2010 13th International IEEE
Annual Conference on Intelligent Transportation Systems
Madeira Island, Portugal, September 19-22, 2010

Analysis of Reservation Algorithms for Cooperative Planning at Intersections

Arnaud de La Fortelle

***Abstract*— Intersections concentrate a too important number of accidents. It is quite obvious they are dangerous places because it is where cars potentially collide due to intersecting trajectories, as opposed to "normal" roads where they are parallel. This paper present a framework designed initially for cybercars (fully automated cars) but that could also be applied — though with major differences — to human driven cars. It is a world where vehicles have to reserve pieces of roads to cross a junction. This work is an enhancement of a previous work that demonstrated the feasibility of such a reservation algorithm.**

***Index Terms*— Automated road, autonomous vehicles, collision avoidance, cybercars, intersection control, path planning, reservation algorithm, traffic control**

I. Introduction

INTERSECTIONS are intrinsically dangerous because vehicles regularly have to cross each other: many trajectories intersect. There are two strategies to avoid collisions: either roads are changed so that trajectories do not intersect anymore *geometrically* e.g. with tunnels or bridges or ensure trajectories are sorted in time so that they do not intersect because of a proper *schedule* e.g. with traffic lights. Of course the first solution is safer but also more expensive and space-consuming. Therefore road planners have developed a large set of possibilities with rondabouts, shortcuts, stop signs, priorities, etc.

The first goal of these rules is to ensure a safe crossing: obeying all the rules (preferably simple) keeps the system safe. We do not want to treat here the case of breaking the rules, but the efficiency. All rules do not have the same efficiency. For example a 4 stops junction (in the USA) is clearly unsuited for 4–lanes urban arterial roads. A traffic light scheme is much more efficient; and is in place, of course. Now the question is: is it possible to do better than traffic lights? Even better than adaptive traffic lights (adapting to traffic flows)? With the same safety design: if every vehicle obey the rules, no accident can occur.

A first analysis shows that there is a margin in the optimization problem. A traffic light system allocates the road to flows (i.e. incoming directions), not to vehicles. Therefore a vehicle–aware system can do better since it can, at least, do the same as a traffic light system.

We introduced in [1] the use of a reservation algorithm in the framework of cybercars systems. In this work, an X–junction was considered and the problem was to ensure vehicles would not collide inside the intersection. Simulation proved this was possible *and* efficient with a reservation algorithm. This work validated the approach by simulation but did not formalize the concepts used. The aim of this paper is to present enhencements in the *conceptual framework* as well as in the *algorithms*. Finally an implementation is shown in simulation to demonstrate this solution is indeed practical — most notably real-time.

Mines ParisTech, Dep. Maths & Systems, Robotics Lab, 60 Bd Saint Michel 75006 Paris France

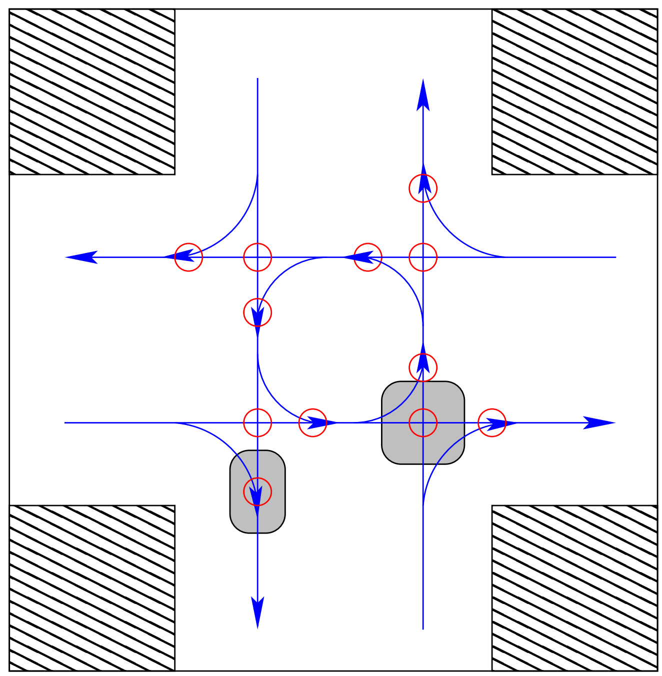

Fig. 1. Picture of a X–junction as presented in [1]. 2D-paths are predefined (blue lines) as well as reservation points (red circles) within each zone (two are depicted in gray) where collisions may occur.

