

# Autonomous vehicle control systems for safe crossroads

Javier Alonso\*, Vicente Milanés, Joshué Pérez, Enrique Onieva, Carlos González, Teresa de Pedro

*Centro de Automática y Robótica (CAR), UPM-CSIC, Carretera de Campo Real, km. 0.200, 28500 La Poveda, Arganda del Rey, Madrid, Spain*

## A B S T R A C T

This article presents a cooperative manoeuvre among three dual mode cars – vehicles equipped with sensors and actuators, and that can be driven either manually or autonomously. One vehicle is driven autonomously and the other two are driven manually. The main objective is to test two decision algorithms for priority conflict resolution at intersections so that a vehicle autonomously driven can take their own decision about crossing an intersection mingling with manually driven cars without the need for infrastructure modifications. To do this, the system needs the position, speeds, and turning intentions of the rest of the cars involved in the manoeuvre. This information is acquired via communications, but other methods are also viable, such as artificial vision. The idea of the experiments was to adjust the speed of the manually driven vehicles to force a situation where all three vehicles arrive at an intersection at the same time.

### Keywords:

Autonomous vehicle  
Intelligent transport system  
Traffic management

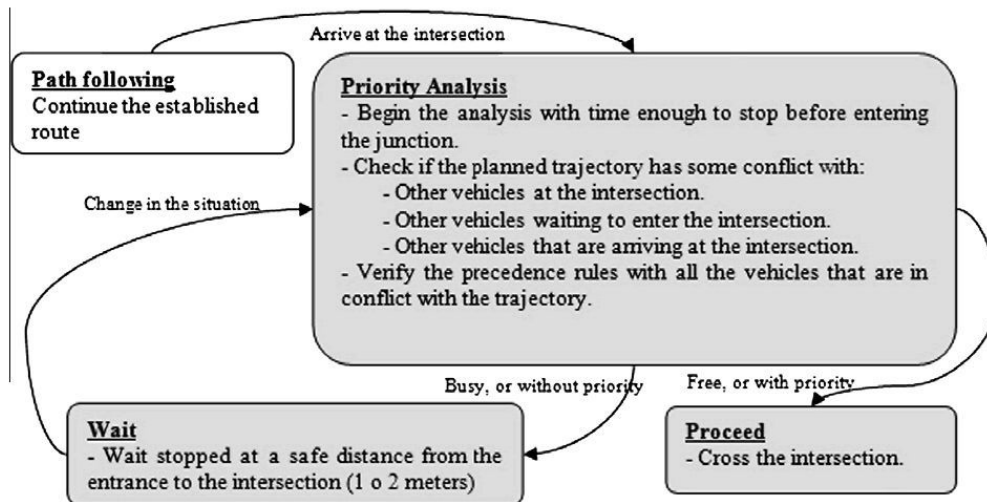
## 1. Introduction

This work was done as part of the AUTOPIA project. The Robotics and Automation Centre (CAR), a joint research centre of Spain's Superior Scientific Research Council (CSIC) and the Polytechnic University of Madrid started the AUTOPIA project in 1996. The research group has been working towards the full automation of autonomous vehicles in urban areas, and is now focusing on cooperative manoeuvres among cars.

The research centre has developed three dual mode cars, and has an experimental track that is designed to resemble an urban area (see Section 4). These cars are fully automated, but can also be operated manually. The accelerator, the brake, and the steering wheel of the vehicles have been automated and are controlled by a computer. The vehicles are also equipped with differential GPS, inertial measurement units (IMU), magnetic compass, and wheel encoders to provide precise positioning (Milanés et al., 2008). The car position is compared with the map of the desired route, and a fuzzy controller sends the needed orders to the accelerator, the brake, and the steering wheel to keep the car on its route (Onieva et al., 2009).

From 2005 to 2008, the AUTOPIA group lead the Control Algorithms package of the European project "EU STREP FP6-028062 CyberCars2: Close Communications for Cooperation between CyberCars" (CC2). A common architecture and a communications standard were developed to allow cooperation between the consortium partners: TNO (Holland), INRIA (France), CRF (Italy), and AUTOPIA (Spain) (Naranjo et al., 2009). But to perform cooperative manoeuvres among several autonomous vehicles, a new algorithm to resolve priority conflicts at intersections was also needed (Alonso, 2009).

The present communication presents two methods of establishing road junction priorities. The first proposed solution uses priority tables, and was the result of an exhaustive study of the different scenarios that may occur at an intersection. The second is a simplification of the first, taking advantage of the knowledge gained in the initial study of the manoeuvre to provide a solution based on priority levels.



**Fig. 1.** General scheme of the intersection manoeuvre.

Each of the autonomous vehicles acts independently. In other words, the control and the decision making system of each vehicle is individual, and depends on the vehicle's current status and the information the system has about the environment. The state at a junction or intersection is described from the viewpoint of one of the vehicles which will be termed the reference vehicle, although each vehicle at the intersection will have to take its own decisions, and all should be consistent. The movement of the cars will be studied from the standpoint of the reference vehicle, and other cars will be referred to according to the direction in which they are approaching the intersection.

The goal of the present work was to permit the vehicles to undertake the mission of coordination so that no central coordinating organ is necessary, i.e., there is no need for infrastructure modifications. The improvement in the management of the flow of vehicles at intersections is expected to reduce traffic congestion problems, since the capacity of intersections is 50% less than the capacity of the route (Board, 2000). We define an intersection as a crossing of two-way roads without any traffic lights managing it. The general outline of the procedure governing the manoeuvre through an intersection for a vehicle is shown in Fig. 1.

## 2. Autonomous vehicles

This section presents the autonomous vehicles used in the experiments (Fig. 2). AUTOPIA's new intersection decision system was implemented on a Citroen C3 Pluriel which is capable of performing autonomous driving. The vehicle is equipped with a double-frequency GPS receiver running in RTK carrier phase differential mode that supplies 2 cm resolution positioning at a refresh rate of 5 Hz. To avoid degradation of the positioning when the correction signal is lost, the vehicle uses its own odometry supplied by a set of built-in sensors in the wheels whose measurements can be read by accessing the Controller Area Network (CAN) bus of the vehicle, and a Crossbow IMU300CC inertial measurement unit (IMU) placed close to the centre of the vehicle. This system allows automated vehicle guidance in urban areas. We have tested this autonomous guidance system for up to 5 min before the drift becomes too large (Milanés et al., 2008).



**Fig. 2.** The CAR's autonomous vehicles.



Fig. 3. The CAR's experimental track.

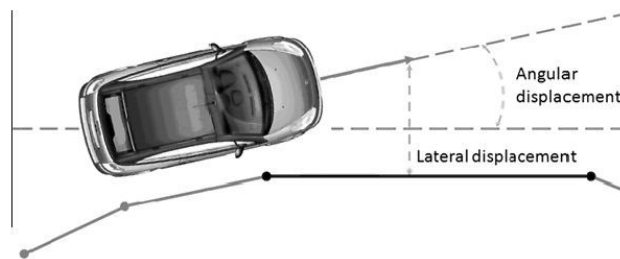


Fig. 4. Lateral and angular deviations along the reference segment.

An on-board computer uses the accurate position and the speed provided by the wheel encoders to send control signals to the steering wheel, the accelerator, and the brake. The computer uses a fuzzy logic controller to calculate the correct target positions of the car actuators (Alonso et al., 2005; Onieva et al., 2010).

The device that makes it possible to act on the throttle pedal is an analogical output card that sends signals to the car's internal engine computer to demand acceleration or deceleration. With respect to the brake pedal, an electro-hydraulic braking system was installed in the trunk of the car. It is capable of generating pressures up to 160 bars (via an electro-proportional pilot) and is connected to the brake circuit of the vehicles through two shuttle valves located before the anti-lock braking system (ABS) (Milanes et al., 2010). For the steering wheel, electric power steering (EPS) is used. A power signal is sent to the EPS motor, bypassing the EPS sensor and its control card (Perez et al., 2009).

Basically, there are two fuzzy controllers that manage the two vehicle control signals – speed and steering, also known as longitudinal and lateral controls. The longitudinal control manages the throttle and brake, and the lateral control manages the steering wheel. The fuzzy steering control is in charge of minimizing trajectory deviations. The inputs of this controller are the lateral and angular deviation along the reference segment (see Fig. 4). The output is the angle that the steering wheel must be turned to correct the deviation of the trajectory. Throttle and brake fuzzy control permit the vehicle to adapt its speed to a reference, reducing or increasing it when necessary in order to maintain its route or a safety distance from the preceding vehicle. The inputs of these controllers are the difference between the desired and the actual speed, the acceleration, and, if the vehicle is performing an adaptive cruise control (ACC) manoeuvre, the distance to the preceding vehicle and the variation of this distance. The outputs of these controllers are the pressures that must be applied onto each pedal in order to minimize the speed errors.

