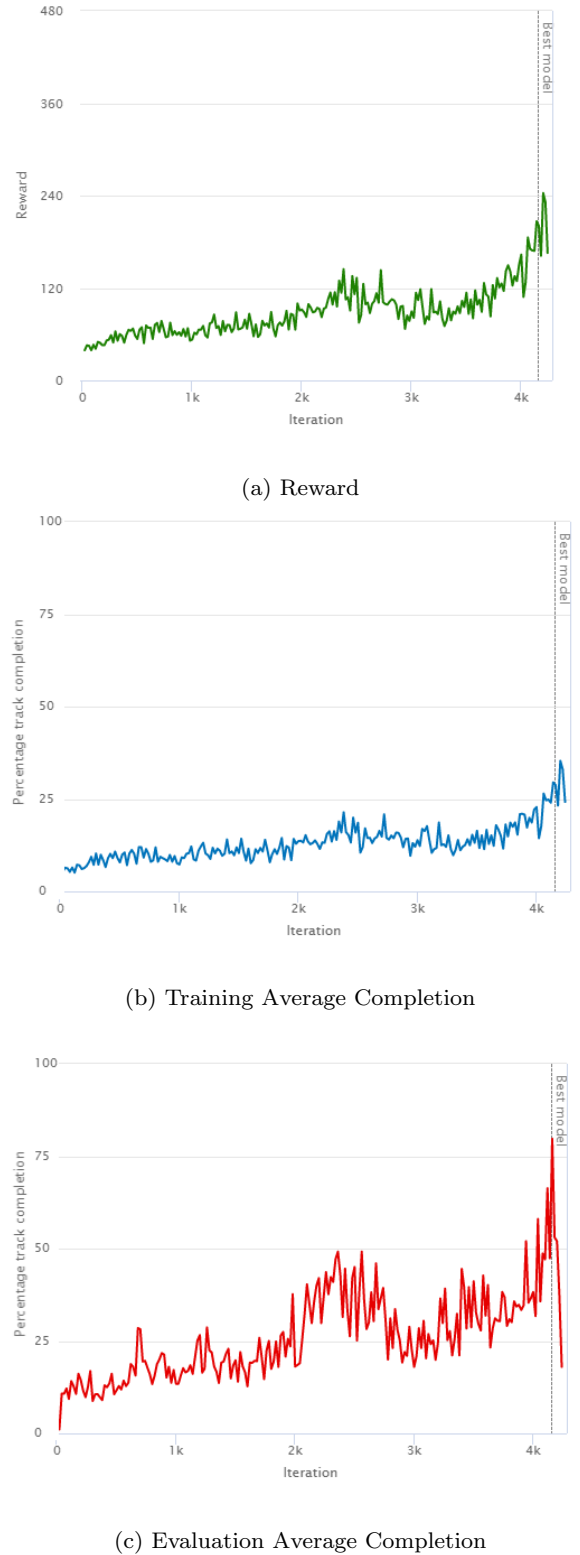


Fig. 13 Model 4 performance

Table 3 Hyperparameters

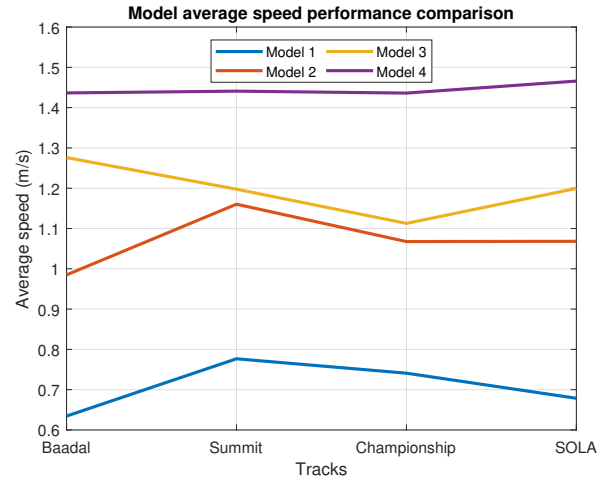
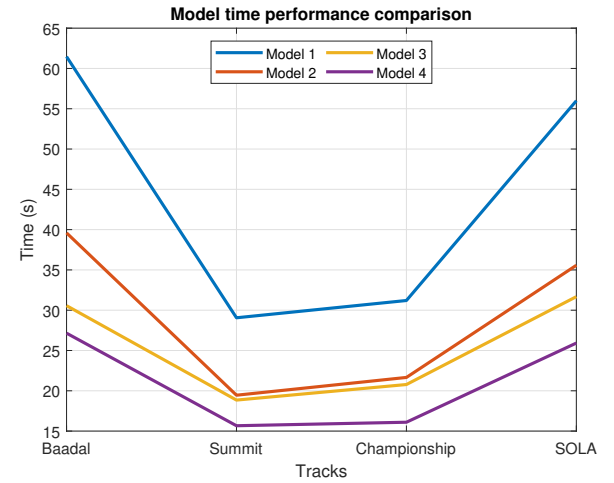
	Model 1	Model 2	Model 3	Model 4
Grad desc batch	64	64	64	64
Entropy	0.01	0.01	0.015	0.01
Discount fact	0.999	0.999	0.95	0.999
Loss Type	Huber	Huber	Huber	Huber
Learning rate	0.0003	0.0003	0.0003	0.0003
# ep update	20	20	20	20
# epochs	10	10	10	10

