



On the conflict with dioid algebra: Autonomous Intersection Management

Abdeljalil Abbas-Turki, Mourad Ahmane, Fei Gao, Wu Jia, Abdellah El-Moudni, Abdellatif Miraoui

► To cite this version:

Abdeljalil Abbas-Turki, Mourad Ahmane, Fei Gao, Wu Jia, Abdellah El-Moudni, et al.. On the conflict with dioid algebra: Autonomous Intersection Management. IEEE International Conference on Industrial Electronics and Applications (ICIEA), Jul 2012, Singapore, Malaysia. pp. 1018 - 1023. hal-00808582

HAL Id: hal-00808582

<https://inria.hal.science/hal-00808582>

Submitted on 5 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the Conflict with Dioid Algebra: Autonomous Intersection Management

Abdeljalil Abbas-Turki, Mourad Ahmane, Fei Gao, Jia Wu, Abdellah El-Moudni and Abdellatif Miraoui
Laboratoire Systèmes et Transport
Université de Technologie de Belfort Montbéliard
90010 Belfort Cedex, France
(e-mail: prenom.nom@utbm.fr)

Abstract—Autonomous Intersection Management (AIM) is an intersection through which vehicles are independently directed according to a given policy. This is an innovative approach based on a positioning system and on the wireless negotiation of the "right of way". However, AIM raises the issue of the feasibility that is strongly related to whether the vehicles are able to cross the intersection without collisions. In this paper, instead of vehicles we used robots in order to prototype an elementary AIM. For that, we present a Petri net model of an elementary AIM. The Petri net model is used to define a centralized protocol that allows the robots to cross the intersection without collisions. This protocol uses an event observer that controls the distribution of "rights of way" to the robots. The event observer is based on a linear system of inequalities under constraints. An elementary AIM of robots is developed. Tests show that the protocol allows a safe intersection with two different scheduling algorithms. Although delayed and several losses of messages, collisions and deadlocks are avoided by means of the event observer.

I. INTRODUCTION

Discrete Event Dynamic Systems (DEDS) are modeled by using Petri Nets (PN) from which mathematical equations are derived to describe the dynamic of the studied system. Dioid algebra is used for analyzing and controlling DEDS. The main advantage of this algebra is the linearization of the equations that models the synchronization between the events of the system, by using the sequences and the series [1]. Nevertheless, the use of dioid algebra is generally limited to Timed Event Graph (TEG) which does not admit conflicts [2].

Some works have attempted to overcome this drawback. New approaches for considering the conflict are introduced [3-7]. However, to the best of our knowledge, there is no a unanimous approach that matches all the raised issues of the conflict, such as deadlocks, combinatorial optimization and so on. These approaches either suffer from combinatorial explosion or are adapted to particular applications such as known solutions to the conflicts. In the following, we will focus on the approach introduced in [9]. This approach allows the modeling of a PN composed of a TEG and an Invariant Resource Sharing System (IRSS). It proceeds as follows. Firstly, the TEG is isolated and modeled by a linear system. Then, the system is submitted to the constraints of invariant marking because of the invariance of the resource.

The scope of this paper is to use this approach in order to guarantee the safety of an autonomous intersection that is

an intersection without any observable signalization. Robots wirelessly negotiate the "rights of way" through an intersection manager. The main issue is to avoid collisions due to delayed or missed of messages. This paper introduces the use of an event observer, which is based on the constraints presented in [9], in order to provide a safe autonomous intersection.

This paper is organized as follows. Firstly, it introduces the autonomous intersection which is the studied system. Then, it gives the analytical model based on Timed Petri net and dioid algebra. The derived mathematical model and the constraints are programmed into a microprocessor that centralizes the negotiations. Two negotiation policies are tested [10-11]. Feedbacks are discussed before the conclusion.

II. AUTONOMOUS INTERSECTION

We call autonomous intersection [8,10-12], an intersection that is managed without any visible signalization. Autonomous intersection decides the sequence of robots that cross the intersection instead of computing the green time given to a particular movement. This allows a customized signalization based on a scheduling algorithm. The schedule is computed according to a given policy such as priorities, respect of the due dates and so on. Each robot wirelessly sends the requests of "right of way" and waits for receiving the "right of way" before crossing the intersection. Autonomous intersection requires equipped robots in order to make them able to: (i) know their positions and (ii) communicate with the surrounding environment. Other equipments are required, for instance the frontal distance measurement is required in order to make each robot follow safely the movement of the ahead robot.