II. State of the Art

The idea of exploring new techniques to control the traffic came up with the development of artificial intelligence. It was applied to improve systems of traffic lights and signals. In the early days of the fuzzy logic pioneers, [2] considered it to make the traffic lights more intelligent. A fuzzy logic controller for an intersection allowed to continuously adequate the traffic lights cycles to the traffic situation. This new fuzzy controller was coupled to a car detection system and adapted the timing of the traffic lights. The results showed an improvement of the efficiency and a reduction of the average delays. In the present, most of fuzzy logic controller and other techniques are used together. [3] proposes a cellular automata for tuning the parameters of the

978-1-4244-7659-6/10/$26.00 ©2010 IEEE

fuzzy logic controller. In fact, the main problem of all the traffic controllers is the great variations that traffic conditions can suffer. This kind of controllers needs information from the scenario, which is supplied by sensors (vehicle detectors) placed at the intersection [4].

Finite States Machines (FSM) emerged like a cheaper alternative to the fuzzy logic controller. [5] uses a statechart to modify the timing cycle of the traffic lights towards a more complex and efficient one. No sensors are required. A more sophisticated automata can be found in [6]. It uses a kind of temporal reservation instead of traffic lights to avoid collisions in a $\top$-shape intersection. There are other approaches to improve the efficiency: genetic algorithms or neural networks [7] can be used to learn intersection crossing patterns that are used to tune the traffic lights or to forecast the traffic flow.

At the end whatever the system, higher crossing rates involve higher speeds and shorter distances between cars. This causes clearly safety problems and one can not improve efficiency at safety's expense. Therefore we need a reliable safety system that guarantees the total absence of collisions. Because the great majority of accidents are caused by a human error, there is an easy solution: to remove the human driver from the driving tasks. We state it here in a simplistic way but it is a serious program for the future. There are some examples of driverless tubes working fine around the world and the aerial unmanned vehicles have already shown their utility. In the case of (cyber-)cars, the most promising approach are cooperative systems. For (current) vehicular systems, the state of the art for intersections is carried by the European Intersafe-2 project [8]; for cybercars the case of cooperation at crossroads has been studied and demonstrated during the European Cybercars-2 project [9].

Controlling the traffic flow is very complex unless we distribute the task among supervisor agents and vehicles. [10] introduces an agent in the crossroads to decide which traffic light configuration is the most appropriate at each moment and a multiagent approach has been extensively developed in [11]. Nevertheless, the translation into real applications is still very limited and tests involving real vehicles are rare [12].

III. Problem statement

In this paper we follow the same framework as in [1]. The following is a summary of this paper's problem statement.

Consider a transportation system with fully autonomous vehicles (or cybercars); a lot of tasks have to be executed to answer the demand of a customer to travel between two points. Our approach consists in decomposing the planning into three levels, each of which using only a relevant subset of the information, thus reducing the complexity:

- **the macroscopic level**, *e.g.* a city or a region;
- **the mesoscopic level**, *e.g.* a city quarter;
- **the microscopic level**, *i.e.* the surroundings of the cybercar.

At the macroscopic level, a *path* is computed. A path, is defined as a succession of edges (road segments) in the graph description of the road network. This is the level of fleet and roads management with a time scale ranging from half an hour to several hours.

At the mesoscopic level, paths are transmitted by the upper level and turned into *trajectories*. A trajectory is a precise space-time curve that the cybercar has to follow. Typical precisions are 10 cm and 1/10 s. The goal of this level is to produce trajectories for all the cybercars circulating in a given area. These trajectories have to be safe, mainly collision-free, but also efficient in terms of throughput (most notably deadlock-free).

At the microscopic level, the cybercar's control moves along the trajectory and ensures that none of these collisions which couldn't have been foreseen at the higher levels occur.

