electronics

MDPI

Article

Autonomous Driving in Roundabout Maneuvers Using Reinforcement Learning with Q-Learning

Laura García Cuenca *, Enrique Puertas, Javier Fernandez Andrés and Nourdine Aliane

School of Architecture, Engineering and Design, Universidad Europea de Madrid, Tajo s/n, Villaviciosa de Odón, 28670 Madrid, Spain; enrique.puertas@universidadeuropea.es (E.P.); javier.fernandez@universidadeuropea.es (J.F.A.); nourdine.aliane@universidadeuropea.es (N.A.)

* Correspondence: laura.garcia@universidadeuropea.es; Tel.: +34-91-211-56-19

Received: 1 November 2019; Accepted: 10 December 2019; Published: 13 December 2019

Abstract: Navigating roundabouts is a complex driving scenario for both manual and autonomous vehicles. This paper proposes an approach based on the use of the Q-learning algorithm to train an autonomous vehicle agent to learn how to appropriately navigate roundabouts. The proposed learning algorithm is implemented using the CARLA simulation environment. Several simulations are performed to train the algorithm in two scenarios: navigating a roundabout with and without surrounding traffic. The results illustrate that the Q-learning-algorithm-based vehicle agent is able to learn smooth and efficient driving to perform maneuvers within roundabouts.

Keywords: reinforcement learning; Q-learning; autonomous driving; roundabouts; machine learning; simulation environment; driving behavior; environment perception

1. Introduction

One of the most challenging problems for autonomous vehicles is complex maneuvering, such as driving in roundabouts in urban and nonurban environments. Roundabouts are a special case of intersection, where a circular traffic flow is established for a change of direction. To navigate successfully in a roundabout, it is necessary to understand the choice of entry and exit lanes, how to apply priority rules, how to interpret the intentions of other drivers, and the existing traffic itself. However, for the correct selection of actions in this particular scenario, a global understanding of the situation of driving in roundabouts is necessary to obtain the best results.

One approach to understanding driving in roundabouts is through artificial intelligence and data mining techniques such as machine learning. For example, in [1] the authors presented rules of behavior to address a roundabout with an autonomous vehicle, modeling the behavior of a human driver through factors such as the speed of the vehicle, the angle of the wheel, the diameter of the roundabout, etc. The authors of [2] presented an adaptive tactical behavior planner (ATBP) for an autonomous vehicle, capable of planning behaviors similar to human drivers when navigating a roundabout. Roundabout safety under shared traffic was studied in [3] through models based on speed and traffic. The authors of [4] presented learning techniques to obtain behaviors of human drivers when approaching a roundabout without signs posted, in order to obtain behavioral profiles applicable to autonomous vehicles. Applying machine learning techniques such as support vector machine (SVM), the authors of [5] presented a prediction model to obtain the vehicle's intention to enter or exit a roundabout. In this study, variables such as vehicle global positioning system (GPS) and multiple sensors were used to obtain directions on the vehicle's path. Another approach using machine learning techniques is presented in [6], where the authors designed a roundabout driving classification using hidden Markov models (HMMs) trained with naturalistic driving data, while the authors of [7] proposed a sequential adaptive reinforcement learning approach for roundabout driving.

As reviewed in the previous literature, methods that use supervised learning techniques label the information and predict more or less accurately the variables to be treated in the problem of autonomous driving in a roundabout. With this methodology, an autonomous vehicle can maintain the course in a roundabout but does not maintain the direction autonomously, and could collide with some obstacle on the road or another vehicle by not having the correct orientation. On the other hand, unsupervised learning methods are not useful for addressing an automatic decision-making problem, because they only group data and it is not the problem to be addressed. If the objective is to autonomously execute the driving of a vehicle in a roundabout, a success rate of around 90% must be obtained in the actions that the expert system of the vehicle must execute to take a correct route. When applying reinforcement learning, the autonomous vehicle is oriented through the training system to make the best decisions, with reward policies applied that penalize the expert system itself when its actions are not correct, until it reaches a very high success rate. That is why learning reinforcement techniques are currently being used in autonomous driving research. Through these techniques, autonomous vehicles can learn to act in environments with uncertainty [8].

In this paper, a Markov decision process (MDP) is used for planning the study of the behavior of an autonomous vehicle to safely navigate a roundabout with the Q-learning algorithm in a simulation environment. To this end, a set of naturalistic driving data was used for speed information and a machine learning model algorithm to learn decision-making. The main contribution of this paper is the design of a tangible learning technique for sequential and automatic decision-making of autonomous vehicles through examples in a simulated environment with roundabout scenarios without traffic, mainly for learning lane tracking, and with traffic, to learn the right maneuvers for approaching, circumnavigating, and exiting the roundabout safely. The designed learning system is able to learn enough to perform optimal driving, deciding whether an action taken has been positive or negative by reinforcing it through the defined rewards policy. The CARLA simulation environment [9] was used to develop, train, and evaluate the proposed approach within two scenarios: navigating a roundabout with and without traffic.

The remainder of the paper is organized as follows: Section 2 presents an overview of the reinforcement learning system's framework and its use in autonomous driving. Section 3 presents the simulation environment. Section 4 gives an overview of the Q-learning algorithm and how it is used to model driving situations, as well as a description of the training policies. Section 5 presents some simulations and experimental results. Finally, conclusions and future work of the present research are given in Section 6.

2. Reinforcement Learning Background

When training a machine learning model, three types of learning can be used, depending on the task to be performed: supervised learning, used mainly for classification and prediction tasks; unsupervised learning, which is suitable for clustering and finding relationships among attributes of data; and reinforcement learning, which creates models to learn patterns by trial and error. The latter is the algorithm used in the present work.