The fuzzy controllers are based on a fuzzy inference machine denominated ORBEX (Spanish acronym for Fuzzy Experimental Computer), which was developed at the IAI (García and de Pedro, 2000). ORBEX allows definition of the autonomous vehicle's behaviour by simple IF-THEN rules that are easily understandable. The details of the fuzzy inference machine are: the t-norm minimum and the t-conorm maximum are used to implement *and* and *or* operators, respectively; Mamdani-type inference (Mamdani, 1976) is used; and the defuzzification operator is the centre of mass.

### 3. Related work

There have been two types of approach to the problem of resolving priority conflicts at intersections. The first is by simulation, considering tens, hundreds, or thousands of vehicles. The main interest in these studies has been to maximize the efficiency of the intersections in terms of delay times, number of vehicles crossing per unit time, etc. Some of the most outstanding centres involved in this area of investigation are the University of Austin in Texas (van Middlesworth et al., 2008; Dresner and Stone, 2008), and the Universities of Paderborn (Naumann et al., 1998) and Stuttgart (Regele, 2008) in Germany. Generally, the goal of these works has been limited, with no attempt to go beyond the study of the proposed techniques, leaving their implementation pending.

The second type of approach is more pragmatic rather than theoretical, and is where the present work best fits. Even though slightly different, work such as that connected to the Defense Advanced Research Projects Agency (DARPA) compe-

tition (Thrun et al., 2006; Urmson et al., 2008; Campbell et al., 2007) has concentrated on detecting the occupation of the intersection by means of artificial vision and laser radar so as to be able to traverse it as soon as possible. Other approaches, such as those of the INRIA (Bouraoui et al., 2006) and the University of Griffith (Kolodko and Vlacic, 2003; Baber et al., 2005) solve the problem for two vehicles at an intersection of two one-way roads. Considering the possible future trajectories of the two vehicles, they determine whether the intersection is safe, and, if not, the vehicle without road priority waits until it can continue its route with no risk of collision. The difference in this work is that the traffic rules are compulsory and the vehicles must obey them.

#### 4. Priority attribution methods at an intersection

When vehicles are close to an intersection they must decide whether to continue along their planned route or to stop before crossing. If the intersection is already occupied they must stop, but if it is free, they must determine whether it is their turn to cross it or if another vehicle has priority. Two methods of priority analysis to resolve this dilemma are presented. Each has its advantages and disadvantages, not so much in terms of their effectiveness or the performance of the vehicles, but because of their flexibility in allowing the desired priority assignments to be configured. In particular, the two methods use the same data and generate the same output -move on or give way- but one permits exceptions to the default traffic rules to be configured.

The methods to be presented consist in checking whether the path of the vehicle has conflicts with other vehicles waiting to enter the intersection or other vehicles that are arriving at it, and applying the precedence rules to all vehicles that are in conflict with its trajectory.

The two methods have a common underpinning – the cartographic system (see details in Alonso et al., 2011). AUTOPIA's vehicles use a graph-based cartographic system for autonomous car navigation. This system has the capability of storing a great amount of information -speed limits, traffic signs, etc.- in addition to the topological (connections among the features of the map) and metric (coordinates) information. The logical structure of the cartography is a directed graph that relates arcs and nodes. The arcs represent the roads of the driving area, and the nodes represent the intersections (or singular points within the driving zone, e.g., a point along a straight segment where the speed limit changes). A node may have several points of entry and exit, and contains the internal paths that connect the points of entry and exit within the node. The route a vehicle has to follow is defined by a succession of nodes. With the cartographic structure, an arc in the vehicle's route includes the information on the node it comes from (initial node), the node it is going to (final node), and the subsequent node of the route (future node); see Fig. 5. Given a node that the reference vehicle is approaching along an arc, the structure also tells one which other arcs lead to it, and hence one can determine which cars are located in these arcs, and hence study any potential conflict.

Using a previously developed vehicle-to-vehicle (V2V) communication system (Milanés et al., 2011), each vehicle has data on the position, speed, next turn intention, accuracy of DGPS, and name of the other vehicles traveling through the experimental area. The methods of solution to be presented use this information and the labels of the arcs that connect to its next node (final node) to determine whether the vehicle has to give way at that intersection.

##### 4.1. First method of resolution: priority charts

To make the right decision at a junction, each vehicle must know which vehicles are already in it, and which vehicles are approaching it and could interfere with its operation. To facilitate access to this information, a relational database was created containing the graph representation of the experimental area. Each vehicle has a copy of the database. Due its small size

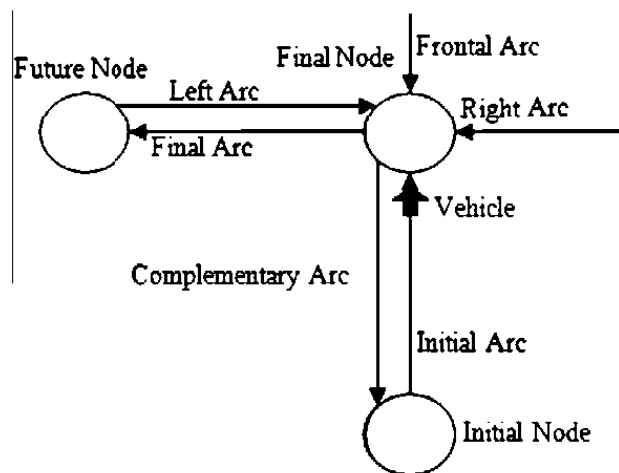


Fig. 5. Known nodes and arcs.

it can be downloaded once the car is near to the driving area. A vehicle nearing an intersection will query the database to obtain the paths leading into the intersection (Alonso et al., 2011), and then will only need to take the vehicles moving along these paths into account.

Having identified these vehicles, the next step is to identify the ones approaching the intersection and discard those that are too far away to interfere with the manoeuvre. Then, once their positions, speeds, and next turn intentions are known, the vehicle that is examining these data will be able to determine whether it can continue or should wait and give way to others. We shall describe a case study in which we first separate the simple cases for which no doubt exists about the control action to take.

These simple cases are:

- If the intersection is no longer occupied, and there is no vehicle approaching it in the vicinity of the junction, the movement will continue unless there is a Stop sign or prohibition indicating otherwise.
- If the intersection is already occupied by another vehicle and its trajectory interferes with our current path, our vehicle must stop before entering and wait for the intersection to clear before it can proceed.

The rest of the cases have no immediate solution. An exhaustive case study has been performed, resulting in a methodological procedure to resolve the conflict of priorities at junctions.

The tool to help in the decision making process will be a vector whose components represent occupancy of the paths leading into the intersection and the turning intentions of the occupying vehicles. Another vector will contain the information representing the traffic signs to include the possibility of altering the priorities of the intersection by Yield, Stop, and even temporary No Entry signs. This last sign is included to permit such situations as an interruption due to maintenance work. A vehicle finding such a sign in its desired path will wait for the sign to be removed.

The useful cases that will be studied take place in a crossroads type intersection (shaped like a +). The occupation of the roads is represented as a vector to allow the method's extension to roundabouts and simplification to a T-junction. Each element of the vector corresponds to a path, and contains a 0 if the corresponding path is empty or a non-null value if it is occupied. The first element corresponds to the path of the reference vehicle, and the following ones to the paths leading into the intersection counter-clockwise from that of the reference vehicle. If there is a vehicle in a path, the value of the corresponding vector component will be the code for its destination, in particular, if the car intends to turn right, a «1», because it is the first exit on the right, if its intention is to go straight on, a «2», and if it wants to turn left a «3». The traffic sign vector is composed analogously, as a series of components corresponding to the paths leading into the intersection, counter-clockwise from the reference vehicle path which is the first component. The sign which will be present at the end of each path reaching the intersection is noted with an «S» for the Stop sign, a «Y» for the Yield sign, and an «N» (No Entry sign) if an exit of the intersection is forbidden.

To clarify this point, consider the example in Fig. 6. The occupancy vector is completed as follows. The first component of the vector corresponds to the vehicle that is evaluating the situation (reference vehicle). The value will be a 1 if the vehicle intends to turn right, as is the case, a 2 if it intends to go straight on, and a 3 if it intends to turn left. The second component of the vector represents the next road in the counter-clockwise direction reaching the intersection. In this case there is no vehicle approaching, so this component will be 0. The next road, in this case that opposite the reference vehicle, corresponds to the third component of the vector. Since there is a car that intends to go straight on, a 2 is assigned to this component. Finally, for the road approaching the intersection from the left, the fourth component will be assigned a 3, since the car approaching the intersection along this road intends to make a left turn. Thus, the occupancy vector for the reference car will be (1,0,2,3).

