Cooperative Intersection Management: Using mini-robots to compare sequenced-based protocols

Florent Perronnet, Abdeljalil Abbas-Turki, Abdellah El-Moudni

IRTES-SET, Université de Belfort-Montbéliard, 90000 Belfort
florent.perronnet@utbm.fr

Abstract—Traffic lights are the most common method to control the traffic, in particular in urban areas. Recent researches have proposed a new way to manage intersections; the method is called Cooperative Intersection Management (CIM). In this paper we present an experiment using mini-robots in order to compare two main approaches of CIM based on the sequence construction. One of them is mainly adapted for human drivers whereas the other requires fully autonomous vehicles. We also test different policies for each protocol. The paper presents the experiment results which prove that it is possible to improve the traffic fluidity compared to classic traffic lights with both protocols.

I. INTRODUCTION

Many recent contributions in the field of intersection management are based on vehicles equipped with wireless technology and positioning systems. Some of them enhance traffic lights by exploring the additional communicated data [5, 10] whereas the others are based on a new approach for controlling the intersection [1-4, 6-8, 12-13]. We call the new approach Cooperative Intersection Management (CIM). CIM is neither a cooperative traffic light nor a cooperative collision warning system [11]. More precisely, it considers an intersection of predefined routes along which vehicles move with a known origin-destination. The vehicles and the infrastructure are able to wirelessly communicate together for the negotiation of access to the intersection. The right-of-way, that is a kind of “green”, is obtained by each vehicle according to its state. The literature considers both ordinary and unmanned vehicles. For ordinary vehicles, the right-of-way is an instruction resulting from the received messages. The main objective of CIM is to provide an efficient intersection by means of a control policy that determines the sequence of vehicles. Thus, CIM is an efficiency-oriented concept, which provides a more precise control than the traditional traffic lights since the access of each vehicle is controlled individually instead of controlling the traffic through the durations of green phases.

In the absolute, a more precise control of the traffic implies better performances. Nevertheless, there are several obstacles that raise theoretical and practical challenges for achieving the expected performances. Indeed, CIM is based on both protocol and control policy that need to be correctly defined.

The protocol determines the way in which the potential zones of collision are shared by conflicting vehicles or flows. Theoretically, time can be saved if two conflicting vehicles cross the intersection within a tight interval of time. Nevertheless, the required time for freeing the potential zones of collision from a conflicting vehicle cannot be underestimated without increasing the risks of collision. This collision risk depends not only on the control of vehicle’s motion but also on the accuracy of positioning system, sampling times, communication delays and so on.

The control policy determines the sequence of vehicles that traverse the intersection. Due to the dynamic nature of the traffic, the decision whether or not a given vehicle has the right-of-way must be taken in a very short time. In other words, the time required for detection and taking the suitable decision should not delay the departure of vehicles. Moreover, an optimal sequence of access to the intersection at a given time is only a local optimality that does not consider the next incoming vehicles. For these reasons, many works propose a control policy that relies on simple rules or quick heuristics [1-3, 6, 9].
A few contributions propose a reservation-based protocol [7, 12] whereas a few others define a fully-decentralized protocol [13]. In the following we will limit our investigation to a sequence-based protocol by means of a centralized intersection. In the sequence-based protocol, for each conflict point there is a stringent order of vehicles. The sequence is determined through the negotiation with the server. The collisions are avoided by means of the default deny rule. In very simple words, vehicles must stop prior the conflict zone, if they are not considered in the sequence. The scope of the paper is to compare two sequence-based protocols through an intersection of mini-robots. The first protocol is CVAS (Cooperative vehicle-actuator system) that has been tested for ordinary vehicles [3]. The second one is TIM (Transparent Intersection Management) in which vehicles are controlled according to an Adaptive Cruise Control. The main objective of this paper is to assess the possible gain of time by using unmanned vehicles, through a framework of mini-robots. The NXT robots for this purpose. This allows to check the feasibility as well as to provide interesting observations. Indeed, many simulations have been proposed to assess the performances of the control system [1], but they must be interpreted through a framework that highlights real-world problems.

