Rear Cross Traffic Alert system for Carla simulator

Enrico Tagliaferri
enrico.tagliaferri2@studio.unibo.it

Lorenzo Guerrini
lorenzo.guerrini5@studio.unibo.it

Abstract

Rear cross traffic alert systems are one of the most important ADAS in a vehicle, due to the fact that rear-end accidents are among the most common. This project analyzes the current approaches and algorithms used for the rear cross traffic alert problem, and proposes a system able to prevent these kinds of accidents. The system uses radar sensors able to detect both vehicles and pedestrian approaching, and engaging the automatic brake in case of imminent collision. The obstacle relevance is then communicated through an MQTT message broker.

1 Introduction

This project is about developing an RCTA system for the Carla simulator (`https://carla.org`).
The RCTA (Rear Cross Traffic Alert) system is an ADAS (Advanced Driver Assistance Systems) useful specially when reversing the vehicle to take it out from the parking slot. The system warns the vehicle driver when it detects perpendicular traffic like people walking, bicycles (not all RCTA systems can detect pedestrians and bicycles) and other vehicles, because they are difficult to be seen.
ADAS are technologies that help the vehicle drivers through a human-machine interface, increasing vehicle and road safety. ADAS are developed to minimize human error, which is the main cause of most crashes[1]. Examples of ADAS includes:

- Adaptive Cruise Control (ACC).
- Rear Cross Traffic Alert (RCTA).
- Vehicle Exit Alert.
- Forward Collision Warning.
- Vehicle Turn Assist.
- Blind Spot Detection.
- Parking Assist.
- Intelligent Headlight Control.
- Pedestrian Protection System.
- Evasive Steering.

According to the national crash database in the US, Forward Collision Prevention systems have the potential to reduce crashes by 29%, Lane Keeping Assistance is shown to offer a reduction potential of 19%, while Blind Zone Detection could decrease crash incidents by 9%[2].
RCTA often uses radar to detect obstacles, but it can also use ultrasonic sensors.
The driver can be alerted by visual warning on dashboard, on rear view mirror or on side mirror and by sound from the rear side of the vehicle. The system may also include the emergency braking.

2 State of the art

Nowadays there exist two different approaches to the rear-end collisions prevention problem [3]. The first approach is based on the kinematic laws, while the second one is based on perception.

2.1 Kinematic approach

The first approach is based on kinematics. The system elaborates in real time the space and time necessary to stop the vehicle, using the laws of motion and the possible values of deceleration and reaction time. Thus, the system is able to determine the minimal distance to stop the movement. If the vehicle approaches beyond this distance, the driver will be notified.

In the continuation will be presented to the reader, some of the major kinetics based algorithms.

It is indicated as follows:

In: M. Meder, A. Rapp, T. Plumbaum, and F. Hopfgartner (eds.): Proceedings of the Data-Driven Gamification Design Workshop, Tampere, Finland, 20-September-2017, published at http://ceur-ws.org

- $d_L = \frac{v_L^2}{\alpha_L}$ the stopping distance of the leading vehicle, from when starts decelerating, with constant deceleration α_L, and an initial velocity v_L.
- $d_F = \frac{v_F^2}{\alpha_F}$ the stopping distance of the following vehicle, from when starts decelerating, with constant deceleration α_F, and an initial velocity v_F.
- *leading vehicle* the vehicle that brakes.
- *following vehicle* the vehicle that has to react to the brake.

2.1.1 The Mazda algorithm

The Mazda algorithm considers a situation in which two vehicles follow each other, initially at a constant speed and at a distance $d > 0$. In the formula used to evaluate the minimal distance to maintain in order to perform a safe brake, with no incidents, are used the following parameters:

- Stopping distances d_L and d_F.
- Deceleration values α_L and α_F of the two vehicles.
- Delay times τ_L and τ_F after which the two vehicles start the brake.
- Velocity difference v_{rel} between the two vehicles.
- Minimal distance R_{min} that the two vehicles need to maintain.