With respect to the traffic sign vector, three types of signs at the intersection are to be distinguished -No Entry, Stop, and Yield. With the No Entry sign, as the vehicle route is designed from an a priori knowledge of the cartography, one can dynamically modify the map by closing a path to traffic which the vehicle otherwise might enter. The study and treatment of this sign is different from that applied to the Stop or Yield signs.

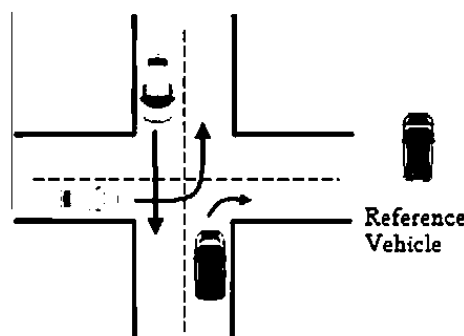


Fig. 6. A crossroads intersection (in the shape of a + sign) with three cars approaching it.

The first step consists on building the No Entry sign vector. We shall explain it in some detail using the example in Fig. 7 in which the reference car is the one approaching from the south. According to the path numbering scheme we are using, since the road to the right of the vehicle is closed temporarily the second component of the traffic sign vector is assigned an  $\langle\langle N \rangle\rangle$ . If it were the road in front of the vehicle, the third component would be marked with  $\langle\langle N \rangle\rangle$ ; and if it were the road to its left, the fourth component would be marked.

The Yield and Stop signs are dealt with differently. We will see it in the same example of Fig. 7. The first component of the reference vehicle's traffic sign vector is in this case  $\langle\langle 0 \rangle\rangle$  as there is no sign in the reference vehicle's path. In case a Stop sign were present in the path of the reference vehicle it would be registered here. The second component corresponds to the lane that gets into the intersection coming from the right – in the example there is a No Entry sign in the lane that goes to the right from the reference vehicle point of view, but the component would be assigned a  $\langle\langle 0 \rangle\rangle$  as far as yields or stops is concerned. The third component corresponds to the lane entering the intersection opposite the reference vehicle. In the case of the figure, this component will be assigned a  $\langle\langle Y \rangle\rangle$  to indicate that the vehicle arriving along this road must yield to the others. And finally the fourth component of the vector corresponds to the traffic sign that is on the lane to the left of the reference vehicle. In this case it would be an  $\langle\langle S \rangle\rangle$  to indicate that the vehicle that approaches along that road must stop and allow other vehicles to pass.

Both, the information of the No Entry signs and the Yield or Stop signs are merged into one vector. Should a Yield sign and a No entry sign be present in the same intersection entrance, a  $\langle\langle NY \rangle\rangle$  would be assigned. In the present case the assignment of the right entrance is only  $\langle\langle N \rangle\rangle$ . Thus the vector of our case study is (0, N, Y, S). Note that in assessing who has priority at the junction the Yield and Stop signs will be treated equally. The treatment of the traffic signs will be discussed in greater detail below.

The determination of whether the vehicle will proceed or not is done with the help of an auxiliary table that contains all the possible vectors associated with a Boolean value that indicates  $\langle\langle Move \rangle\rangle$  or  $\langle\langle Stop \rangle\rangle$ . The procedure thus consists of using this table to look up the value associated with the occupancy vector. The difficulty of implementing this method is the preparation of the table with all possible situations. There are 111 basic combinations that cover the case of two or three vehicles arriving at an intersection, but without taking into account the intersection signs (3 for one vehicle, 27 for two vehicles, and 81 for three vehicles, in total 111 cases). And there are more than 700 further cases when the traffic signs are also considered (111 basic cases multiplied by, at least, 7 possible sign combinations).

In designing the auxiliary table, the coherence of the decisions is crucial. For this reason it is necessary to ensure that, given a situation where there is a risk of collision, permission to access the intersection has been granted to a single vehicle, and the rest must wait their turn. In the manoeuvres where only two cars are involved the coherence of decisions can be ensured by comparing the system's responses to the two vehicles. Obtaining the occupancy vector of one of the vehicles from the other is straightforward. If the occupancy vector is rotated towards the left, one obtains the vector from the standpoint of the vehicle arriving at the intersection from the right of the first car. By comparing the results of the two vectors, one can instantly ensure their coherence. In the case of three or more vehicles being involved, the same method of checking the coherence is used, but pairwise.

#### 4.2. Second method of resolution: priority levels

This method uses priority levels to determine whether one can continue or must wait for the other vehicles to pass. Each vehicle determines its own priority level and compares it with the priority levels of the other vehicles that are at the intersection. It is not necessary for the vehicles to communicate their priorities for comparison because, by using the next turn intention (which is sent within the communications package), the position, and the speed, they can determine the priority levels of the other vehicles.

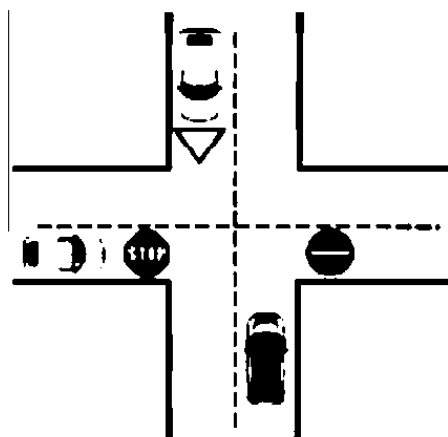


Fig. 7. Traffic signs at an intersection with three cars approaching it.

As with the previous method, there are some basic scenarios that, once established, require no further study:

- If the intersection is empty and no other vehicle is in the vicinity of the intersection, the car will continue unless there is a Stop sign present.
- If there is already a vehicle in the intersection or a vehicle estimated to enter the intersection earlier, and that vehicle interferes with the desired trajectory of the car, our reference vehicle should stop before entering the intersection and wait until the situation changes.

For other cases, a priority evaluation system is needed. When several cars arrive at an intersection simultaneously, or when the intersection is free but there are several vehicles waiting to enter it, the procedure that will be followed is: If the car is coming to an intersection, one must:

1. Look for a Stop sign (using information from the map, or the signal detection algorithms of such systems as the Zigbee sensors (Milanés et al., 2010)).
  - If there is a Stop sign, the car must stop before entering the intersection, then re-evaluate the situation considering the Stop sign as a Yield sign.
  - If there is no Stop sign, proceed to Step 2.
2. Review the occupancy of the intersection.
  - If the intersection is free, proceed to Step 3.
  - If another vehicle is at the intersection but does not interfere with the planned trajectory, proceed to Step 3 (Table 1 can be used to check whether the trajectories of the two vehicles are incompatible).
  - If another vehicle is at the intersection and interferes with the planned trajectory, our reference car must stop before entering the intersection and re-evaluate the situation when the intersection is free.
3. Check for vehicles approaching the intersection on the other converging roads (using V2V communication and the local map).
  - If there are no vehicles in the vicinity, the car can continue its route.
  - If there are vehicles waiting at the intersection or approaching it, but their trajectories do not interfere, the car can continue.
  - If there are vehicles waiting at the intersection or approaching it, and their trajectories interfere with the trajectory of our reference vehicle, then a priority analysis is required to determine whether our vehicle has the highest priority and therefore must cross first or must wait.

If the reference vehicle has stopped before entering the intersection, it must follow the above procedure but skipping the first step. The flowchart of the intersection operation is shown in Fig. 8.

A truth table (see Table 1) is a simple method for detecting conflicts between vehicles at a crossroads intersection. Fig. 9 shows the numbers assigned to the paths relative to the path of the reference vehicle. The courses a vehicle can take are defined as a pair of these digits – the first corresponding to the entry path, and the second to the exit path. The reference vehicle can go along courses 12, 13, or 14. Similarly, the courses available to the other vehicles are: 21, 23, 24, 31, 32, 34, 41, 42, and 43.

Table 1 shows the cases in which the routes planned by two vehicles are potentially compatible. If the intersection of the two routes is marked by a 0 it means there is no conflict. If the intersection is marked with a 1 it means they are incompatible. If it is marked with an X, in principle there is no conflict, but this could change if the size of the intersection is not large enough relative to the turning radius of the cars.