information on which behaviors get a higher reward and must be chosen.

The average speed of Model 1 is $0.7m/s$, Model 2 is $1.07m/s$, Model 3 is $1.2m/s$, and Model 4 is $1.45m/s$. In Model 2 the speed was 33% higher than in Model 1, while the speed in Model 3 was 11% faster than in Model 2, lastly the speed in Model 4 was 18% higher than Model 3. There is a clear increase in the speed in the second model as compared to the first. Figure 14 shows the speed performance of every on each track. This is due to the conservative approach of Model 1. The reason for the short speed difference between Model 2, 3 and 4 is the environment. The topography of each track limits the agent to its top speed. Model 1 is the benchmark used to determine the effectiveness and efficiency of the RL FIS integration. The times in which the tracks were completed are coherent and consistent. Tracks with similar lengths take approximately the same time. Model 2 improves the performance of Model 1. It navigates the Baadal Track 56% faster, the SOLA Speedway 57% faster, the Championship Track 44% faster and the AWS Summit Raceway 49% faster, showing a sustained improvement of almost 50% on every track. Model 3 performs better than Model 2. It completes the Baadal Track 30% faster, the SOLA Speedway 12% faster, the Championship Track 4% faster, and the AWS Summit Raceway 3% faster. Model 4 performs better than Model 3. It completes the Baadal Track 13% faster, the SOLA Speedway 22% faster, the Championship Track 29% faster, and the AWS Summit Raceway 24% faster. The performance on longer tracks is better because of the higher chances of finding straight segments of road. The improvement on the performance on short tracks was not as significant. The comparison of time performance of each model can be seen in Figure 15.

7 Conclusions and Future Work

This work implemented a FIS coupled with RL to create an explainable model behavior and that can be explained with representative examples. Both algorithms

**Fig. 14** Model speed performance comparison**Fig. 15** Models time performance comparison**Table 4** Model performance (SS.SSS)

	Model 1	Model 2	Model 3	Model 4
Baadal	61.471	39.596	30.564	27.149
Summit	29.060	19.452	18.847	15.664
Championship	31.204	21.657	20.778	16.099
SOLA	55.993	35.574	31.680	25.923

belong to the human-like (approximate reasoning) approach of AI. The FIS offers an open explanation of its behavior by closely analysing its linguistic variables and its knowledge base. As the fuzzy model follows the expertise of a group of specialists, it is relatively easy to understand. A relatively simple explanation should be enough to clarify how the inference system operates to an inexperienced user. However, this is highly dependent on the complexity of the system: the amount of rules, linguistic variables, and membership functions. Explanations coming from representative examples are

also accessible through a FIS. The aggregation and center of mass methods employed during the defuzzification process offer a direct explanation of the reward given to the agent. The output membership functions are triggered depending on the knowledge base to obtain a conclusion. The aggregated conclusion generates a waveform, which distribution is centered on the output linguistic variable. This way, whenever the agent is located on a given scenario an aggregation curve and a centroid are always available. Making easy to understand why a state produces certain action for the agent during the exploitation phase. These factors could be presented in a comprehensible and friendly visual or sound user interface to the occupants of an autonomous vehicle as an explanation of its navigation behavior. The user interface could be integrated into an advanced driver-assistance systems (ADAS). Constant feedback from the passengers can be used to update and create navigation preferences for different users. Trust and other feelings while using the self-driving system can be measured with non-invasive techniques aiming to improve the user experience.