Reinforcement learning (RL), according to [10], is a machine learning technique that defines how a set of sequential decisions will result in the achievement of a goal. This is considered to be a trial-and-error method, where the environment indicates the usefulness of the result. According to the authors of [11], in their experimentation, they considered that RL is a branch of artificial intelligence in which an agent learns a control strategy when interacting with the environment. In the same way, the authors of [12] considered that RL is capable of learning and making decisions by interacting repeatedly with its environment. Currently, several machine learning algorithms use the reinforcement learning paradigm as the basis for implementation, such as adaptive heuristic critic (AHC) [13], Q-learning [14], Markov decision process [15], and deep Q-learning [16]. RL is currently applied in several areas, such as computer networks [17], traffic control [18], robotics [19], object recognition [20], facial recognition [21], and autonomous driving [22], among the most prominent areas of research.

As far as RL algorithms applied to autonomous driving, many studies can be found in the literature. For example, in [23] a successful application of RL is described for autonomous helicopter flight, while in [24], the authors describe an experiment in RL to direct a real robot car based on data collected directly from real experiments. In [25], the ability of a system to learn behaviors to allow safe navigation for autonomous driving at intersections is explored.

RL is also used in simulated environments, and multiple examples can be found. For example, in [11,26,27] various RL methods are proposed in which autonomous vehicles learn to make decisions while interacting with simulated traffic. In [28,29] the authors present different methods for deep end-to-end RL to navigate autonomous vehicles, and in [30] a speed control system is designed using RL.

In this paper, the system that was developed is based on the RL paradigm. The system's framework consists of an agent (the autonomous vehicle) that interacts with a driving simulation environment, a finite state space (S), a set of available actions (A), and a reward function (R), where the main agent's task is to find the policy π: S × A → [0, 1]. According to [31], the general framework is based on interactions between agents, where the environment is characterized by a set of states, in which the agents can achieve actions of the set itself. The agents interact with the environment and transition from state $x^{(t)} = x$ to $x^{(t+1)} = x$ by selecting an action $a^{(t)} = a$. The interactions between the agent and the environment come from the agent's observations about the state of the environment, where it selects an action and finally receives feedback or reward from the environment according to the selected action. That is, when the agent observes the state of the environment $x^{(t)}$ at time (t), it selects an action and makes a transition to state $x^{(t+1)}$ at time (t + 1). Subsequently, the environment issues a reward $r^{(t+1)}$ for the agent. Figure 1 shows the general agent–environment interaction system.

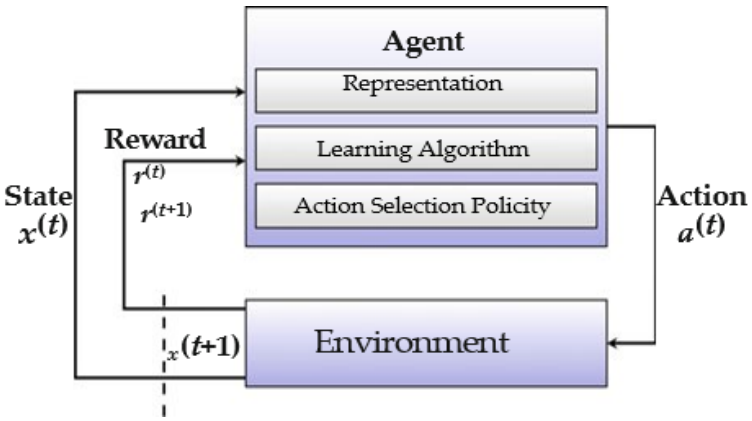


Figure 1. Reinforcement learning paradigm for the agent–environment interaction system framework.

3. CARLA Simulation Environment

CARLA is a simulation environment for autonomous driving systems. It consists of open-source code and protocols that provide digital assets, such as urban layouts, buildings, and vehicles, used to corroborate and evaluate decisions and design approaches in driving simulations. It follows the client–server paradigm. CARLA supports the configuration of various sensor sets and provides signals to train driving strategies, such as GPS coordinates, acceleration, steering wheel, etc. It also provides information related to distance travelled, collisions, and the occurrence of infractions, such as drifting into the opposite lane or onto the sidewalk. The environment model consists of a simulated autonomous driving system where driving in a roundabout is defined. According to the direct perception approach, dimensional video data (red, green, blue (RGB) camera, semantic segmentation camera, depth camera, and object vision detection) and a set of naturalistic conduction data are processed into meaningful data about roads. The agent model consists of an autonomous vehicle that interacts with the simulated environment through different actions (acceleration, braking, determining roundabout diameter,

adjusting vehicle speed, determining roundabout center, starting and ending the route, determining deviation angle from the center of the lane). It learns from the feedback resulting from the GPS position of the vehicle at stage $x^{(t)}$ at time (t) in the simulated environment, using the Q-learning algorithm and a reward system. Figure 2 shows the architecture of our system, and Figure 3 provides maps and some views of the simulation environment used. It includes various urban scenarios, including a roundabout. The range of the map is 600 m × 600 m, containing a total of around 5 km of road.