The method uses the occupancy of the roads that enter the intersection, the turning intentions of the vehicles, and the priority traffic signs (Stop or Yield) at the intersection. Traffic signs used to indicate routes with priority (priority, end of priority, and priority at the next intersection) were not taken into account since they are always accompanied by Yield or Stop signs on the non-priority routes that cross the priority path.

The priority assignment algorithm will take into consideration the vehicles' turning intentions, the occupancy of the rest of the converging paths, and of course the traffic signs. Priority will be assigned according to the traffic code.

An initial assessment of the priority for all the vehicles converging towards the intersection is made by considering the traffic signs at the intersection and their turning intentions. Schematically, initial priorities are assigned for both the reference vehicle and the others according to the following set of rules, from higher to lower precedence:

**Table 1**  
Intersection conflict truth table.

	21	23	24	31	32	34	41	42	43
12	X	0	0	0	1	X	X	1	X
13	1	1	1	0	1	X	X	1	1
14	1	1	1	1	1	1	X	1	1

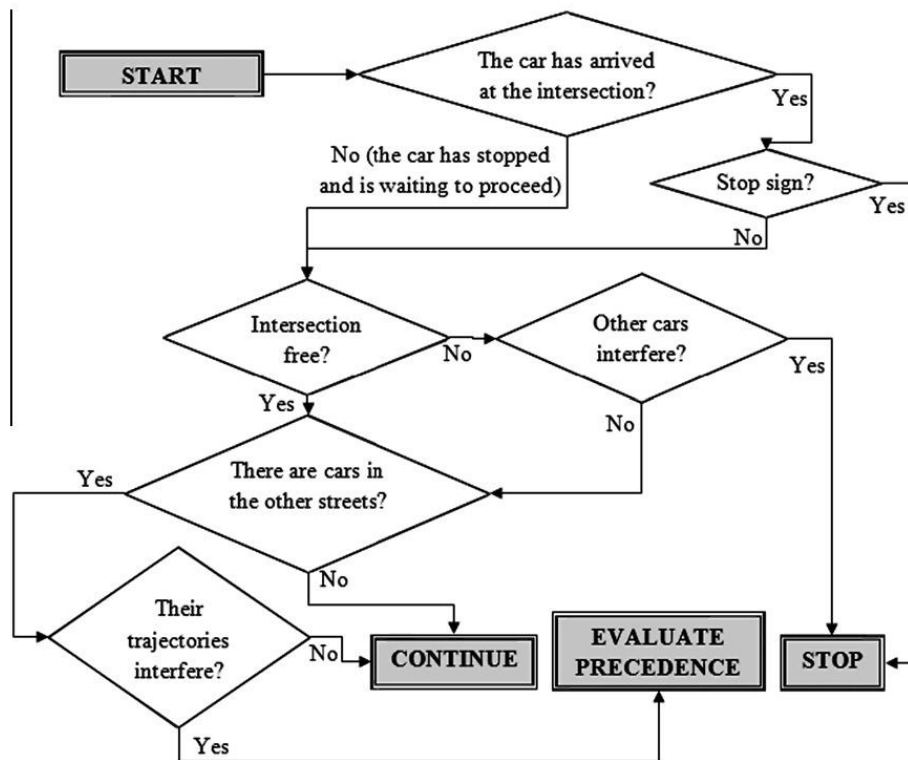


Fig. 8. Decision flow chart at an intersection.

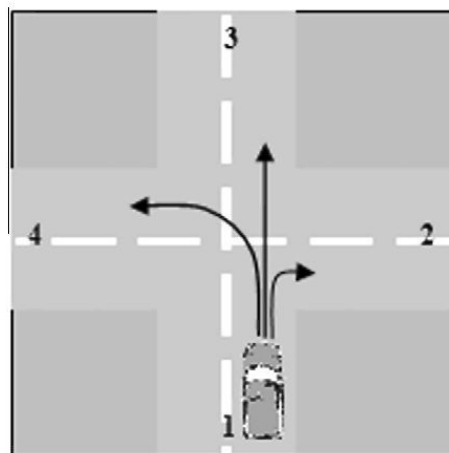


Fig. 9. Numbering of roads intersecting at a crossroads.

- If there is a Yield or Stop sign then the initial priority is *Very Low*.
- If the vehicle's intention is to turn left then its initial priority is *Low*.
- If a vehicle's intention is to go straight on then its initial priority is *Normal*.
- If a vehicle's intention is to turn right then its initial priority is *High*.

These initial priorities are modified if another car is approaching the intersection on the path to the right of the reference vehicle. The priorities are changed as follows:

- If the path on the right is free then a + is added to the priority.
- If the intention of the vehicle coming from the path on the right is to turn left then the priority is *Unchanged*.
- If the intention of the vehicle coming from the path on the right is to go straight on or turn right a - is added to the priority.



After the priorities for all the vehicles have been computed, they are ordered from the lowest to the highest. Thus, a vehicle can assess its own priority and those of the rest of the vehicles at the intersection and check whether its priority is the highest. When two or more vehicles with the same priority are attempting to enter the intersection, the conflict will be resolved by using the time stamp of the first message each one sent expressing its turning intention. Should two vehicles make the demand with the same time stamp, the one with the highest ID will have the priority.

## 5. Experiments

To corroborate the proposed methods, two scenarios were chosen from among those possible – in particular, we selected scenarios in which the vehicle must stop and give way for some time. The tests were performed with two manually driven vehicles, Platero and Rocinante, which have a DGPS positioning system and a communications system that can send their position, speed, and intention to a requesting vehicle. The reference vehicle with automatic driving capabilities, Clavileño, is capable of receiving that information from the other two vehicles and acting accordingly. In one of the scenarios, only one of the other vehicles must be given way to, and in the other both vehicles must be given way to. These scenarios are:

**Scenario 1:** The reference vehicle intends to go straight ahead, but a vehicle approaches in the opposite direction with the intention of making a left turn, and another vehicle approaches from the right whose intention is also to carry straight on (Fig. 10). In this scenario the vehicle has to give way to the car on its right only. The latter will be the first to pass because the car on its right intends to turn left and must give way to the first vehicle. A lower priority is assigned to vehicles whose intention is to turn left because they require more space and time for the manoeuvre. The reference vehicle will thus give way only to the car on its right.

**Scenario 2:** The reference vehicle intends to go straight ahead, but a vehicle approaches in the opposite direction with the intention of carrying straight on, and another vehicle approaches from the right whose intention is also to carry straight on (Fig. 11).

In this scenario the reference vehicle has to give way to the car on its right, and this in turn to the car on its own right. The expected behaviour is, therefore, for the car to wait at the junction until the other two vehicles have passed.

These tests were conducted in the experimental area of the Robotics and Automation Centre (CAR) (Fig. 3).

### 5.1. First method of resolution: priority tables

#### 5.1.1. Scenario 1

The reference vehicle intends to go straight on, but a vehicle whose intention is also to go straight on is approaching on its right and another is approaching from in front intending to turn left (Fig. 10).

The occupancy vector generated in this case is "2230". In the absence of traffic lights that alter the priority, the associated action that the database indicates is "Stop". The vehicle stops. Eventually, after the car coming from the right of the reference vehicle has passed, the occupancy vector becomes "2030", whose associated action is: "Move", so the reference vehicle moves forward.

The diagram (Fig. 12) shows the speeds of the three vehicles. Clavileño (the automatic car) stops upon reaching the intersection (second 12.5). Rocinante (manually driven) also stops at approximately the same instant in order to give way. While the two cars remain stopped, Platero (manually driven) continues. Clavileño remains stopped for a short period of time and then continues its movement after Platero has left the intersection. Rocinante remains stopped for more time until its turn to the left is considered safe. It resumes its course and the test is finished.

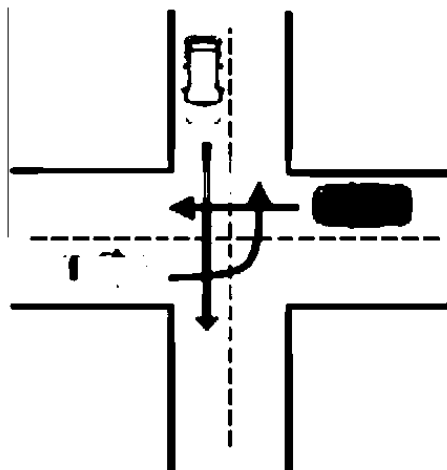


Fig. 10. Test scenario 1.