The agents that followed the policies of Models 1 to 4 aimed to navigate on the center of the simulated track. This behavior is not optimal in a physical system. However, this paper aimed to demonstrate that the combination of a FIS and RL can achieve navigation on a closed racing track. The consistency of the models is directly related with their action space and the characteristics of the environment in which the agents navigate. The percentage of completed laps of models 1 and 2 in all the racing tracks per evaluation round in the AWSTM platform is above 80%. This rate drops to 50% in model 3 and to 20% in model 4. The action space for models 1 and 2 allowed them to navigate smoothly through all the courses. Nevertheless, the maximum speeds of models 3 and 4 make their consistency considerably drop. The first two models achieved a valid policy in few epochs because of the conservative approach of the reward FIS. As the same knowledge base was used for the other models, it did not grant enough situational knowledge to the agent. This caused their poor track completion rate. A better defined knowledge base is required to guarantee a constant performance on all models. The speed of the vehicle needs to be integrated into the rules. Although this will ensure navigation consistency, it will also make the policy more complex. External and non considered factors also influenced on the performance of the model. The training tracks had a minimum width of 106cm. The efficiency of the vehicle on narrower tracks decreased drastically. The policy of each model is not fit to navigate on reduced spaces. The current models should be retrained or new ones must be

developed to achieve successful navigation on the rest of the available tracks on the DeepRacerTM platform.

Further research can be done on explainable deep reinforcement learning fuzzy inference systems applied to autonomous navigation. A first alternative is to modify the presented work, such that the vehicle can navigate on the center of a lane. The highway driven scheme must be followed to simplify the development of this system. A more specific set of rules must be developed to remain inside a lane. Curves represent the biggest challenge as they are not banked. Another future work alternative is to increase the granularity of the steering angle. The action space of the agent would increase from nine possible actions to twenty one. The complexity of the system would require a more complex knowledge base and to tune the hyperparameters to train the model. The amount of epochs required to achieve a valid policy should increase to at least twice of the ones observed in this paper. A third option of potential developments is to use the AWSTM option to add obstacles to the track. This would require to equip the DeepRacerTM with either a stereo camera or a LIDAR to measure distance to the surrounding elements on the track. The agent should be able to navigate without colliding with objects located inside the track. The knowledge base should be upgraded and also the DNN used by the agent. A 5 layer network should be used because of the complexity of the required policies. The exploration phase of the agent should be extended to guarantee a better performance.

The developed models can also be tested in other simulated or physical platforms and to test how the system and its policies extendable are. The migration of to new platforms implies the consideration of other parameters as inertia, friction, and noise. The model requires coded information available on a Python dictionary. As long as it receives valid information it will produce an expected explainable action. Simulated platforms as CARLA SimulatorTM or the LGSVL SimulatorTM are used to validate autonomous driving models. The environment of both platforms can be modified so that it resembles the native of the AWS DeepRacerTM. Vehicles can be equipped with short and long range distance sensors, cameras, IMU's and ECU sensors. These capabilities along with ROS integration, allow the use of sensor fusion and estimation techniques. Physical platforms as the MIT RACECAR or F1/10 are an open source alternatives to the one offered by AWSTM. They use NVIDIATM GPUs to process the data coming from state-of-the-art sensors as stereo cameras, LIDAR, IMU and encoders. Intensive tests can be performed on this platforms in controlled environments. The complexity of the tests can be increased depending on the perfor-

mance of the policies. Suggested tests are situational driving, take overs, lane merge, intersections and multi agent interaction.

Acknowledgements The authors would like to thank Nora Clancy Kelsall for her English Language editing, and Dr. David Balderas-Silva and Dr. Renato Galluzzi review services. This research is being supported by the Laboratory of Computer Inteligente, Mechatronics and Biodesign (CIMB) at Tecnologico de Monterrey.

Funding

Funding was provided by Tecnologico de Monterrey - Grant No. A00996397, and Consejo Nacional de Ciencia y Tecnologia (CONACYT) by the scholarship 679120.

The authors would like to acknowledge the financial support of the Novus Grant with PEP no. PHHT032-19ZZ00013, TecLabs, Tecnologico de Monterrey, in the production of this work.