This paper is organized as follows. First it introduces the two protocols CVAS and TIM, and their respective policies. Second it describes the experiment framework. Third, the experiment results are presented and analyzed. Finally, we discuss the efficiency of both protocols and policies before concluding on the possible gain of time.

II. PROTOCOLS

In this paper we consider two sequence-based protocols. The main difference between both lies in the interpretation of the default deny rule. In CVAS the vehicle is not allowed to get through the conflict zone, until it gets the right of way whereas in TIM, the vehicle is allowed to move if it appears in the sequence sent by the server, but the movement is limited by the precedent vehicle. From practical point of view, these two interpretations imply the following differences:

- **Signalization**: The signalization is discrete in CVAS. That means each vehicle considers two states, which are green and red communicated to the driver agent. In TIM, the signalization is continuous. The driver-agent receives the kinematic data of the precedent vehicle in the sequence.
- **Driver agent**: In CVAS, the driver agent can be either a human or a control system whereas TIM is limited to controlled vehicles.
- **Server duty**: In CVAS, it is enough that the server considers only the batch of vehicles that can cross together the intersection with the initially authorized vehicles [1]. In TIM, the server must build
a sequence of all present vehicles in the intersection. Before describing each protocol we briefly present the intersection architecture.

A. Intersection architecture

Both protocols are based on a client-server model. Indeed, several clients can communicate with the server in order to negotiate the right-of-way. The studied intersection is given in [3]. We have defined a specific split of the intersection into three zones; the storage zone, the conflict zone and the exit zone (Fig. 1). The vehicles move from the storage zone to the exit zone.

Each vehicle begins to negotiate with the server as soon as it enters into the storage zone. If the communication has a problem and the server is not able to send a message to the vehicle, it must stop before the conflict zone. We will now describe each protocol and their respective policies.

B. Cooperative vehicle-actuator system

During its journey each vehicle has to cross successively each zone in the order (Fig. 2). Firstly a vehicle enters into a storage zone with the default deny. It follows the vehicle knows its lane and is able to inform the server that it requires a right-of-way. The red sign forces a client to stop before the conflict zone. Once the server sends a right-of-way to the vehicle, it is authorized to reach the conflict zone followed by the exit zone. A vehicle in the departure state sends messages to the server in order to give back the right-of-way.

The role of the server is to construct the presence list according to the received messages. Then an algorithm considers vehicles so as to define the sequence of passage. Messages containing rights-of-way are sent to concerned vehicles. We have defined two main policies. The first one is called FCFS (First Come First Served) and we call the second one DCP (Distributed Clearing Policy).

1. First come first served

For the FCFS algorithm, the server considers the exact order of the arrived vehicles. Actually the received messages determine the sequence. Thus it could happen that a delayed message implies that two vehicles are considered in the wrong order, which generates a deadlock. When messages arrive at the right order we can apply exactly FCFS, otherwise we must detect and solve the deadlock. Fortunately that critical situation can be detected easily and solved by observing the position of each vehicle. The server is capable of allowing two vehicles in the sequence, in order to fix the problem.

2. Distributed clearing policy

The DCP algorithm considers vehicles as convoys according to the intervehicular distance. Indeed we have proven that for CVAS, it is more efficient to give the right-of-way to a follower vehicle which is near another authorized vehicle because it is costly in terms of time to switch from one lane to another [1].

C. Transparent intersection management

In the TIM protocol, we consider vehicles equipped with an adaptive cruise control system able to adapt the vehicle speed according to the observed frontal obstacle. The speed of vehicle is determined by the temporal distance $h$ that separate the vehicle from the obstacle. This temporal distance is evaluated through the sensors such as embedded radars in front. TIM considers that for each vehicle there are two kinds of obstacle, i.e. real and virtual. Each vehicle has a preprogrammed virtual obstacle that is the end of the storage zone. The other virtual obstacles are added by the server through the wireless communication. The real obstacle is the one detected by the embedded sensor. The server computes the sequence then it sends it to vehicles. Each vehicle is able to read its position.
and the position of the precedent vehicles in the sequence. According to their distance from the exit zone, it determines its speed $V(h_r, h_{pv}, h_{sv})$. $h_r, h_{pv}$ and $h_{sv}$ are respectively the real headway observed by embedded sensor, the preprogrammed virtual headway which is the distance of the robot from the end of the storage zone and the virtual headway resulted from the positions of precedent robots in the sequence communicated by the server. Hence, $h_{sv}$ is determined by the sequence. All precedent vehicles are considered. If the vehicle is not considered by the server, i.e. not in the sequence, we have:

$$V(h_r, h_{pv}, +\infty) = \min\left(v(h_r), v(h_{pv})\right)$$  \hspace{1cm} (1)

When the vehicle is in the sequence, we have considered two TIM policies.