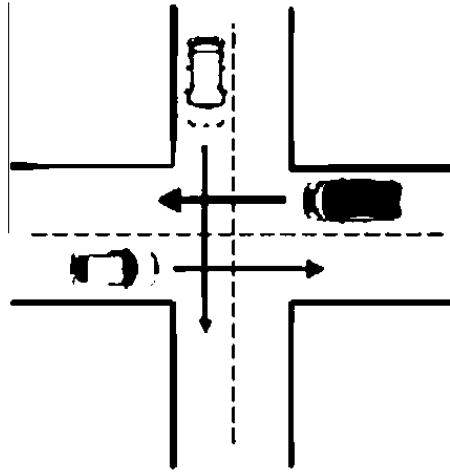


Fig. 11. Test scenario 2.

The figure (Fig. 13) shows the evolution of the positions of the three vehicles. Some time stamps have been added to identify the positions of the vehicles at certain moments. One observes how Clavileño remains stationary, giving way to Platero, and then restarts while Rocinante remains stationary because Clavileño has priority. This result is consistent with the expected behaviour. The positions of the three vehicles (Clavileño, Platero, and Rocinante) are shown throughout the experiment: before Clavileño stops, while Clavileño is yielding, once Clavileño has started again but Rocinante is still yielding, and after Rocinante starts moving again.

#### 5.1.2. Scenario 2

The reference vehicle intends go straight on, but a vehicle whose intention is also to go straight on is approaching from its right and another incoming vehicle is also intending to go straight on. In this scenario the reference vehicle has to give way to the car on its right, and the latter in turn to the one on its right (Fig. 11). The expected behaviour is, therefore, for the reference vehicle to wait at the junction until the other two vehicles have passed.

The occupancy vector generated by the vehicles in this case is "2220". In the absence of traffic lights that alter the priority, the associated action indicated by the database is "Stop" Once the first vehicle has left the intersection the occupancy vector becomes: "2200", whose associated action is also "Stop". When the second vehicle has also left the intersection the occupancy vector becomes: "2000", whose associated action is "Move", so that the reference vehicle moves forward.

In the speed diagram (Fig. 14), one observes how the manually driven vehicles (Platero and Rocinante) do not actually stop, but alter their speeds to give way. Rocinante accelerates slightly while Platero brakes gently. Once the two vehicles have left the intersection Clavileño resumes its movement.

Fig. 15 shows the positions of the three vehicles and the time stamps added to identify their positions at certain moments. One observes how Clavileño remains stationary, giving way to Platero. This is consistent with the expected behaviour. The positions of the three vehicles (Clavileño, Platero, and Rocinante) are shown throughout the experiment: before Clavileño stops, while it is giving way, and when it starts again.

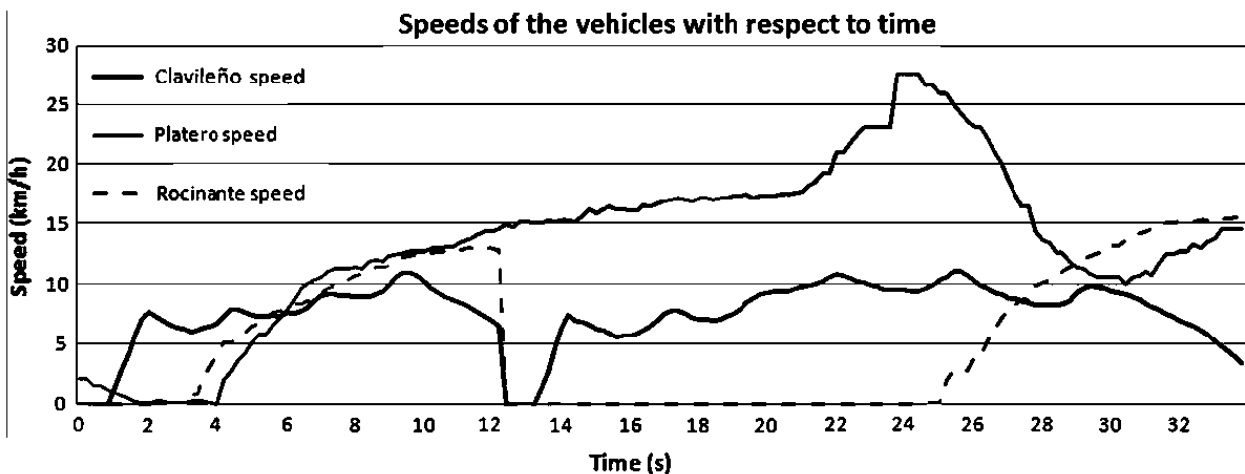


Fig. 12. Vehicle speed diagram for the first scenario, using the first solution method: priority tables.

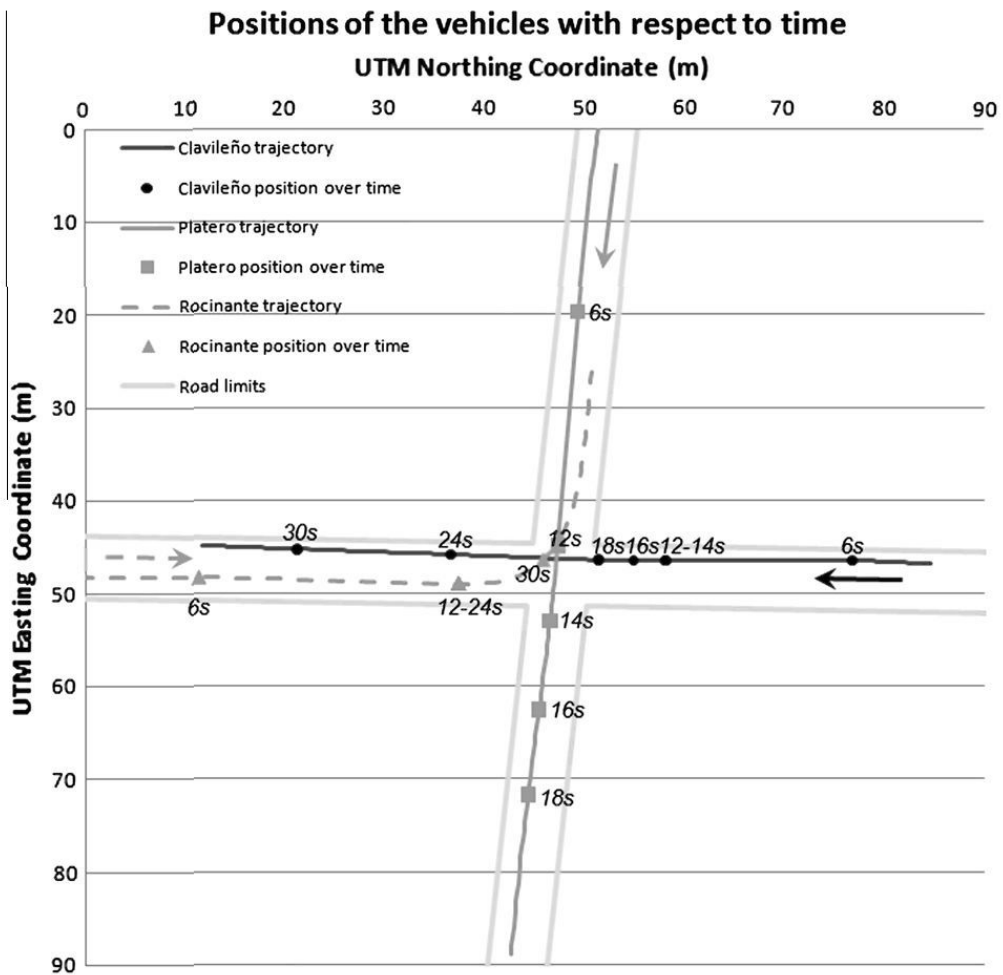


Fig. 13. Positions of the vehicles during the first scenario, using the first method of resolution: priority tables.