Code availability

All the versions of the Deep Reinforcement Learning Fuzzy Inference System are located in this Github repository Rolix57/RL-FIS.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. M. Montemerlo, S. Thrun, H. Dahlkamp, D. Stavens, and S. Strohband., Winning the DARPA Grand Challenge with an AI robot, Proceedings of the AAAI National Conference on Artificial Intelligence, Boston, MA, (2006). AAAI.
2. Chris Urmson, et al, Tartan Racing: A Multi-Modal Approach to the DARPA Urban Challenge, DARPA, (2007)
3. SAE International J3016. Accessed: April 12, 2020. [Online]. Available: <https://www.sae.org/standards/content/j3016.201401/>
4. S. Veres, L. Molnar, N. Lincoln, P. Morice, Autonomous vehicle control systems—A review of decision making, *Proc. Inst. Mech. Eng. I, J. Syst. Control Eng.*, vol. 225, no. 2, pp. 155–195 (2011)
5. M. Althoff, R. Lösch, Can automated road vehicles harmonize with traffic flow while guaranteeing a safe distance? in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pp. 485–491 (2016)
6. C. Vagg, C. J. Brace, D. Hari, S. Akehurst, J. Poxon, L. Ash, Development and field trial of a driver assistance system to encourage eco-driving in light commercial vehicle fleets, *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 796–805, (2013)
7. H. Andersen et al., Trajectory optimization for autonomous overtaking with visibility maximization, in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pp. 1–8. (2017)
8. K. Zhang, A. Yang, H. Su, A. de La Fortelle, K. Miao, Y. Yao, Service-oriented cooperation models and mechanisms for heterogeneous driverless vehicles at continuous static critical sections, *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1867–1881, (2016)
9. C. Menéndez-Romero, M. Sezer, F. Winkler, C. Dornhege, W. Burgard, Courtesy behavior for highly automated vehicles on highway interchanges, in *Proc. IEEE Intell. Vehicles Symp. (IV)*, pp. 943–948 (2018)
10. L. Li, D. Wen, Y. Yao, A survey of traffic control with vehicular communications, *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 425–432, Feb. (2014)
11. P. Morignot, J. P. Rastelli, and F. Nashashibi, Arbitration for balancing control between the driver and ADAS systems in an automated vehicle: Survey and approach,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, pp. 575–580 (2014)
12. A. Broggi, S. Debattisti, M. Panciroli, P. Porta, Moving from analog to digital driving, in *Proc. IEEE Intell. Vehicles Symp. (IV)*, pp. 1113–1118 (2013)
13. M. Althoff, M. Koschi, and S. Manzingier, CommonRoad: Composable benchmarks for motion planning on roads, in *Proc. IEEE Intell. Vehicles Symp. (IV)*, pp. 719–726, (2017)
14. P. Morignot, F. Nashashibi, An ontology-based approach to relax traffic regulation for autonomous vehicle assistance, in *Proc. IASTED Int. Conf. Artif. Intell. Appl.*, pp. 10–17 (2013)
15. N. Du, F. Zhou, E. Pulver, D. Tilbury, L. Robert, A. Pradhan, X Yang, Examining the Effects of Emotional Valence and Arousal on Takeover Performance in Conditionally Automated Driving, *Transportation Research Part C: Emerging Technologies*, (pdf), 112, pp. 78-87 (2020)
16. S. Jayaraman, C. Chandler, D. Tilbury, X. Yang, A. Pradhan, K. Tsui, L. Robert, Pedestrian Trust in Automated Vehicles: Role of Traffic Signal and AV Driving Behavior, *Frontiers in Robotics and AI*, 6(117) (2019)
17. G. Vasiljević, D. Miklič, I. Draganjac, Z. Kovačić, P. Lista, High-accuracy vehicle localization for autonomous warehousing, *Robotics and Computer-Integrated Manufacturing*, Volume 42, Pages 1-16 (2016)
18. F. Schneemann, I. Gohl, Analyzing driver-pedestrian interaction at crosswalks: A contribution to autonomous driving in urban environments, *IEEE Intelligent Vehicles Symposium (IV)*, Gothenburg, pp. 38-43. (2016)
19. L. Claussmann, M. Revilloud, S. Glaser, D. Gruyer, A study on al-based approaches for high-level decision making in highway autonomous driving, in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, pp. 3671–3676, (2017)
20. L. Claussmann, M. Revilloud, D. Gruyer, S. Glaser, A Review of Motion Planning for Highway Autonomous Driving, *IEEE Transactions On Intelligent Transportation Systems* (2019)
21. E. Balal, R. L. Cheu, T. Sarkodie-Gyan, A binary decision model for discretionary lane changing move based on fuzzy inference system, *Transp. Res. C, Emerg. Technol.*, vol. 67, pp. 47–61, (2016)
22. S. Lefèvre, A. Carvalho, F. Borrelli, A learning-based framework for velocity control in autonomous driving, *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 1, pp. 32–42, (2016)
23. N. Li, D. W. Oyler, M. Zhang, Y. Yildiz, I. Kolmanovsky, R. Girard, Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems, *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 5, pp. 1782–1797, (2018)