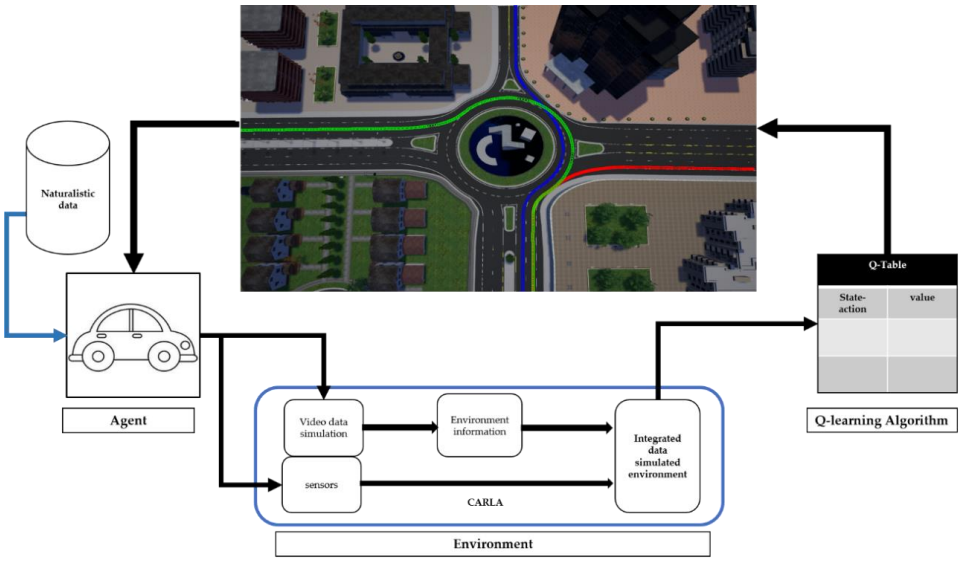
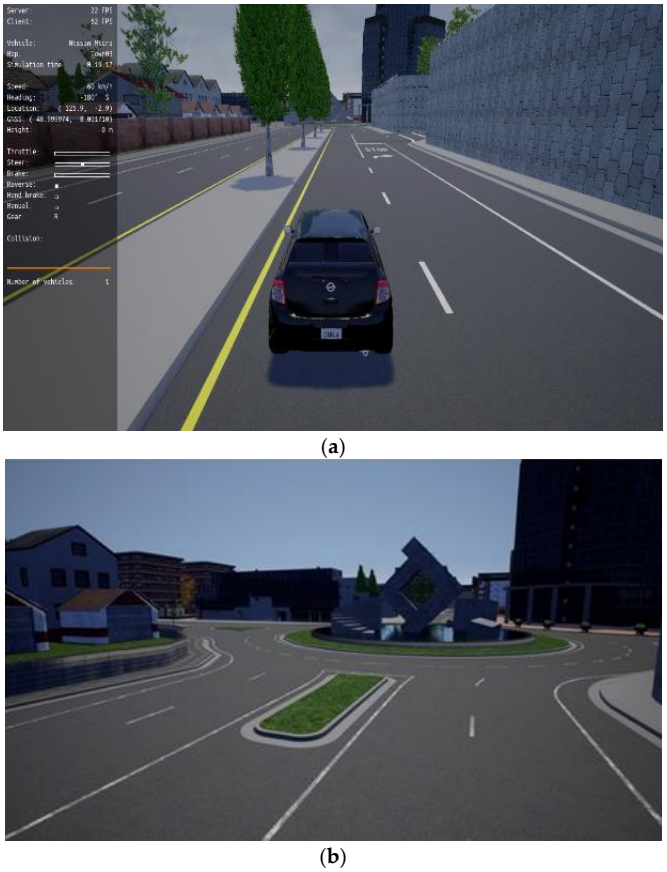


Figure 2. Architecture of the specific system for the task of autonomous driving planning for safely navigating a roundabout in a simulation environment.



(**a**)

(**b**)

Figure 3. CARLA: (**a**) Maps and information sensors; (**b**) view of the urban environment.

The autonomous vehicle is controlled by different types of commands in the simulation environment: (1) Steering: The steering wheel angle is represented by a real number between −40° and +40°, which correspond to full left and full right, respectively. (2,3) Throttle, brake: These are represented by real numbers between 0 and 1. (4) Hand brake: A Boolean value is used to indicate whether the hand brake is activated or not. The data acquisition system includes (1) an RGB camera, equipped with semantic segmentation of the simulation environment and including 3D location and orientation with respect to the car's coordinate system; (2) a semantic segmentation pseudo-sensor camera, providing support for experiments of perception; and (3) a sensor providing GPS position and information on roads, lane markings, traffic signs, sidewalks, fences, poles, walls, buildings, vegetation, vehicles, pedestrians, etc. In addition to the observations and actions, information such as the diameter and center of the roundabout, the start and end of the route established for the roundabout, and the angle of deviation from the center of the lane are recorded.

4. Machine Learning Model

This section describes a CARLA environment for planning the behavior of a vehicle to navigate a roundabout using the Q-learning algorithm. In addition, a naturalistic driving dataset is used to provide contextual information. This section explains the concepts of a roundabout scenario and the application of the Q-learning algorithm in this context as well as the reward policy.

4.1. Roundabout Scenario

Roundabouts present specific challenges in the complexity of driving behavior, in terms of high variance in the number of lanes and increased uncertainty in perception due to the road geometry. It is crucial that autonomous vehicles exhibit natural behavior on roundabouts for the safety and smooth flow of shared traffic between them and manual vehicles [32]. The approach followed in this paper is to formulate the driving task within a roundabout as a Markov decision process (MDP) problem. The experiments performed were based on simulated and real data.

The roundabout shape used in the experiment is shown in Figure 4, where the exits are marked as A for the first one, B for the second, and so forth. Typical vehicle behavior in a roundabout consists of the following steps: approach, cross the roundabout, and exit. The possible driving paths are drawn in different colors with their centerlines. In the CARLA framework, a route is defined by the tuple {Start_point, End_point}, as follows:

- Exit A, GPS route {vehicle position, A};
- Exit B, GPS route {vehicle position, B};
- Exit C, GPS route {vehicle position, C}.

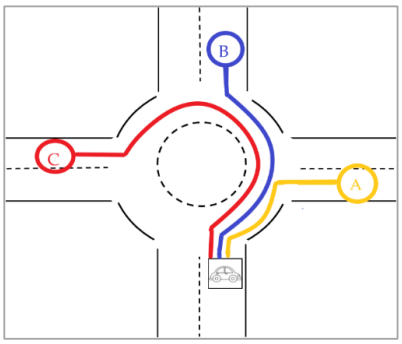


Figure 4. Roundabout trajectories in the experiment; A, B, and C are driving paths.

The final objective is to determine the behavior strategy so that the autonomous vehicle will enter the roundabout and navigate the exits (A, B, C) correctly and safely, regardless of whether or not there are other vehicles on the road.