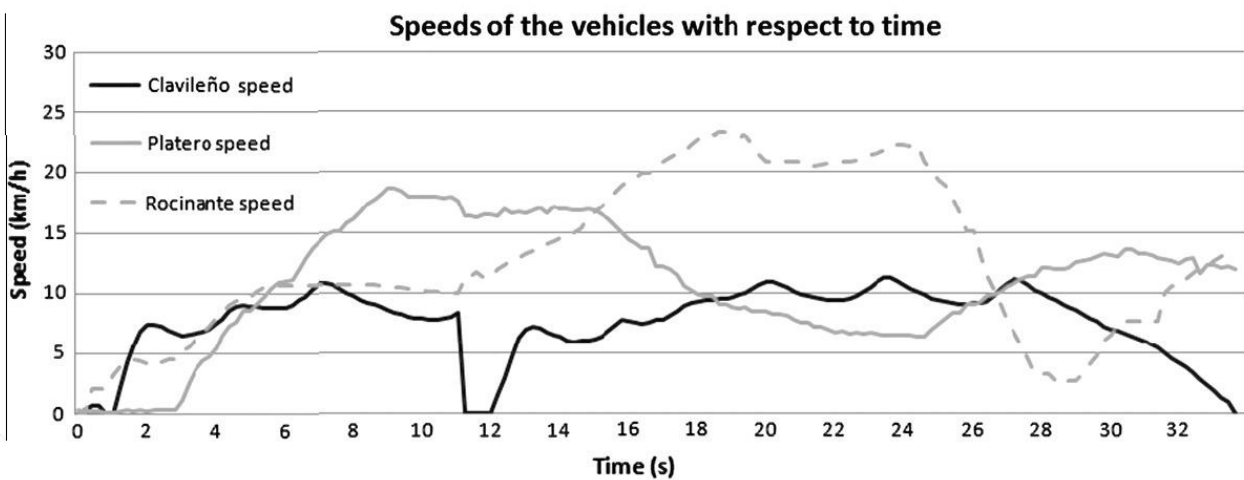


Fig. 14. Vehicle speed diagram for the second scenario, using the first method of resolution: priority tables.

## 5.2. Second method of resolution: priority levels

The priority levels provided by this method are: "Very Low-, Very Low, Very Low+, Low-, Low, Low+, Normal-, Normal, Normal+, High-, High, High+". In the explanation of the experiments, abbreviations will be used when referring to these levels.

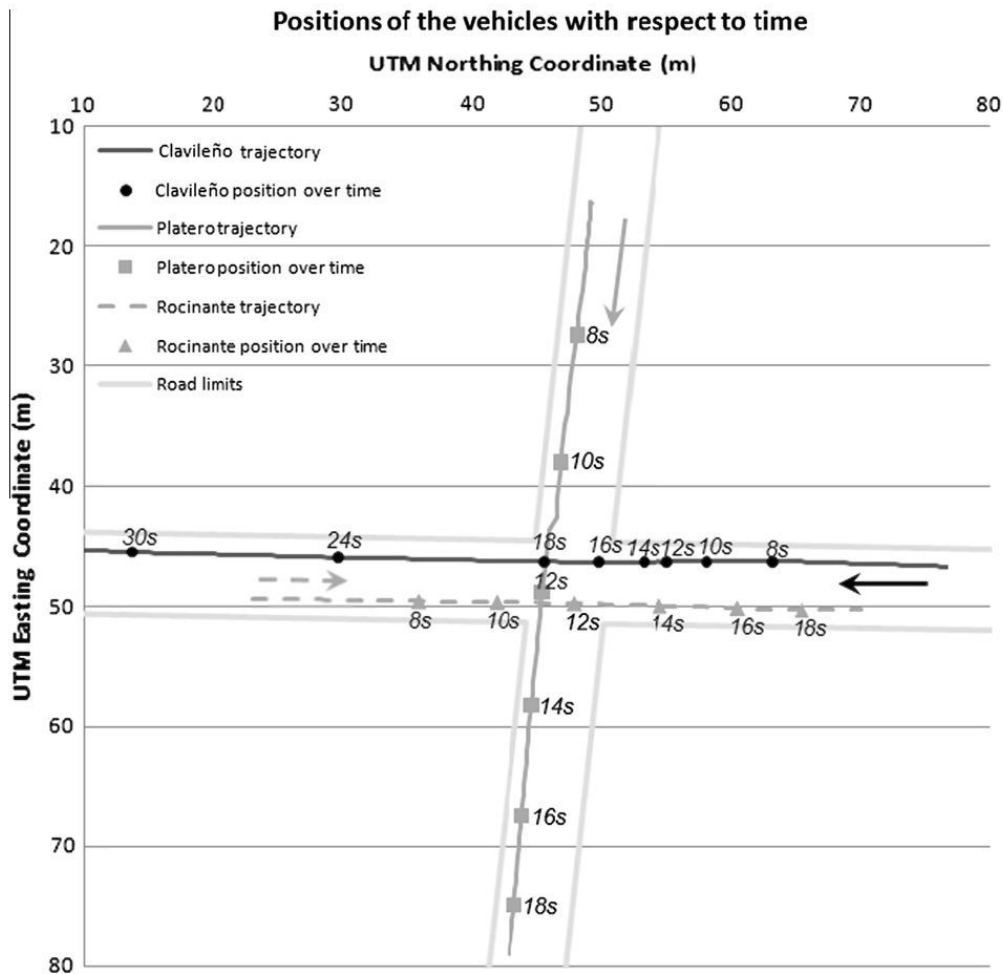


Fig. 15. Positions of the vehicles during the second scenario, using the first method of resolution: priority tables.

### 5.2.1. Scenario 1

This is the same scenario as in the first method of resolution. In this scenario (Fig. 10) according to the rules given in Section 3.2, the priority level of the reference vehicle is "N-", that of the vehicle approaching on its right is "N", and that of the incoming vehicle is "L+". The reference vehicle's priority is not the highest, so it yields. Once the vehicle with the highest level of priority has passed (the one approaching from its right), the new levels are: "N+" for the reference vehicle, and "L+" for the one approaching from the front. As our reference vehicle now has a higher priority, it moves on, and the vehicle approaching from the front must keep waiting.

Fig. 16 shows the evolution of the speeds of the vehicles over time. The first one to stop at the intersection is Platero. Later, Clavileño stops at the intersection. Rocinante continues its movement for a few seconds until it reaches the intersection. It slows down but then passes through without coming to a stop. Once it has left the intersection, Clavileño is the one that must continue. It starts again, and Platero is the last one to cross the intersection.

Fig. 17 shows, as in the previous tests, the position of the cars with some time stamps added. As in the first method, one observes how Clavileño stops while Rocinante passes, and then proceeds before Platero may enter the crossing point. The experiment confirms the expected behaviour of the automatic vehicle in this scenario.

### 5.2.2. Scenario 2

Again this scenario is the same as that of the first method (Fig. 11). The expected behaviour is, therefore, for the car to stop and wait until the other vehicles have passed. According to the rules in Section 3.2, the priority level of the reference vehicle is "N-", that of the vehicle approaching on its right is "N-", and that of the incoming vehicle is "N+". The reference vehicle's priority is not the highest, so it yields. Once the vehicle having the highest level of priority has passed (the one approaching from the front), the new levels are "N-" for the reference vehicle, and "N+" for the one approaching from its right, so the reference vehicle yields again. Once it is left as the only vehicle in the intersection, it can proceed and go through.

The diagram of speeds with respect to time (Fig. 18) shows how all three vehicles stop before entering the intersection. It is several seconds before Platero finally starts again. Once Platero has passed through the intersection, it is Rocinante who

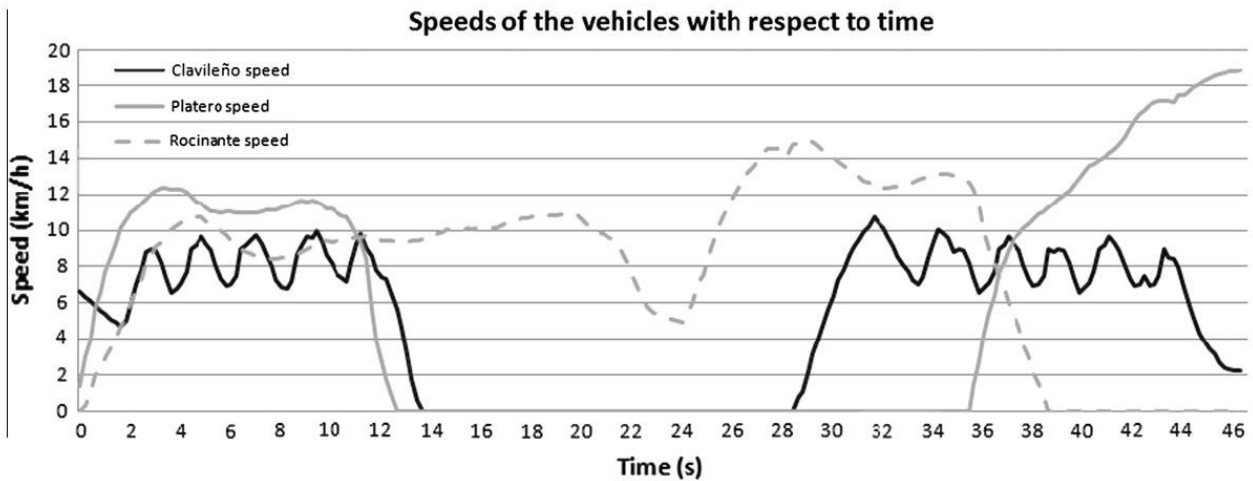


Fig. 16. Vehicle speed diagram for the first scenario, using the second method of resolution: priority levels.