Our previous paper [1] validated the reservation approach as presented in [13], [14] as an appropriate answer to this problem. However simulations have shown that such an approach is not complete. Though you cannot have, by construction, collisions at junctions, it is however possible to have collisions before or after this place. So the first problem to be solved is how to avoid pushing collision to other places? The second problem is a question of efficiency. There are many ways to build reservation algorithms and to book reservation points. Is it possible to have an efficient algorithm? Even optimal? And the last question demonstrates the need for a better framework where optimality criteria could be defined.

IV. Limitation of previous approach

First implementation of the reservation algorithm proved the validity of the approach. However, after analysis of simulations, some drawbacks appeared clearly.

The most visible limitation concerns the existence of collision, before or after the crossroad. The explanation is simple: vehicles only interact through reservation within the intersection. Therefore there is absolutely no garanty there is no collision outside; and there are quite a few when incoming flows are important.

A second very visible drawback is about speed. Vehicles indeed cross the intersection at low speed, especially when flows are important. The explanation lies in the "distributed" nature of the first reservation algorithm. When a vehicle approaches the intersection, it requests a set of reservations (for all crossed reservation points) corresponding to a trajectory with its current speed. If this request fails, the vehicle do another reservation request with a slightly lower speed, iteratively, until there is positive answer (in fact the implementation is a bit more complex but this is the principle). It is therefore predictable that vehicles tend to cross the junction at low speed, and the more incoming vehicles, the slowest.

Last limitation, the reservation algorithm itself (when a vehicle books reservation points) was not very efficient. With few vehicles it ran efficiently but became excessively slow as soon as there were more vehicles. The description of the algorithm in the above paragraph explains it. Since requests are generated by the vehicles themselves, the more occupied is the crossroad, the less chance the request have to be satisfied. Therefore vehicles tend to send more and more request in a congested environment. Analysis of simulations shown this could be up to several hundreds of requests until one is satisfactory.

V. Optimization of reservation requests

Since the reservation algorithm is really at the heart of our approach, our first goal was to analyze more in-depth the problem of reservation request. Moreover the simulator from [1] was implemented with some visualization features and it helped a lot understanding the underlying constraints and degrees of freedom. It was also clear that it was better to first design carefully a new reservation algorithm and, in a second step solve the problem of collision outside the crossroad since the main constraint comes from reservations.

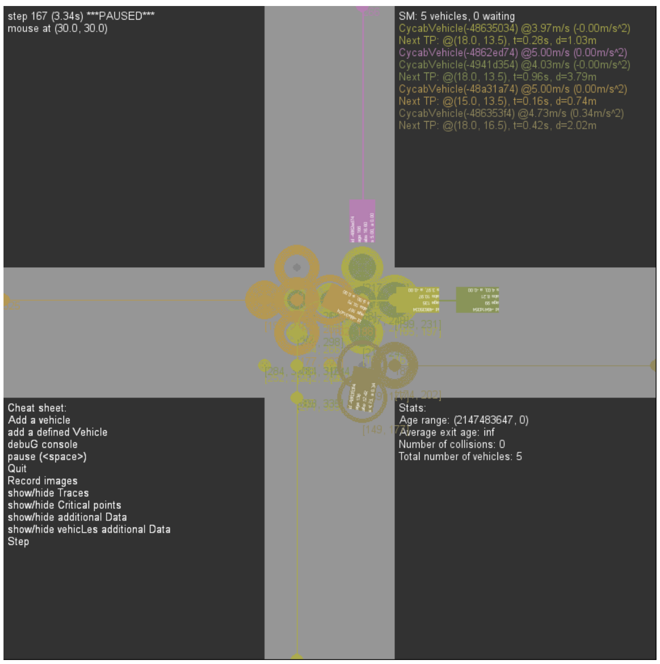

Fig. 2. Simulation at a X-type junction. Each reservation point is represented with colored anneals. The color represents the vehicle and the radius represents the time: center is the future and exterior present. Hence, in such a picture one can read both trajectories and successive reservation at a reservation point.

Figure 2 shows a typical situation. Considering both trajectories (i.e. space-time paths) and reservation points, the reservation problem appears as the matching of availabilities of both vehicles and intersection. However, problem is not symmetrical. A vehicle can (in most cases) wait while intersection has a limited availability, particularly in high traffic cases where the algorithm should be efficient.