4.2. Q-Learning Algorithm

Based on RL, an MDP was modeled for the task of planning the behavior of a vehicle to safely navigate a roundabout on paths A, B, and C. The adaptative model uses a Q-learning algorithm [14], a commonly used algorithm to solve Markov decision processes.

In Q-learning, the action value function Q^{π}(s, a) is the expected return E $[R_t|s_t$ = s, a_t = a] for a state–action pair following a policy π, where R_t = reward, s_t = state, and a_t = action. Given an optimal value function Q^{π}(s, a), the optimal policy can be inferred by selecting the action with maximum value max a Q^{π}(s, a) at every time step. It is based on finding a function Q(s, a) toward an estimation of the function value (Q-value). The function Q(s, a) represents the utility of taking action a in state s. Given the function Q(s, a), the optimal policy is the one that selects for each state the action associated with the highest expected accumulated value (Q-value). The function Q(s, a) is updated using the following equation for adjusting temporal differences:

$$Q(s_t,\, a_t) = Q(s_t,\, a_t) + \alpha(r_{t+1} + Qmax(s_{t+1},\, a) - Q(s_t,\, a_t)).$$

This equation adjusts Q(s, a) based on the current and predicted reward if all subsequent decisions were optimal. In this sense, the function Q(s, a) converges toward the optimal values of the function. The machine learning model can use the Q-values to evaluate each decision that is possible in each state. The decision that returns the highest Q-value is the optimum. The whole procedure of model operation based on the Q-learning algorithm for this paper is shown in Algorithm 1 in Appendix A.

The Q-value derived from the performance of an action is the sum of the immediate reward provided by the environment and the maximum value of Q for the new state reached. The transition to the next state is defined by function T, affected by parameter g, referred to as the discount factor. Formally,

$$s_{t+1} \leftarrow T(\, s_t\, ,\, a_t\,);\; Q(s_t,\, a_t) \;=\; r_{t+1} \;+\gamma Q_{max}\,(s_{t+1}\,);\quad 0 \le \gamma \le 1,$$

where the values of Q would be updated using the following:

$$Q(s_t,\, a_t) = Q(s_t,\, a_t) + \alpha(r_{t+1} + \gamma Qmax(s_{t+1},\, a) - Q(s_t,\, a_t)),$$

with $0 \le \alpha,\, \beta \le 1$. The learning mechanism is set using parameter α. For example, if α = 1, the new value of Q(s, a) does not take into account the previous history of the value of Q, but will be the direct reward added to the maximum value of Q for the new state corrected by the γ factor.

In the presented algorithm, the values of the Q function are modified and are organized as a table with information about the new states and actions being explored. Thus, each row corresponds to a different state, and each column stores information about the value of the actions. Specifically, element (i, j) of the table represents the value of performing from state s_i if the action is a_j. Table 1 is a Q-table, obtained by implementing in any of the total states acquired by the algorithm given in Appendix A, for example, driving through exit A.

Table 1. Q-table: state ({deviate_angle: possible steering wheel turn rewards}).

State	Value Action1	Value Action 2	Value Action 3	Value Action 4	Value Action 5
s_1	{−0.39: −0.163}	{0.13: −0.1149}	{−0.27: −0.1369}	{−0.2: −0.125}	{0.38: −0.1612}
s_2	{0.09: −1.111}	{0.24: −0.416}	{−0.24: −0.416}	{0.0: −1.0}	{−0.22: −0.454}
s_3	{0.13: −0.769}	{−0.25: −0.133}	{0.23: −0.129}	{0.12: −0.833}	{−0.13: −0.769}
s_4	{0.29: −0.344}	{−0.29: −0.140}	{−0.14: 0.116}	{−0.08: 0.108}	{0.21: −0.126}
s_5	{0.17: −0.120}	{−0.05: 0.105}	{−0.12: 0.113}	{−0.18: 0.121}	{−0.39: −0.163}

This Q-table grows rapidly by having to store all the state–action combinations. That is why only some states of the experiment without traffic are shown. In an automatic decision-making problem

such as the one presented in this paper, the number of possible states can be overwhelming, making it very expensive to collect all the experiments and their updates, and making the problem unmanageable from the computational point of view.

4.3. State–Space Training Model

In the Markov decision process context, the state space can be defined as follows:

$$s_{veh} = \left\{ x,\ y,\ v_x\ ,\ v_y\ ,\ r_p\ ,\ lane_i,\ lane_d \right\}$$

where x, y, v_x, v_y are the GPS position and velocity vector components along the x, y axes of the vehicle and r_p is the distance to the next GPS position within the route to exit A, B, or C. The coordinate transformation that aligns the vehicle position and velocity along the vehicle axis makes the vehicle state invariant to road geometry. $lane_l$, $lane_r$ are binary values; the value is 1 if a left or right lane exists with respect to the center of the lane.

To manage with traffic, the simulated vehicle uses a perception system that detects and tracks other participants through different sensors, as explained in Section 3. The vehicle's trajectory control within a roundabout is achieved through the speed and wheel angle. The concept of defining the trajectory of the autonomous vehicle is based on the optimal control approach presented in [33] and successfully implemented in [34]. The viability of planned trajectories is guaranteed by imposing real predictions on the speed [1], acceleration, and exit of the roundabout.