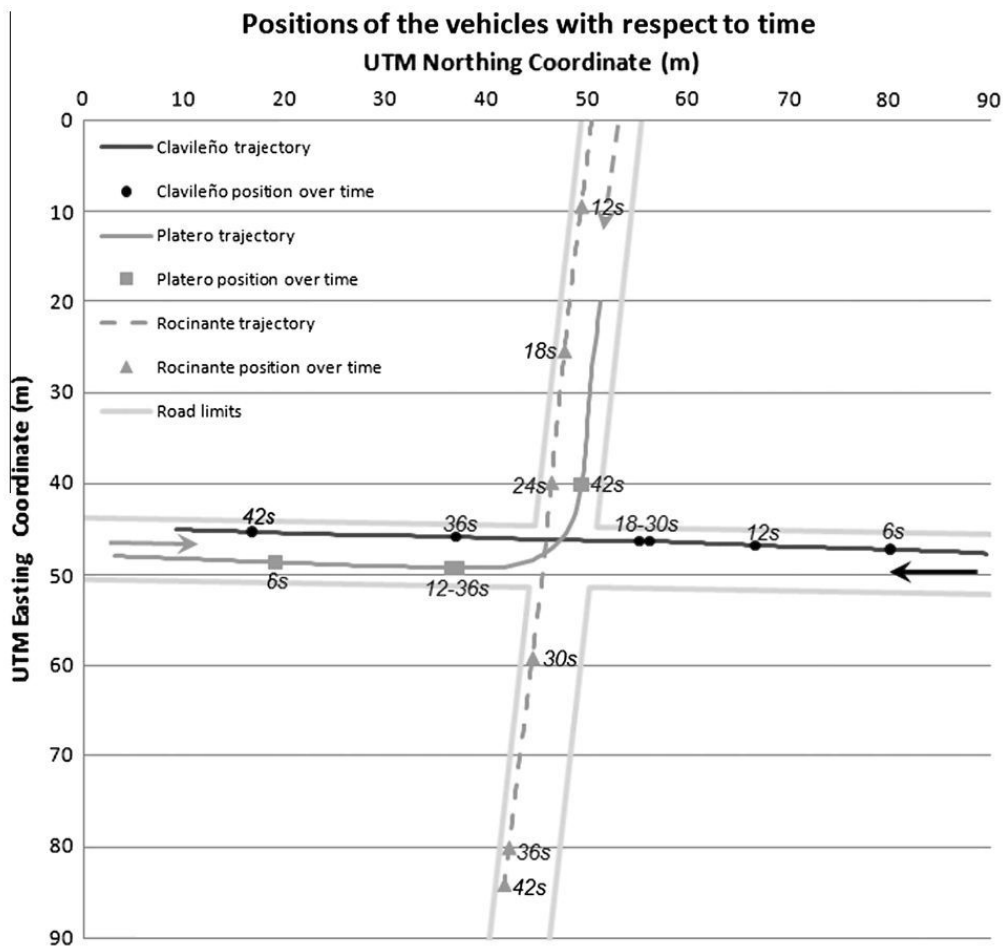


Fig. 17. Positions of the vehicles during the first scenario, using the second method of resolution: priority levels.

accelerates and goes on through. Once the intersection is free, Clavileño becomes the vehicle with the highest priority, and continues.

The positions and time stamps of the three vehicles are shown in Fig. 19. One observes how Clavileño yields to Rocinante, which in turn yields to Platero. The experiment confirms the behaviour expected of the automatic vehicle in this scenario.

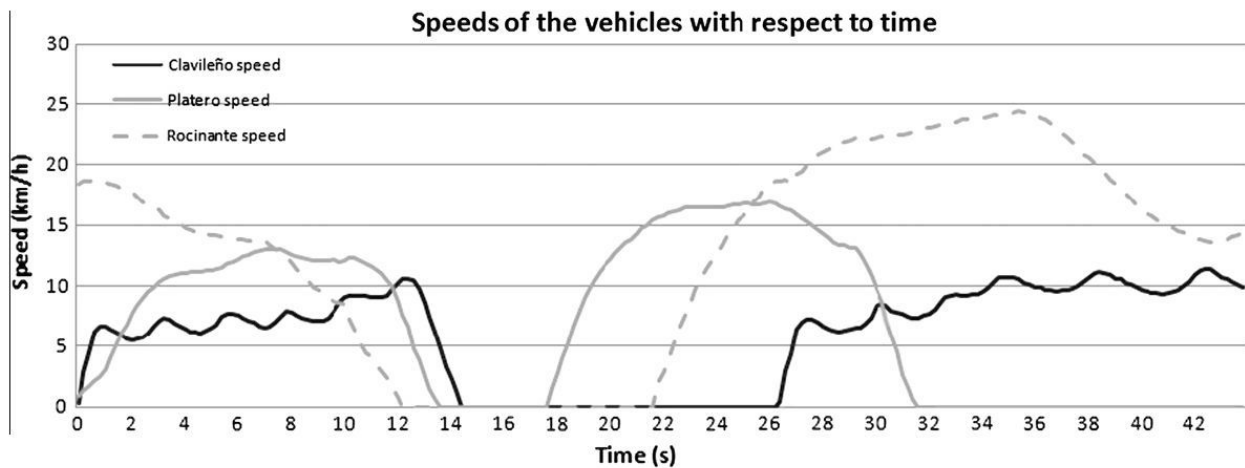


Fig. 18. Vehicle speed diagram for the second scenario, using the second method of resolution: priority levels.

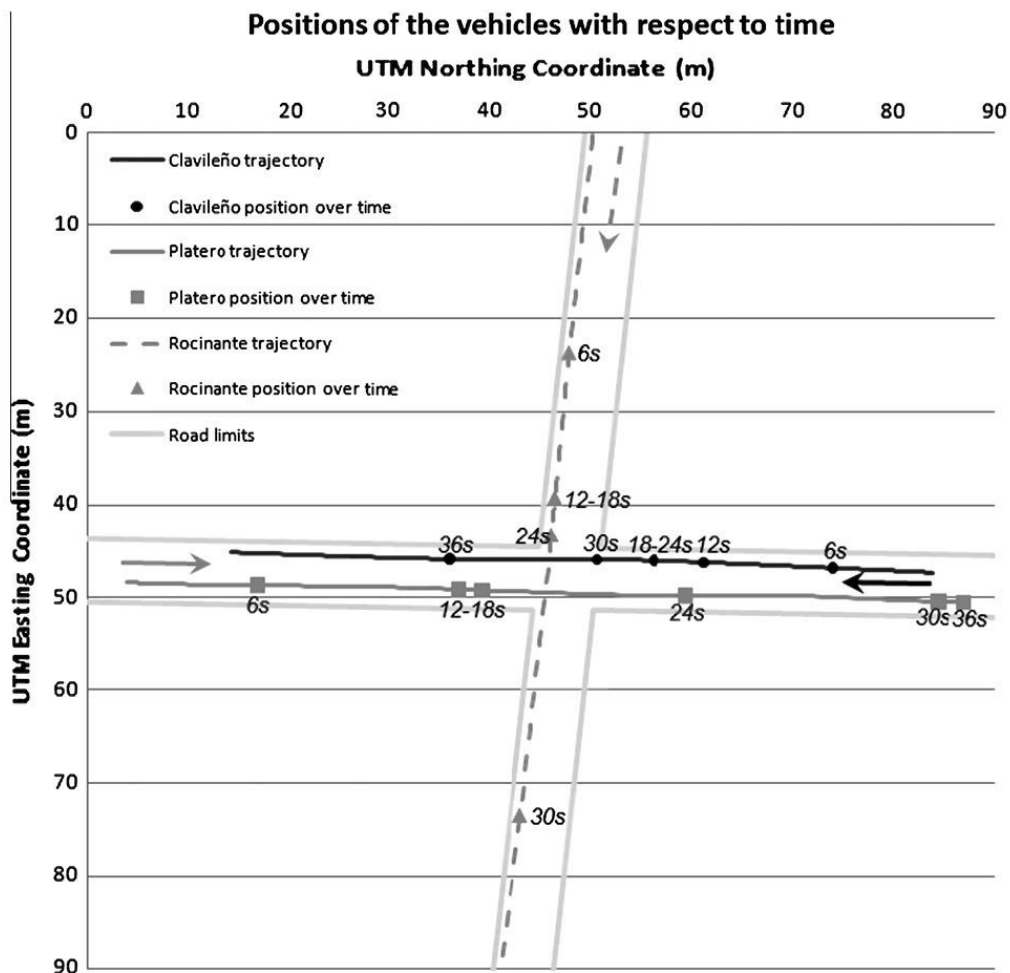


Fig. 19. Positions of the vehicles during the second scenario, using the second method of resolution: priority levels.