1. **Classical TIM**

   In the classical TIM, when the vehicle appears in the sequence sent by the server, it considers $h_{pv} = +\infty$. In other words, the vehicle removes the preprogrammed virtual obstacle from its list of obstacles, whereas it adds new all precedent vehicles in the sequence as obstacles. It computes its speed as follows:

   $$V(h_r, +\infty, h_{sv}) = \min\left(v(h_r), v(h_{sv})\right)$$ \hspace{1cm} (2)

   Where

   $$v(h_{sv}) = \min\left(v(h_{sv_1}), ..., v(h_{sv_k})\right)$$  \hspace{1cm} (3)

   By considering $k$ precedent vehicles in the sequence and $h_{sv_i}$ is the virtual obstacle created by the $i^{th}$ vehicle in the sequence.

2. **Advanced TIM**

   In the advanced TIM, the difference lies on the consideration of the end of the storage zone. In the classical TIM, each vehicle tries to adapt its speed according to precedent vehicles. If the intersection is congested, vehicles can stop at the beginning of the storage zone. Unfortunately space is lost. In advanced TIM we want to improve the use of the space by allowing vehicles to move until the end of the storage zone, in the case of congestion. Thus each vehicle computes its speed as follows:

   $$V(h_r, h_{pv}, h_{sv}) = \min\left(v(h_r), \max\left(v(h_{pv}), v(h_{sv})\right)\right)$$ \hspace{1cm} (4)

### III. EXPERIMENT FRAMEWORK

#### A. NXT Robot

Each robot is equipped with an intelligent brick, a color sensor, an ultrasonic sensor and motors (Fig. 3).

![NXT Robot](image)

The intelligent brick is capable to communicate with other bricks by Bluetooth communication. A program is executed by the brick to negotiate a right of way or to compute priority functions. The color sensor is used like a positioning system by reading a specific road marking. In addition each motor gives information to the robot which permits to measure the traveled distances. The ultrasonic sensor is placed in front of the robot for providing the inter-robots distance which is especially used by a follower robot ($h_r$ estimation), in order to adapt its speed. A robot has also the knowledge of the intersection architecture ($h_{pv}$ estimation) in order to help the motion control system for stopping before the conflict zone.

#### B. Test-bed

We have realized a test-bed using white paper and blue paper so as to provide a road marking (Fig. 4).
The white color represents the base color of the road. The blue stripes provide position markers which are used to delimit the different zones (storage, conflict and exit zones). Thus each robot is able to know its position at each found stripe.

C. Conducted experiment

Each protocol presented before has been tested through our test-bed with its corresponding policies. As you can see on the picture (Fig. 4), we have three robots distributed on two lanes, robots 1 and 3 on lane 1 and robot 2 on lane 2. For each test we start in that exact order: robot 1 → robot 2 → robot3. The experiences have been successfully repeated many times without collision. We have filmed tests in order to be able to analyze the results. We have kept significant videos for each combination of protocol / policy in order to determine important times by a precise observation. We will now present the obtained results.

IV. EXPERIMENT RESULTS

The following results have been obtained by using the observation of each robot in the videos. Indeed, we have recorded the arrival time in each blue stripe so as to analyze the journey of each robot during the experiment. Thus, we are able to know the occupation time of each zone by each robot and to compute many other useful times like evacuation times. Firstly we will explain the switch time and the inter-robots time which are involved in the obtained evacuation time. Then we will present the comparison between evacuation times for each protocol and their policies.