Since the crossroad is the rare resource, best use it efficiently, hence spend as little time as possible in the junction. This turns out to impose to cross the intersection at maximum speed. One can imagine have different maximum speeds depending on the kind of trajectory (straight or turning). Here we took a unique maximum speed. Since this speed is known for all vehicles, it is much easier to calculate possible entrance times for a vehicle: at these times, all the needed reservations points are available. Then take the minimum such entrance time that is compatible with the vehicle speed on entrance lane and the vehicle gets the best time to enter the crossroad.

Note that such an algorithm is very different from the previous approach. Indeed, we switch from a rather distributed algorithm (the intersection only checks vehicles requests) to a more centralized algorithm where the intersection calculates the best arrival time for all requests. Since analysis and implementation have shown that processing power required for such calculations is limited, we decided to keep this algorithm. It has indeed many advantages: there is only one request per vehicle entering the junction. Due to maximum speed, we believe it is much more efficient from a traffic point of view. And finally it keeps the fundamental property of the reservation algorithm: no collision can occur within the junction. So we solved two limitations: low speed within intersections and the too numerous requests.

VI. Collision avoidance outside of the intersection

The second improvement concerned the collisions that occurred outside of the intersection. We analyzed this was due to non-interacting vehicles. A simple solution is to set up a car-following control loop, well known in the automotive industry as Advanced Cruise Control (ACC). Real ACC are pretty complex to take into account complexity of real life but a first approximation can be to set the acceleration of a vehicle at a safe level considering interdistance with previous car and speeds of both vehicles.

Unfortunately, this completely breaks the reservation algorithm. Consider a case where a vehicle gets its reservation for the soonest arrival time. If it has to slow down because of another vehicle in front of him, it will fail to enter into the junction at the right time. This shows one assumption was false: a vehicle cannot always reach the intersection entrance at any desired time after a given date.

Fortunately enough, this assumption remains true for the first vehicle in the entrance lane. This is why we did the following slight modification: only the first vehicle in a lane can request the crossroad for a reservation. Following vehicles remain under control of an ACC.

Now we have to ensure any vehicle can pass by the intersection at full speed. This means you should not stop right at the intersection entrance (it would yield an infinite acceleration). Therefore the entrance lane is divided in two pieces: there is a small “acceleration lane” just before the intersection; the rest is “normal”. Vehicles have five possible states linked to the kind of lane there are on. The following is true simultaneously for each lane.

1) In state 0, a vehicle is on a normal road and it is controlled by its ACC.
2) The first vehicle with state 0, when the vehicle in front of him (if any) switches to state 2, sends a request and

adapt its speed to arrive at the crossroad's entrance on due time. It switches to state 1 meaning it not controlled anymore by ACC but it is driven by its arrival time.
3) When a vehicle with state 1 arrives sufficiently close to the intersection, onto the acceleration lane, it switches to state 2. This happens only when the following vehicle cannot collide with him: then the following vehicle (state 0) switches to state 1.
4) Then the vehicle switches to state 3 when it enters the crossroad, where it is driven by the reservation algorithm (here: constant maximum speed).
5) After the intersection, the vehicle switches to state 4 which is an ACC state equal to state 0.

Note that, in order to prevent a vehicle to enter the acceleration lane with state 0, the beginning of the acceleration lane is considered as an obstacle in state 0. Therefore, in the worst case, a vehicle in state 0 will stop at the beginning of the acceleration lane until it gets a reservation.

With such laws, collision disappeared from simulations. In fact, when running sufficiently long the simulator with special parameters, it happened some collisions occurred right after the crossroad, but very seldom. A careful analysis shows this can happen. Indeed, reservation completely prevents collisions within the crossroad and ACC prevents collision outside. However transitions between states were carefully designed only *before* the intersection, not after. If two vehicles exit the crossroad on the same lane. It may happen (this is indeed rare) that the first vehicle need to slow down in the exit lane so that the follower, once arriving at the exit point, cannot avoid it since required acceleration (defined by ACC) is above the limit.

This problem was not solved during this study. It is in fact related to several difficult problems. First it is the question of transitions between closed loop control (*e.g.* ACC) and open loop control (*e.g.* reservation points). How can we ensure complete consistency regarding collision between both families? Second, slow down at exits are related to traffic management within roads: if a road is jammed, the intersection cannot exit to this road. This means the next intersection (creating the congestion) should privileged this congested road. How can we design reservation policies with priorities? This problem seems related to polling systems.