The training model is used through a learning approach, where 70% of the actions are applied randomly (exploration model) and 30% of the remaining actions are based on actions already learned (exploitation model). That is why the Q-learning algorithm is used, based on the experience generated during the exploration of the environment for the training model. Each model is first trained without any other vehicles, with the goal of learning the optimal policy, and then is retrained with other vehicles using random initialization. The training model is based on the vehicle's GPS positions with respect to the center of the lane. The model is copied to the main network every 10,000 iterations. Deviation from the center of the lane, as depicted in Figure 5, is calculated as follows:

- Vehicle position: Current vehicle position;
- Previous position: Previous vehicle position closest to the center of the lane;
- Next position: Ten positions forward from the center of the lane;
- Vector L: Vector formed by the vehicle position and its previous position,

$$Vl = V[(x_a - x_b),\ (y_a - y_b)];$$

- Vector J: Vector formed by the previous and next vehicle positions,

$$Vj = V[(x_b - x_c),\ (y_b - y_c)].$$

After defining the parameters, the angle of deviation from the center of the lane, in degrees, is calculated according to

$$\alpha = \ arcos\left(57,2958 \frac{(VL_X \cdot\ Vj_x) + \left(Vl_y \cdot VJ_y\right)}{|L| \cdot |J|} \right).$$

4.4. Reward Policy

In Q-learning, the reward policy acts as an objective function from an optimization problem point of view. For the current scenario, human behavior was taken into account, where the objective is to navigate safely and efficiently in a roundabout without impeding the flow of traffic. A system of double rewards was used, depending on the current state of the vehicle:

- **Reward 1.** The vehicle travels on the road along the center of the lane. In this case, smooth turns would be made to correct the position of the line. The implemented reward system has the goal of rewarding smooth turns of the steering wheel over sharp ones. Considering a range for the degrees of angle deviation, the reward function is inversely proportional to the deviation angle: $Reward = \frac{1}{\alpha}$.
- **Reward 2.** When the vehicle is instead traveling along the lane and deviates following an exit path of the roundabout, sharper turns are more suitable, and are rewarded in those cases where the steering wheel turns back to the center of the lane: $Reward = \frac{1}{(1-\alpha)}$.

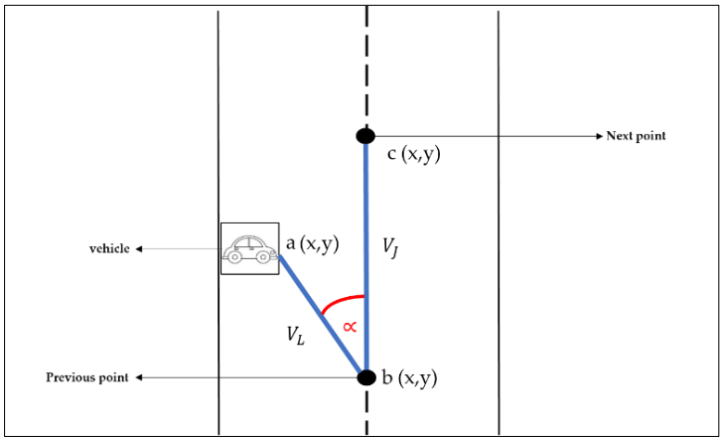


Figure 5. Calculation of deviation from the center of the lane.

5. Experimental Results

In this section, two experiments are described: the first one deals with entry and exit A without traffic, whereas the second one deals with entry and exits B/C with traffic. In the simulation the following nomenclature was used.

The symbolic description Δ was used as follows: $\Delta = \{e, vm, mr, desv, dmr\}$. The target vector metrics contain the exit (e), the average speed (vm) for a given distance within the defined route, the efficiency of the reward system (mr), the average deviation (desv), and the average distance traveled (dmr) for each attempt by the vehicle to satisfactorily exit from the roundabout. The action–state cycle is repeated until the vehicle reaches the correct exit. If the vehicle ends up crashing or crossing one of the bounding lanes while in the action–state cycle, the training program is interrupted and the vehicle starts again. In traffic scenario (B, C), the action–state cycle is discretized through the reward function, which depends on the positions of other vehicles inside the roundabout. This discretization is carried out under techniques based on the nearest neighbor, and is used to provide a simple way to divide the state space into regions. The vehicle updates the trajectory information every 60 seconds, sends it to the training program, and takes the appropriate action specified by the training program. Figure 6 shows the two scenarios established in the experiment, where the test vehicle is in red.

The behavior was tested in the same scenario in which the vehicle was trained. The number of participating vehicles and their routes were fixed for a given experiment, but their initial positions varied randomly. The training dataset consisted of 96 hours of driving the vehicle manually within the simulation environment. The route included four roundabouts with three exits, A, B, and C. During the learning phase, the trained dataset was evaluated after 100 iterations using the trajectory examples. The agent was trained for a total of 10,000 episodes, with each lasting for 100 samples or until a collision occurred. For training, roundabouts with no traffic were considered, and the speed of the vehicle was set up according to the predictive model obtained in [1].

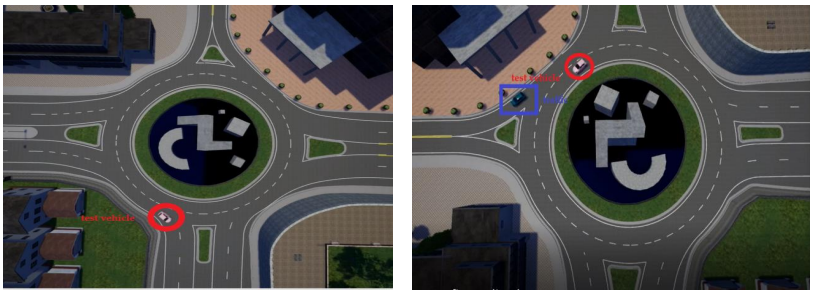
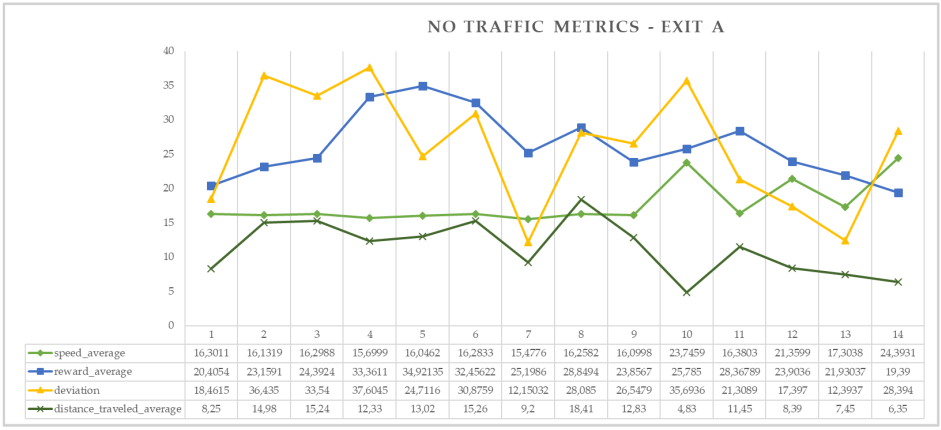