A. Inter-robots and switch times

Because of the small conflict zone, for this particular test-bed, only one vehicle at a time can be present in the conflict zone, it follows that we have a transition time between two vehicles even if they are on the same lane. We define the time space \( d \) which represents the inter-robots time between a robot and its follower on the same lane. We define also the space time \( s \) between two robots from any two conflicting lanes. The time \( s \) includes the time for crossing the intersection, the acceleration time and the security time. We have measured these times by observation of the videos. It results that the average crossing time is about 2.7 seconds and the acceleration time is insignificant. The figure below shows the average transition times (Fig. 5).

We can observe that in each test the inter-robots time is around 2.8 s, this time results from the parameters of the car-following model used, which is based on the ultrasonic sensor. The parameters are defined for offering a good security distance.

We have a significant difference of about 2 seconds between the CVAS protocol and the TIM protocol for the switch time. This difference is mainly due to the communication delay, indeed we have observed a time of at least 1 second so that a brick receive at least one message from another. In CVAS the robot which has the right-of-way sends a message to the server before the server sends a message to the new authorized vehicle, so we have an average time of at least 2 seconds for the transition. Moreover we know that the crossing time is about 2.7 seconds, so we have about 1 second for the security time.

The results between the policies of TIM are similar because the acceleration time is almost zero,
so even if a vehicle is stopped it is able to reach its maximal speed quickly.

The difference between the two policies of CVAS is only due to a slight difference on the communication delay measured during the tests, but this difference is theoretically inexistente.

B. Evacuation time

The evacuation time is the measured time between the arrival of the first vehicle and the exit of the last vehicle. The evacuation time is composed of the sum of travel times of robots. The travel time of each robot grows according to the presence of the others. Indeed a follower-robot will lose at least the minimum inter-robots time $d$ whereas a conflict-robot can lose at least one switch time $s$. Moreover we know that we have exactly two transition times during each experience because we have three robots.

As we can see on the figure (Fig. 6), the best protocol in terms of efficiency is TIM with about 12.7 seconds whereas CVAS is between 14.4 and 16.7 seconds. The reason is the communication time required for the CVAS protocol in order to ensure the safety of the intersection. Indeed CVAS is designed for a human use while TIM needs fully automated vehicles. Bluetooth communication for these tests is very slow compared to other potential communication systems, so it is possible to improve CVAS to have results close to TIM if the communication time is gained.

We observe a significant improvement for DCP compared to the FCFS algorithm. Indeed the FCFS method for this particular order of arrival forces two switches while DCP permits to replace one switch by an inter-robots time. As we know the inter-robots time is less than the switch time, so we are sure to reduce the evacuation time in many situations with possible follower vehicles.

The results between the two policies of TIM are very close because of the insignificant acceleration time. Nevertheless, the advanced TIM allows improving the use of the storage zone.

V. Discussion

From the result given in the previous subsection, we can derive the following observations:

- Regardless communication delay, the results of DCP are very interesting and are close to TIM results.
- Whatever the TIM policy, we have $s > d$, because two conflicting robots need a time to adapt their speed if they are in conflict and if they arrive at the same time. For this reason, it is important to do not limit TIM to only FCFS and to extend the policy to other scheduling algorithms.
- Classic and advanced TIMs provide similar performances. The advantage of the advanced TIM in terms of space is not altered by a loss of performances. Nevertheless, this observation needs to be confirmed through experiments of real vehicles.

VI. Conclusion

We have conducted experiments with mini-robots in order to compare two protocols which are CVAS and TIM. CVAS is interesting for humans because it allows discrete signaling system while it permits to enhance the sequence by executing a real-time optimization algorithm such as DCP. TIM requires autonomous vehicles. It offers better performances and thus the constraint of using autonomous vehicles is justified.

Several issues deserve further investigations. Indeed, it is necessary to propose new results involving more vehicles in a more complex configuration of intersection. It is also necessary to prove the efficiency of each protocol using real vehicles equipped with efficient available technologies. Finally we have to consider the safety
issue for both human driver protocol and automated vehicle protocol.

In a future work we will define clearly the TIM protocol and we will develop a new sequence-based policy able to improve the evacuation time.

REFERENCES