Here we see that the question of reservation algorithms for cooperative planning at intersections raises fundamental questions. These questions are currently analyzed in our team to provide enhanced algorithms.

VII. Application

The last part of this work was related to a real demonstration. Vehicles were provided by the European Cybercars 2 project: there were several fully automated vehicles (cybercars and automatized cars, see Figure 4). For the sake of simplicity, the demonstration track was drawn with a unique intersection, as an eight-shape circuit. So we analyzed a *closed* system consisting in a eight loop rather than an open X–junction and we adapted the simulator to this case (see Figure 3). There were several important modifications to do (e.g. system is closed so that trajectories are infinite) and though these changes, the fundamental properties of our algorithms were kept.

Due to system requirements, the implementation of simulator (python-based) were translated into a more robust C++ implementation. This new reservation module was in fact dedicated to this special case but we can claim that our algorithms also served as a basis for a real-time implementation that ran during combined CityMobil & Cybercars 2 demonstration in the French city of La Rochelle in 2008. A video is available on YouTube (`www.youtube.com/v/JkS0D__Vz3o`). It shows that these algorithms are compliant with real application.

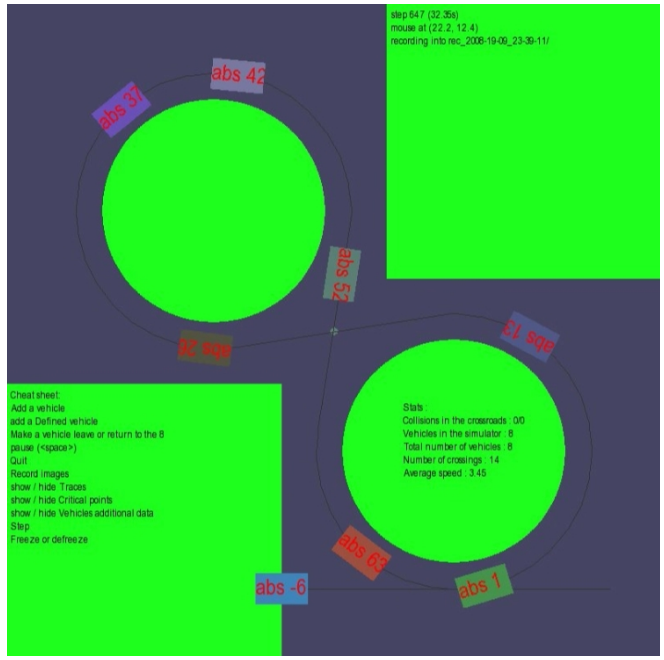

Fig. 3. Simulation of a eight circuit. This was the shape of a real demonstration track set up in 2008 in La Rochelle.

Fig. 4. Cooperative intersection with fully automated vehicles demonstrated at La Rochelle, France, in 2008, using a reservation algorithm.

VIII. Conclusion and future work

We presented here extensions of a first work [1] that proved a reservation algorithm could be used for intersection management in the framework of a fully automated and supervised system. Several enhancement were implemented:

- a simpler and much more efficient reservation request system using a more centralised approached than the previous one and consuming less bandwidth;
- this enhancement in turn increased the average speed of the vehicles at intersection, hence improving the efficiency of the system in term of flows;
- we introduced five states related to vehicle control to manage the transition between lines where vehicles are controled by an ACC algorithms and the crossroad where a trajectory is fully specified. This solved the problem of collision occuring sometimes before the intersection;
- a real experiment with an implementation derived from this work was presented.

These enhancements show the potential for future optimization. There are first problems to be solved, such as remaining collision after the intersection or tests with more complex junctions. Indeed, this work, as well as previous work, is only a proof of concept. Our work was mainly an answer to a concrete problem (how to ensure cybercars do not collide at an intersection?) that was raised during the Cybercars-2 European project. Since it was related to other problems (e. g. driver assistance at intersections), we tried to do a more precise analysis of the requirements and developed this solution that seemed to be promising. We believe the set of algorithms presented here is fundamental for this approach: building a model of the cross-road (with clothoids or straight lines as virtual rails); building reservation zones; managing the reservation zones on request by vehicles; controlling vehicles using several control states to manage different stages: far, next to or within the junction.