Figure 6. Experimental results: (**a**) no-traffic scenario, (**b**) traffic scenario.

For the two scenarios based on the simulated environment's recorded trajectories for the observed vehicle, the vehicle decision distribution is $a \in \Delta$. The proposed framework was evaluated using the metrics of performance previously cited to enter and exit roundabouts with and without traffic. The speed obtained from the predictive model [1] was adjusted in the trained model to the different segments of a roundabout, where convergence of the vehicle's behavior was observed within the simulation environment. During data collection, drivers involved in the experiment met the following conditions: (1) they used routes with roundabouts with different diameters, (2) they used single and multiple lanes; and (3) they used the same vehicle equipped for testing. An important feature of the naturalistic driving data used for the simulated environment is that the RL algorithm could learn decision-making when arriving at a roundabout, and human behavior seems more promising when it comes to a real-life changing environment.

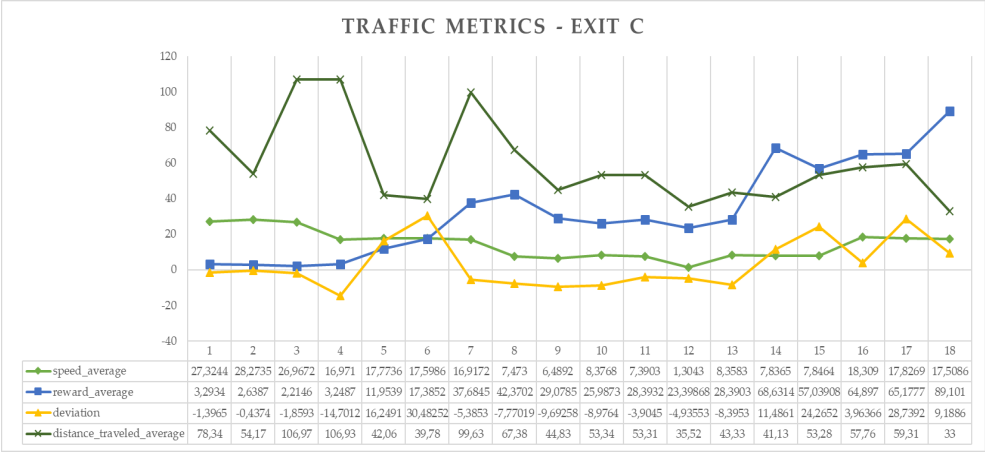
To apply reinforcement learning and obtain an optimal solution through this methodology, the considered reward function was characterized by being bounded between two limit values in its measurement: (−, +). In the results, values with (−) correspond to the vehicle leaving on the left side of the state space and vary linearly between the limit values, and values with (+) correspond to when it leaves on the right side. Figures 7 and 8 show the return of the approach metrics for the simulations without traffic and with traffic, as well as the metric vectors obtained for each situation. As can be seen in the graphs, the metrics converge toward the value (mr) during the training phase, with the exception of traffic simulation data, where they diverge at a given point. This divergence is the result of the discretization of the reward function at the moment when the test vehicle must stop to yield to another vehicle inside the roundabout. Another significant aspect is the average speed (vm) in both experiments. In the case of traffic, the speed is reduced compared to the case without traffic, as well as the distance traveled (dmr) by roundabout typology.

Figure 9 shows that the trajectory for exit A is observed by the vehicle according to the simulation results, where the red line represents the path that the vehicle must follow and the blue line is the obtained path after the reinforcement learning algorithm is applied.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
speed_average	16,3011	16,1319	16,2988	15,6999	16,0462	16,2833	15,4776	16,2582	16,0998	23,7459	16,3803	21,3599	17,3038	24,3931
reward_average	20,4054	23,1591	24,3924	33,3611	34,92135	32,45622	25,1986	28,8494	23,8567	25,785	28,36789	23,9036	21,93037	19,39
deviation	18,4615	36,435	33,54	37,6045	24,7116	30,8759	12,15032	28,085	26,5479	35,6936	21,3089	17,397	12,3937	28,394
distance_traveled_average	8,25	14,98	15,24	12,33	13,02	15,26	9,2	18,41	12,83	4,83	11,45	8,39	7,45	6,35

Figure 7. Metrics result for the no-traffic experiment: exit A.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
speed_average	27,3244	28,2735	26,9672	16,971	17,7736	17,5986	16,9172	7,473	6,4892	8,3768	7,3903	1,3043	8,3583	7,8365	7,8464	18,309	17,8269	17,5086
reward_average	3,2934	2,6387	2,2146	3,2487	11,9539	17,3852	37,6845	42,3702	29,0785	25,9873	28,3932	23,39868	28,3903	68,6314	57,03908	64,897	65,1777	89,101
deviation	-1,3965	-0,4374	-1,8593	-14,7012	16,2491	30,48252	-5,3853	-7,77019	-9,69258	-8,9764	-3,9045	-4,93553	-8,3953	11,4861	24,2652	3,96366	28,7392	9,1886
distance_traveled_average	78,34	54,17	106,97	106,93	42,06	39,78	99,63	67,38	44,83	53,34	53,31	35,52	43,33	41,13	53,28	57,76	59,31	33

Figure 8. Metrics result for traffic experiment: exit C.