The result is then the critical distance to maintain to avoid an incident. If braking begins at a shorter distance, with the same parameters, the incident will be unavoidable.

$$R_{warning} = \frac{1}{2}(d_L - d_F) + v_F\tau_F + v_{rel}\tau_L$$

2.1.2 The Stop distance algorithm

The algorithm works with the stopping distances of the two vehicles. If the distance difference is lesser than the computed distance, the driver is notified. Another parameter considered in the formula is the reaction time to the leading vehicle brake. This is the time passed between the brake signal of the leading vehicle and the moment in which the driver of the following vehicle starts braking.

2.2 Perceptual approach

The perceptual approach considers the human perception of the driver and it is therefore based on Time to Collision evaluation (also called TTC). The TTC is an expression of how much time the following vehicle needs to collide with the leading one, if the velocity and acceleration values are maintained still, and it is only evaluated if the distance betweene the vehicles is decreasing. A typical value is between 2 and 5 seconds[4].

2.2.1 The Honda algorithm

The Honda algorithm sets a TTC threshold equal to 2.2 seconds. The minimal stopping distance is evaluated from this TTC value and from the relative velocity of the two vehicles. Furthermore an additional safety parameter can be considered, in this case set to 6.2 meters.

2.2.2 The Honda collision mitigation system

The Honda collision mitigation system considers notifying the driver with three alarms, signaled respectively with a TTC value equal to three, two and one seconds. In the initial phase the system is in an *attention* state; as soon as the TTC drops under the first threshold ($TTC = 3$s) the driver is notified. If the TTC decreases furthermore under the other two thresholds, the driver is notified demandingly. Some passive security measures can be utilized in the process. In any case, the goal of this system is to reduce the impact damages, not to completely prevent it.

2.2.3 The Hirst and Graham algorithm

The Hirst and Graham algorithm considers a minimal distance adjustment, based on the actual vehicle velocity. The TTC already considers the velocity of the following vehicle, but in this algorithm this parameter is multiplied by a factor (set to 0.4905m) and then summed to the distance value, evaluated using an initial TTC value of three seconds.

$$R_{min} = TTC \cdot \Delta v + 0.4905 v_F$$

3 Proposed system

The project goal is to realize a working RCTA system. The system alerts via MQTT, via visual warning, via sound warning and with an automatic braking when another vehicle or person is perpendicularly approaching while the owner's vehicle is reversing.

The project includes three scripts:

- The script for the ego vehicle with the RCTA system.
- A script for pedestrian traffic.
- A script for vehicle traffic.

3.1 The ego vehicle script

This is the main script, where the RCTA system is implemented. It spawns a vehicle parked in a parking slot. This vehicle has to reverse to get out of that parking slot.

3.1.1 Radars setup

The RCTA system uses two radars placed in the back of the car, on the left and right sides. The radars are placed facing different sides, both with a horizontal field of view (FoV) equal to 120° and a vertical FoV of 20°. The choice of the value of the horizontal FoV was made to guarantee that the radars were able to capture the maximum number of obstacles approaching the parked vehicle, while the one for the vertical FoV was made to minimize the risk of detecting the road under the vehicle as an obstacle.

Another important parameter was the range distance. Similar to the previous horizontal FoV parameter, this one has been set considering that the radars need to be able to capture the maximum number of obstacles. Giving the fact that this system operates in a static environment, where the velocity of the actors are fairly low, the range has been set equal to a value of 40 meters. The choice was made considering the latter assumption and the typical range values for radars used in systems like this one.

3.1.2 Detection and alert system

Once the radars are set the next interesting thing is the detection and alert system. One important notice about the system is that it needs to be activated under certain conditions. In fact, the system starts working only when the reverse gear is engaged and when a temporal threshold is reached.

The chosen algorithm for the system is the Hirst and Graham algorithm, discussed in the chapter 2. Algorithms using kinematic approach have been excluded, due to the fact that was required the use of the acceleration, both of the ego vehicle and the obstacle. For the first one there was no problem because the Carla API provides the Vehicle object with a getter to obtain the vehicle acceleration. The problems come with the evaluation of the acceleration for the obstacles: for these objects was not possible to obtain the acceleration without calculating it. So, to reduce the risk to evaluate wrong values for the acceleration, a perceptual approach was chosen.