24. Q. Huy, S. Mita, H. T. N. Nejad, L. Han, Dynamic and safe path planning based on support vector machine among multi moving obstacles for autonomous vehicles, *IEICE Trans. Inf. Syst.*, vol. E96-D, no. 2, pp. 314–328 (2013)
25. S. Lefèvre, D. Vasquez, C. Laugier, A survey on motion prediction and risk assessment for intelligent vehicles, *ROBOMECH J.*, vol. 1, no. 1, pp. 1–14 (2014)
26. A. Constantin, J. Park, K. Iagnemma, A margin-based approach to threat assessment for autonomous highway navigation, in *Proc. IEEE Intell. Vehicles Symp. (IV)*, pp. 234–239 (2014)
27. M. Ardeit, P. Waldmann, F. Homm, N. Kaempchen, Strategic decision-making process in advanced driver assistance systems, *IFAC Proc. Volumes*, vol. 43, no. 7, pp. 566–571 (2010)
28. C. Chen, A. Seff, A. Kornhauser, and J. Xiao, Deep-Driving: Learning affordance for direct perception in autonomous driving, in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2722–2730 (2015)
29. L. Yang, X. Liang, T. Wang, E. Xing, Real-to-virtual domain unification for end-to-end autonomous driving, in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 530–545 (2018)
30. M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, K. Zieba, End to End Learning for Self-Driving Cars, (2016)
31. W. Chen, T. Qu, Y. Zhou, K. Weng, G. Wang and G. Fu, Door recognition and deep learning algorithm for visual based robot navigation, *IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, Bali, pp. 1793–1798 (2014)
32. Y. Zhu et al., "Target-driven visual navigation in indoor scenes using deep reinforcement learning," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, pp. 3357–3364. (2017)
33. C. Richter, R. Nicholas, Safe Visual Navigation via Deep Learning and Novelty Detection, *Robotics: Science and Systems XIII* (2017).
34. E. Kaufmann, A. Loquercio, R. Ranftl, A. Dosovitskiy, V. Koltun, D. Scaramuzza, Deep Drone Racing: Learning Agile Flight in Dynamic Environments, *Conference on Robot Learning (CORL)*, Zurich (2018)
35. S. Jung, S. Hwang, H. Shin, and D. H. Shim, "Perception, Guidance, and Navigation for Indoor Autonomous Drone Racing Using Deep Learning," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2539–2544, (2018)
36. Z. Shou, X. Di, Reward Design for Driver Repositioning Using Multi-Agent Reinforcement Learning (2020)
37. W. Samek, K. Müller, Towards Explainable Artificial Intelligence, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, Springer International Publishing (2019)
38. S. Han, J. Pool, J. Tran, W. Dally, Learning both weights and connections for efficient neural network, *Advances in Neural Information Processing Systems (NIPS)*, pp. 1135–1143 (2015)
39. A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, *International Conference on Learning Representations (ICLR)*. (2018)
40. W. Samek, T. Wiegand, K. Müller, Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models, *ITU Journal: ICT Discoveries - Special Issue 1 - The Impact of Artificial Intelligence (AI) on Communication Networks and Services 1(1)*, 39–48 (2018)
41. B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, R. Sayres, Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV), *International Conference on Machine Learning (ICML)*, pp. 2673–2682, (2018)
42. G. Montavon, W. Samek, K. Müller, Methods for interpreting and understanding deep neural networks. *Digital Signal Processing* 73, 1–15 (2018)
43. S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, K. Müller, Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications* 10, 1096 (2019)
44. P. Koh, P. Liang, Understanding black-box predictions via influence functions, *International Conference on Machine Learning (ICML)*. pp. 1885–1894 (2017)
45. A. Adadi, M. Berrada, Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI), *IEEE Access*, vol. 6, pp. 52138–52160 (2018)
46. J. Haspiel, N. Du, J. Meyerson, L. Robert, D. Tilbury, X. Yang, A. Pradhan, Explanations and Expectations: Trust Building in Automated Vehicles, *IEEE International Conference on Human-Robot Interaction* March, pp 119–12 (2018)
47. A. Holzinger, C. Biemann, C. Pattichis, D. Kell, What do we need to build explainable AI systems for the medical domain?, *Explainable AI for the Medical Domain* (2017)
48. K. Amarasinghe, K. Kenney, M. Manic, Toward Explainable Deep Neural Network Based Anomaly Detection, *11th International Conference on Human System Interaction (HSI)*, Gdansk, pp. 311–317 (2018)
49. A. Fernandez, F. Herrera, O. Cordon, M. Jose del Jesus and F. Marcelloni, Evolutionary Fuzzy Systems for Explainable Artificial Intelligence: Why, When, What for, and Where to?, *IEEE Computational Intelligence Magazine*, vol. 14, no. 1, pp. 69–81, (2019)
50. J. Fürnkranz, D. Gamberger, and N. Lavrac, *Foundations of Rule Learning*. New York, N Y, U SA: Springer, 2012
51. L. Kuncheva, How good are fuzzy if-then classifiers?, *IEEE Trans. Syst. Man, Cybern. B*, vol. 30, no. 4, pp. 501–509, (2000)
52. C. Mencar, J. Alonso, Paving The Way to Explainable Artificial Intelligence With Fuzzy Modelling, *Fuzzy Logic and Applications: 12th International Workshop*, pp. 215–226 (2018)
53. E. Morales-Vargas, C. Reyes-García, H. Peregrina-Barreto, F. Orihuela-Espina, Facial Expression Recognition With Fuzzy Explainable Models, *Models and Analysis of Vocal Emissions for Biomedical Applications: 10th International Workshop*, (2017)
54. B. Keneni et al., "Evolving Rule-Based Explainable Artificial Intelligence for Unmanned Aerial Vehicles," in *IEEE Access*, vol. 7, pp. 17001–17016 (2019)
55. Y. Deng, Z. Ren, Y. Kong, F. Bao, Q. Dai, A Hierarchical Fused Fuzzy Deep Neural Network for Data Classification, *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 4, pp. 1006–1012 (2017)
56. C. Lee and C. Teng, Identification and control of dynamic systems using recurrent fuzzy neural networks, *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 4, pp. 349–366, (2000)
57. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature*. 521 (7553): 436–444 (May 2015).
58. L. Deng, D. Yu, Deep Learning: Methods and Applications, *Foundations and Trends, in Signal Processing Vol. 7, Nos. 3–4* (2013)
59. S. Russel, P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd Edition, 2010