Figure 9. Vehicle exit A according to simulation results.

6. Summary and Discussion

In this paper, a framework for reinforcement learning for autonomous driving in roundabout scenarios is proposed. The problem is tackled as a Markov decision process, where the behavioral planning of the vehicle for safely navigating roundabouts uses the Q-learning algorithm. The approach was implemented using the CARLA simulation environment. Simulations carried out in this work used a set of naturalistic driving data from [1], including environmental information, as well as machine learning models for predicting steering angle and vehicle speed. The main contribution of this paper is the design of a tangible learning technique for sequential and automatic decision-making of autonomous vehicles through examples in a simulated environment. In the experiments carried out, the behavior process benefited from a guided policy in automatic decision-making in terms of tangible learning as determined by the implemented Q-learning algorithm. The resulting behavior after the iterative adaptation of the Q-value function allowed the autonomous vehicle to choose the appropriate actions between the start and end of the defined scenarios through GPS positioning in the reward function. The proposed method was evaluated in a challenging roundabout scenario with and without traffic by discretizing the reward function in a high-definition driving simulator. The results, in comparison with other learning methods, show that the autonomous vehicle had improved directionality against the direction of other vehicles, adapted the average speed in a more realistic way in an environment with traffic, and improved the deviation of the vehicle's rotation steering angle without hitting obstacles.

For future work, roundabouts with several exits and shapes as well as other scenarios will be considered. It would also be interesting to simulate the proposed framework using simulation of urban mobility (SUMO) [35], including simulating complete roundabouts (exit D). Another line of work is to compare the results obtained in this paper with application of the deep Q-learning algorithm. Finally, collecting more trajectories for analysis of the training and adaption phases is also desirable.

Author Contributions: L.G.C. was responsible for the design and implementation of the reinforcement learning algorithm, the learning model, and validation. E.P. was responsible for selecting the machine learning algorithms and the general data mining process. J.F.A. was responsible for designing naturalistic driving data. N.A. was responsible for drafting the paper. All authors contributed to writing and reviewing the final manuscript.

Acknowledgments: The Regional Government of Madrid under the SEGVAUTO supported this work in part under the 4.0-CM: P2018/EMT-4362 grant, and the National Plan for Research PN I+D+i under the TRA2016-78886-C3-2-R grant.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The model operation based on the Q-learning algorithm for this paper for the whole procedure is shown in this section.

Algorithm 1 Q-Learning Algorithm.

```
learning_policy = 0.1
learning_rate = 0.1
epochs = 0
discount = 0.6
satisfied_objective = 0

While satisfied_objective ! = 1:
# Info recollected of the environment (speed, sections, position, center distance, state, reward last action, objective satisfied)
    state, reward, satisfied_objective, info = read_environment()
    if (random.random() < learning_Policy) or (state not in q_table):
        action = random_action()
    else:
        action = np.argmax(q_table[state])

# Action = (accelerator, steering wheel, brake)
# Apply on the agent the action
    Control(action)
# Calculate the new value of q
    current_q = q_table[old_state, old_action]
    max_future_q = np.max(q_table[state])
    new_q = (1 - alpha) * old_value + alpha * (reward + gamma * max_future_q)
    new_q = (1 - learning_rate) * current_q + learning_rate * (reward + DISCOUNT *max_future_q)

    q_table[state, action] = new_value
    old_state = state
    old_action = action
epochs + = 1
```

References

1. García Cuenca, L.; Sanchez-Soriano, J.; Puertas, E.; Fernandez Andrés, J.; Aliane, N. Machine Learning Techniques for Undertaking Roundabouts in Autonomous Driving. *Sensors* **2019**, *19*, 2386. [CrossRef] [PubMed]
2. Rodrigues, M.; McGordon, A.; Gest, G.; Marco, J. Autonomous Navigation in Interaction-based Environments—A Case of Non-Signalized Roundabouts. *IEEE Trans. Intell. Veh.* **2018**, *4*, 425–438. [CrossRef]
3. Deluka Tibljaš, A.; Giuffre, T.; Surdonja, S.; Trubia, S. Introduction of Autonomous Vehicles: Roundabouts Design and Safety Performance Evaluation. *Sustainability* **2018**, *10*, 1060. [CrossRef]
4. Rodrigues, M.; Gest, G.; McGordon, A.; Marco, J. Adaptive behavior selection for autonomous vehicle through naturalistic speed planning. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–7. [CrossRef]
5. Zhao, M.; Kathner, D.; Jipp, M.; Soffker, D.; Lemmer, K. Modeling Driver Behavior at Roundabouts: Results from a Field Study. In Proceedings of the IEEE Intelligent Vehicles Symposium, Los Angeles, CA, USA, 11–14 June 2017.
6. Aoude, G.S.; Desaraju, V.R.; Stephens, L.H.; How, J.P. Behavior classification algorithms at intersections and validation using naturalistic data. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 601–606.
7. Gritschneder, F.; Hatzelmann, P.; Thom, M.; Kunz, F.; Dietmayer, K. Adaptive learning based on guided exploration for decision-making at roundabouts. In Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, 19–22 June 2016. [CrossRef]
8. Sutton, R.; Barto, A. *Reinforcement Learning*; MIT Press: Cambridge, MA, USA, 2018.
9. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. *arXiv* **2017**, arXiv:1711.03938. Available online: http://arxiv.org/abs/1711.03938 (accessed on 23 November 2019).
10. Gatti, C. *Design of Experiments for Reinforcement Learning*; Springer International Publishing: Cham, Germany, 2015. [CrossRef]
11. Nageshrao, S.; Tseng, E.; Filev, D. Autonomous Highway Driving using Deep Reinforcement Learning. *arXiv* **2019**, arXiv:1904.00035. Available online: http://arxiv.org/abs/1904.00035 (accessed on 15 November 2019).
12. Sutton, R.; Barto, A. Reinforcement Learning: An Introduction. In *Adaptative computation and ML Series*; MIT Press (Bradford Book): Cambridge, MA, USA, 1998; p. 322. ISBN 0-262-19398-1.
13. Kantarci, B.; Foschini, L.; Corradi, A.; Mouftah, H.T. Inter-and-intra data center VM-placement for energy-efficient large-Scale cloud systems. In Proceedings of the 2012 IEEE Globecom Workshops, Anaheim, CA, USA, 3–7 December 2012. [CrossRef]
14. Watkins, C.J.C.H.; Dayan, P. Q learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
15. Littman, M.L. Markov Decision Processes. *Int. Encycl. Social Behav. Sci.* **2015**, 573–575. [CrossRef]
16. Yang, Z.; May, L.G. A Theoretical Analysis of Deep Q-Learning, 1–56. *arXiv* **2019**, arXiv:1901.00137v2.
17. Hui, T.C.; Tham, C.-K. Adaptive provisioning of differentiated services networks based on reinforcement learning. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2003**, *33*, 492–501. [CrossRef]
18. Choy, M.C.; Srinivasan, D.; Cheu, R.L. Cooperative, hybrid agent architecture for real-time traffic signal control. *IEEE Trans. Syst. Man Cybern. –Part A Syst. Hum.* **2003**, *33*, 597–607. [CrossRef]
19. Andrew Bagnell, J. Reinforcement Learning in Robotics: A Survey. *Springer Tracts Adv. Robot.* **2014**, *97*, 9–67. [CrossRef]
20. Paletta, L.; Pinz, A. Active object recognition by view integration and reinforcement learning. *Robot. Auton. Syst.* **2000**, *31*, 71–86. [CrossRef]
21. Adolphs, R.; Tranel, D.; Damasio, H.; Damasio, A. Impaired recognition of emotion in facial expressions following bilateral damage to the human amygdala. *Nature* **1994**, *372*, 669–672. [CrossRef] [PubMed]
22. Brechtel, S.; Gindele, T.; Dillmann, R. Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 392–399. [CrossRef]