But we have now only begun the mathematical analysis of these algorithms and their interdependencies. For example it is not completely clear how control states should be adapted to solve the problem of residual collisions at the exit of the intersection; how the geometry is linked to command laws; and how geometry and control is linked to reservation zones. A better understanding would certainly allow further optimization. But the main point is to obtain *proofs*. Our claim is to develop a system that is safe by design meaning we are able to prove that, under suitable assumptions, there will be no collision. And what would be even more satisfactory would be a system where each vehicle has a *certificate* of no collision, meaning reservation algoritms produce an information (maybe with complicated algorithms) easily checkable (i.e. with fast algorithms) between vehicles that ensures the safety. This work is currently undergoing.

Aknowledgement

This work has been partially funded by the European FP6 project Intersafe 2. The implementation of new features was carried out during the internship of Mines ParisTech student Colin Valière in collaboration with Universidad de Las Palmas de Gran Canaria (Prof. Gustavo Marrero Callico).

References

[1] O. Mehani and A. de La Fortelle, “Trajectory planning in a crossroads for a fleet of driverless vehicles,” in *11th International Conference on Computer Aided Systems Theory (EUROCAST 2007)*, Las Palmas de Gran Canaria, Spain, february 2007, pp. 1159–1166.

[2] C. P. Pappis and E. H. Mamdani, “A fuzzy logic controller for a traffic junction,” *IEEE Transactions On Systems, Man, And Cybernetics*, vol. 7, no. 10, pp. 707–717, october 1977.

[3] M. Shakeri, H. Deldari, A. Rezvanian, and H. Foroughi, “A novel fuzzy method to traffic light control based on unidirectional selective cellular automata for urban traffic,” in *Proceedings of 11th International Conference on Computer and Information Technology (ICCIT 2008)*, Khulna, Bangladesh, december 2008, pp. 300–305.

[4] G. H. Kulkarni and P. G. Waingankar, “Fuzzy logic based traffic light controller,” in *Second International Conference on Industrial and Information Systems (ICIIS 2007)*, Sri Lanka, august 2007, pp. 107–110.

[5] Y. Huang, S. Lee, and Y. Liu, “A supervisor of traffic light systems using statecharts,” in *2007 IEEE International Conference on Networking, Sensing and Control*, London, UK, april 2007, pp. 862–867.

[6] Q. S. Wu, X. B. Li, M. B. Hu, and R. Jiang, “Study of traffic flow at an unsignalized T-shaped intersection by cellular automata model,” *European Physical Journal B*, vol. 48, no. 2, pp. 265–269, november 2005.

[7] A. Čivilis, “Prediction of crossroad passing using artificial neural networks,” in *7th International Baltic Conference on Databases and Information Systems*, 2006, pp. 229–234.

[8] “Intersafe-2,” 2008–2011, EC Research project. [Online]. Available: http://www.intersafe-2.eu

[9] “Cybercars-2,” 2006–2008, EC Research project. [Online]. Available: http://www-rocq.inria.fr/cybercars2

[10] V. Hirankitti and J. Krohkaew, “An agent approach for intelligent traffic-light control,” in *Proceedings of the First Asia International Conference on Modelling & Simulation (AMS’07)*, 2007.

[11] K. Dresner and P. Stone, “A multiagent approach to autonomous intersection management,” *Journal of Artificial Intelligence Research*, vol. 31, pp. 591–656, march 2008. [Online]. Available: http://www.jair.org/papers/paper2502.html

[12] L. Bouraoui, S. Petti, A. Laouiti, T. Fraichard, and M. Parent, “Cybercar cooperation for safe intersections,” in *2006 IEEE Intelligent Transportation Systems Conference (IEEE ITSC 2006)*, Toronto, Canada, february 2006, pp. 456–461.

[13] K. Dresner and P. Stone, “Multiagent traffic management: A reservation-based intersection control mechanism,” in *The Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 04)*, New York, USA, july 2004, pp. 530–537.

[14] ——, “Multiagent traffic management: An improved intersection control mechanism,” in *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 05)*, Utrecht, The Netherlands, july 2005, pp. 471–477.