## 6. Conclusions

Two methods of resolution have been implemented and tested to deal with the problem of several vehicles approaching an intersection. Both methods are designed to be general for two-way roads, and are intended to be applicable to an unlimited number of vehicles. The differences between them are subtle. Their behaviour in the experiments was similar, since all they have to determine is whether or not the vehicle must give way, and they both produce the same output. The first

method uses priority tables, and all possible situations involving multiple cars must be encoded and associated with the action to be performed. This encoding has to be meticulous. The second method assigns priority levels to vehicles dynamically. Its programming is more complex since it not only queries the database, but also assigns a priority level to each vehicle involved. Because of this the second method has simpler data requirements and is therefore more reliable. The first method is fairly configurable, and any behaviour can be implemented by making as many exceptions to the rule as desired. The second case is wholly algorithmic, so that the behaviour is inherent in the rules. As stated, results of both methods are the same, so the reason for select one or the other in our case is based on the need to modify priorities. This can be done in the first method modifying the data base.

Two scenarios were used to test these methods. Their intention was to show that the vehicle is capable of determining whether to stop or continue, and when to make the decision. The results of the experiments for each method of resolution were consistent with the expected behaviour, thus confirming the validity of both methods.

While the focus of most other research groups has been on detecting the occupancy of the intersection by means of artificial vision, laser radar, etc., and traversing the intersection when the situation is safe, the focus of the present methods has been to seek to organize traffic in compliance with traffic regulations. Using the cartographic information of the intersection, which is known a priori, and obtaining by means of V2V communications the positions, velocities, and turning intentions of the other vehicles in the area, a decision is made as to whether the vehicles have to stop and give way, or may continue on their trajectory. In this way, a resolution of conflicts at intersections is provided that is comprehensive, consistent, and reliable.

## Acknowledgments

The authors express their gratitude to the TRANSITO (TRA2008-06602- C03-01) Project of the Plan Nacional, GUIADE (P9/08) Project of the Ministerio de Fomento, and to the MARTA Project (CENIT-20072006).

## References

- Alonso, J., December 2009. Cooperative Driving Manoeuvres Among Autonomous Vehicles. Ph.D. Thesis. Facultad de Informática, Universidad Politécnica de Madrid.
- Alonso, J., Milanés, V., Onieva, E., Pérez, J., González, C., de Pedro, T., 2011. Cartography for cooperative manoeuvres: autopia's new cartography system for cooperative manoeuvres among autonomous vehicles. *Journal of Navigation* 64, 141–155.
- Alonso, J., Serrano, N., de Pedro, T., González, C., García, R., 2005. Optimization of an autonomous car fuzzy control system via genetic algorithms. In: *I International Workshop on Genetic Fuzzy Systems*. pp. 101–106.
- Baber, J., Kolodko, J., Noel, T., Parent, M., Vlacic, L., March, 2005. Cooperative autonomous driving: intelligent vehicles sharing city roads. *IEEE Robotics & Automation Magazine* 12 (1), 44–49.
- Board, T.R., 2000. Highway Capacity Manual, fourth ed., Washington DC.
- Bouraooui, L., Petti, S., Laouiti, A., Fraichard, T., Parent, M., 17–20 September 2006. Cybercar cooperation for safe intersections. In: *Proc. IEEE Intelligent Transportation Systems Conference ITSC '06*. pp. 456–461.
- Campbell, M., García, E., Huttenlocher, D., Miller, I., Moran, P., Nathan, A., Schimpf, B., Zych, N., Catlin, J., Chelarescu, F., Fujishima, H., Kline, F.R., Lupashin, S., Reitmann, M., Shapiro, A., Wong, J., 2007. Team Cornell: Technical Review of the Darpa Urban Challenge Vehicle. Tech. rep., Cornell University.
- Dresner, K.M., Stone, P., 2008. Mitigating catastrophic failure at intersections of autonomous vehicles. In: *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, vol. 3. International Foundation for Autonomous Agents and Multiagent Systems, Estoril, Portugal. pp. 1393–1396.
- García, R., de Pedro, T., 2000. First application of the orbex coprocessor: control of unmanned vehicles. *EUSFLAT–ESTYLF Joint Conference. Mathware and Soft Computing* 7 (2–3), 265–273.
- Kolodko, J., Vlacic, L., 2003. Cooperative autonomous driving at the intelligent control systems laboratory. *IEEE Intelligent Systems* 18 (4), 8–11.
- Mamdani, E.H., 1976. Application of fuzzy logic to approximate reasoning using linguistic synthesis. In: *Proceedings of the Sixth International Symposium on Multiple-Valued Logic*. IEEE Computer Society Press, pp. 196–202.
- Milanes, V., Gonzalez, C., Naranjo, J., Onieva, E., De Pedro, T., 2010. Electro-hydraulic braking system for autonomous vehicles. *International Journal of Automotive Technology* 1 (11), 89–95.
- Milanés, V., Alonso, J., González, C., Bouraooui, L., Ploeg, J., de Pedro, T., 2011. Cooperative maneuvers in close environments among cycabs and dual-mode cars. *IEEE Transactions on Intelligent Transportation Systems* 12 (1), 15–24.
- Milanés, V., Naranjo, J., Gonzalez, C., Alonso, J., de Pedro, T., 2008. Autonomous vehicle based in cooperative GPS and inertial systems. *Robotica* 26 (5), 627–633.
- Milanes, V., Onieva, E., Vinagre, B., González, C., Pérez, J., Alonso, J., 2010. Sistema de asistencia a la conducción basado en una red de comunicaciones de bajo coste. *DYNA* 85 (3), 245–254.
- Naranjo, J.E., Bouraooui, L., Garcia, R., Parent, M., Sotelo, M.A., 2009. Interoperable control architecture for cybercars and dual-mode cars. *IEEE Transactions on Intelligent Transportation Systems* 10 (1), 146–154.
- Naumann, R., Rasche, R., Tacke, J., 1998. Managing autonomous vehicles at intersections. *IEEE Intelligent Systems* 13, 82–86.
- Onieva, E., Alonso, J., Pérez, J., Milanés, V., de Pedro, T., 2009. Autonomous car fuzzy control modeled by iterative genetic algorithms. In: *Proc. 2009 IEEE International Conference on Fuzzy Systems*. Korea, pp. 1615–1620.
- Onieva, E., Milanés, V., González, C., de Pedro, T., Perez, J., Alonso, J., 2010. Throttle and brake pedals automation for populated areas. *Robotica* 28 (4), 509–516.
- Perez, J., Gonzalez, C., Milanes, V., Onieva, E., Godoy, J., de Pedro, T., 14–17 April 2009. Modularity, adaptability and evolution in the autopia architecture for control of autonomous vehicles. In: *Proc. IEEE International Conference on Mechatronics ICM 2009*. pp. 1–5.
- Regele, R., 2008. Using ontology-based traffic models for more efficient decision making of autonomous vehicles. In: *ICAS '08: Proceedings of the Fourth International Conference on Autonomic and Autonomous Systems*. IEEE Computer Society, Washington, DC, USA, pp. 94–99.
- Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., Mahoney, P., 2006. Stanley: the robot that won the darpa grand challenge. *Journal of Field Robotics* 23 (1), 661–692.

Urmson, C., Anhalt, J., Bae, H., Bagnell, J.A., Baker, C., Bittner, R.E., Brown, T., Clark, M.N., Darms, M., Demitrish, D., Dolan, J., Duggins, D., Ferguson, D., Galatali, T., Geyer, C.M., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T., Kolski, S., Likhachev, M., Litkouhi, B., Kelly, A., McNaughton, M., Miller, N., Nickolaou, J., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Sadekar, V., Salesky, B., Seo, Y.-W., Singh, S., Snider, J.M., Struble, J.C., Stentz, A., Taylor, M., Whittaker, W.L., Wolkowicki, Z., Zhang, W., Ziegler, J., June 2008. Autonomous driving in urban environments: boss and the urban challenge. *Journal of Field Robotics* Special Issue on the 2007 DARPA Urban Challenge, Part I 25 (1), 425–466.

van Middlesworth, M., Dresner, K., Stone, P., May 2008. Replacing the stop sign: unmanaged intersection control for autonomous vehicles. In: *AAMAS Workshop on Agents in Traffic and Transportation*. Estoril, Portugal, pp. 94–101.