23. Kim, H.J.; Jordan, M.I.; Sastry, S.; Ng, A.Y. Autonomous Helicopter Flight via Reinforcement Learning. In *Advances in Neural Information Processing Systems 16*; Thrun, S., Saul, L.K., Schölkopf, B., Eds.; MIT Press: Cambridge, MA, USA, 2004; pp. 799–806.
24. Riedmiller, M.; Montemerlo, M.; Dahlkamp, H. Learning to drive a real car in 20 minutes. In Proceedings of the Frontiers in the Convergence of Bioscience and Information Technologies, Jeju City, Korea, 11–13 October 2007; pp. 645–650. [CrossRef]
25. Isele, D.; Rahimi, R.; Cosgun, A.; Subramanian, K.; Fujimura, K. Navigating Occluded Intersections with Autonomous Vehicles Using Deep Reinforcement Learning. In Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, QLD, Australia, 21–25 May 2018; pp. 2034–2039. [CrossRef]
26. Jansson, A.; Grönberg, R. Autonomous Driving in Crossings using Reinforcement Learning. Master's Thesis, Chalmers University of Technology, Gothenburg, Sweden, 2017.
27. Cañas, J.I. Simulación en UE4 para realizar Aprendizaje Automático. Bachelor's Thesis, Autonomous University of Barcelona, Barcelona, Spain, 2018; pp. 1–11.
28. Okuyama, T.; Gonsalves, T.; Upadhay, J. Autonomous Driving System based on Deep Q Learning. In Proceedings of the 2018 International Conference on Intelligent Autonomous Systems, Singapore, 1–3 March 2018; pp. 201–205. [CrossRef]
29. Sallab, A.; Abdou, M.; Perot, E.; Yogamani, S. Deep Reinforcement Learning framework for Autonomous Driving. *Electron. Imaging* **2017**, 70–76. [CrossRef]
30. Zhang, Y.; Sun, P.; Yin, Y.; Lin, L.; Wang, X. Human-like Autonomous Vehicle Speed Control by Deep Reinforcement Learning with Double Q-Learning. In Proceedings of the IEEE Intelligent Vehicles Symposium, Changshu, China, 26–30 June 2018; pp. 1251–1256. [CrossRef]
31. Gatti, C.J.; Embrechts, M.J. Reinforcement learning with neural networks: Tricks of the trade. In *Advances in Intelligent Signal Processing and Data Mining*; Georgieva, P., Mihayolva, L., Jain, L., Eds.; Springer: New York, NY, USA, 2012; pp. 275–310.
32. Chen, J.; Yuan, B.; Tomizuka, M. Model-free Deep Reinforcement Learning for Urban Autonomous Driving. *ArXiv* **2019**, arXiv:1904.09503. Available online: http://arxiv.org/abs/1904.09503 (accessed on 1 November 2019).
33. Werling, M.; Ziegler, J.; Kammel, S.; Thrun, S. Optimal Trajectory Generation for Dynamic Street Scenarios in a Fren´et Frame. In Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 987–993.
34. Kunz, F.; Nuss, D.; Wiest, J.; Deusch, H.; Reuter, S.; Gritschneder, F.; Scheel, A.; Stübler, M.; Bach, M.; Hatzelmann, P.; et al. Autonomous Driving at Ulm University: A Modular, Robust, and Sensor-Independent Fusion Approach. In Proceedings of the IEEE Intelligent Vehicles Symposium, Seoul, Korea, 28 June–1 July 2015; pp. 666–673.
35. Krajzewicz, D. Traffic Simulation with SUMO–Simulation of Urban Mobility. In *Fundamentals of Traffic Simulation*; Barceló, J., Ed.; Springer: New York, NY, USA, 2010; pp. 269–293. [CrossRef]