60. L.A. Zadeh, Fuzzy sets, Information and Control, Volume 8, Issue 3, June 1965, Pages 338-353
61. L.A. Zadeh, Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, IEEE Transactions on Systems, Man, and Cybernetics, Volume: SMC-3, Issue: 1, Jan. 1973
62. P. Ponce-Cruz, F. Ramírez-Figueroa, Intelligent Control Systems with LabVIEW, Springer, 2010
63. P. Ponce-Cruz, Inteligencia Artificial con Aplicaciones a la Ingeniería, Editorial Alfaomega, 2011
64. R. Knapp, U. Agarwal, R. Djamschidi, S. Layeghi, M. Dastamalchi, The Use of Fuzzy Set Classification for Pattern Recognition of the Polygraph, IEEE 3rd International Fuzzy Systems Conference, 1995
65. D. Driankov, A. Saffiotti, Fuzzy Logic in Autonomous Navigation, Springer-Verlag, Berlin Heidelberg (2001)
66. D. Wu, Twelve Considerations in Choosing between Gaussian and Trapezoidal Membership Functions in Interval Type-2 Fuzzy Logic Controllers, IEEE International Conference on Fuzzy Systems, June 2012
67. AWS DeepRacer Developer Guide, 2020 Amazon Web Services, Inc., PDF Accessed: April 12, 2020. [Online]. Available: <https://docs.aws.amazon.com/deepracer/latest/developerguide/awsracerdg.pdf>