The Hirst and Graham algorithm considers an initial TTC value equal to 3 seconds, but after some tests, it appeared that a value of 0.4 seconds is more indicated. In addition, to make the system work only in the critical situations, it is activated when the distance between the vehicle and the obstacle is decreasing. This control is done by checking if the relative velocity, read by the radars, is less than 0. Once the system falls below this TTC value, an alert is triggered, both visual and aural, and an MQTT message is sent (see chapter 3.1.3 for the discussion about MQTT system). The system is triggered even if the obstacle is nearer than a 1.5m from the ego vehicle.

Giving the fact that the Hirst and Graham algorithm returns the minimum distance at which to start braking to avoid the collision, this distance value is used in the automatic braking feature. If the system is in an alert state (so if the current TTC is greater than the TTC threshold) the collision distance is evaluated, using the algorithm discussed before. If the current distance is lesser than the computed collision distance, the automatic brake engages and stops the vehicle. This feature is activated only if the vehicle is moving: therefore if the ego vehicle is still and someone walks behind it, the obstacle is detected but the automatic brake will not be triggered.

The system is also able to detect the direction from which the obstacle is coming. This is done by comparing the distance values read by the two radars. If only one radar detects the obstacle, it's easy to determine from which direction the obstacle is coming. Otherwise, if both radars detect it, then the difference of the two distances is evaluated. If it is less than a certain threshold (set to 2m), it means that the obstacle is behind the ego vehicle.

3.1.3 MQTT communication system

For the MQTT communication the system relies on the HiveMQ broker service (`www.hivemq.com/`). The system publishes a message on the topic *svs-rcta* each time the TTC is lesser than the threshold. To prevent a flooding effect on the channel, a timer is set, so that the system can publish a message once each a specified time interval (now set to 1s).

3.2 The pedestrian traffic script

The pedestrian traffic script simply spawns a pedestrian on the side of the street where the ego vehicle is parked. The pedestrian walks behind the car that is getting out of the parking space. When the person reaches a distance where he becomes useless for the purpose of the project, he just despawns, respawning in the same place where it had spawned before.

3.3 The vehicle traffic script

The vehicle traffic script is similar to the pedestrian traffic script. It spawns two vehicles on the street where the ego vehicle is parked and they go in opposite directions (one from the left side to the right

side, the other one from the right side to the left side, looking at the front part of the ego vehicle). Like the pedestrian traffic script, this script despawns the vehicles when they reach some predefined coordinates, i.e. when they go out of the range of the utility for this project, and it respawns them in the same point they had spawned before.
The throttle of the vehicles is randomly set in this way:

```
traffic_vehicle.apply_control(
    carla.VehicleControl(
        throttle=random.uniform(0.7, 1)
    )
)
```

This happens so the vehicles do not meet at the same point every time.

4 Conclusions

In this report it is presented the development of a Rear Cross Traffic Alert system for the Carla simulator. Existing approaches have been explored, including kinematics and perceptual ones, and the Hirst and Graham algorithm has been chosen in this system, due to its suitability for real-time obstacle detection. The proposed system utilizes dual rear-mounted radar sensors to monitor cross-traffic and employs a combination of visual, auditory and MQTT-based alerts, to notify the driver of potential hazards. Additionally, an automatic braking feature is integrated to mitigate collision risks. The system is complete with two additional scripts, used to simulate some traffic situations (both pedestrian and vehicle traffic).

References

[1] Brookhuis, Karel A.; de Waard, Dick; Janssen, Wiel H. "Behavioural impacts of Advanced Driver Assistance Systems–An Overview". European Journal of Transport and Infrastructure Research. (https://journals.open.tudelft.nl/ejtir/article/view/3667). *European Journal of Transport and Infrastructure Research*, 2001-06-01.

[2] Aleksa, Michael; Schaub, Andrea; Erdelean, Isabela; Wittmann, Stephan; Soteropoulos, Aggelos; Fürdös, Alexander "Impact analysis of Advanced Driver Assistance Systems (ADAS) regarding road safety – computing reduction potentials". (https://etrr.springeropen.com/articles/10.1186/s12544-024-00654-0). *European Transport Research Review*, 2024-06-27.

[3] F. Bella and R. Russo A Collision Warning System for rear-end collision: a driving simulator study *Procedia - Social and Behavioral Sciences*, 20:676-686, 2011

[4] M. M. Minderhoud and P. H.L. Bovy Extended time-to-collision measures for road traffic safety assessment *Accident Analysis & Prevention*, 33(1):89-97, 2001