A. Studied System

The studied system is an elementary intersection of two lanes. Each lane is one-way. Each robot that moves in a particular lane observes the three following events: The arrival, the access to the intersection and the departure. So respectively, each lane is split into three zones: the storage zone, the conflict zone and the leaving zone. Color rectangles are used to give the position to the robots (see Fig 1). Two different colors are used: blue for Lane 1 and pink for Lane 2. We call the color rectangles 'position markers'. Each robot is equipped with a color sensor to recognize the position markers (see Fig 2).

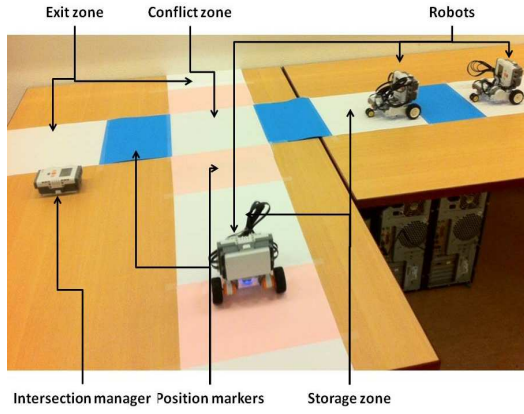


Fig. 1. Prototype of an elementary intersection system

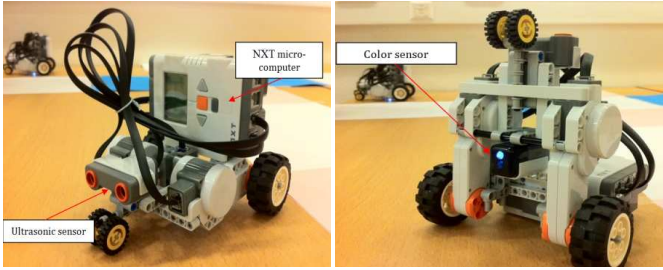


Fig. 2. Robot: Front view and bottom view

So, a discrete point positioning technology is used in order to confirm or to correct the estimated position of the robot.

The position markers are ordered according to the nominal direction of the lane. When a robot detects the first position marker, it sends a request for the "right of way" to the intersection manager. The robot cannot cross the second position marker without a "right of way". If the robot detects the third position marker, it returns the received "right of way" to the intersection manager.

B. Intersection Manager

The intersection manager is a microprocessor able to communicate with the robots. Three modules are connected in order to form the intersection manager:

- A communication module receives data from robots and sends the "rights of way" to the robots;
- An event observer evaluates the performance and avoids collisions;
- A scheduler computes the sequence according to a given strategy.

The interactions between the three modules are shown in Fig3. The communication module receives the requests from the robots. The requests are analyzed by the event observer to evaluate the performance of the intersection. The performance evaluations with the requests are sent to the scheduler that proposes a sequence. According to the sequence, the event observer selects the robots that will receive the "rights of way". These "rights of way" are sent to the robots by means of the

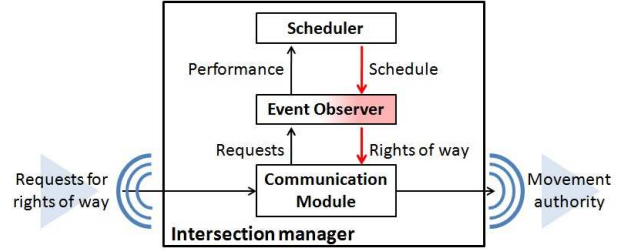


Fig. 3. Architecture of the intersection manager

communication module. So, the event observer is in the core of the safety issue because it has to prevent the collisions of robots.

C. Safety Issue

From the description given previously, the safety issue is supported by a centralized intersection manager and a discrete point positioning system. More precisely, the event observer receives the data from the robots according to the detected position markers. Moreover, the position markers help the robots to precisely estimate their positions at the critical points of the infrastructure. In other words, the three position markers are required for avoiding collisions, even if the second position marker seems to be the only direct answer to the safety issue because it makes stop the robot prior the conflict zone. This is due to the fact that the movement of the robots is not as predictable as we can usually expect [13]. Besides, wireless communication is not as quick and as reliable as in theoretical study. Indeed, the saturation of bandwidth has already shown that collisions are possible with two position markers [14].

So, a centralized event observer is required to make the robot safely cross the intersection. Indeed, it centralizes the presence list and makes a selection of the robots that actually will receive the "right of way". So, the event observer is in the core of the safety system. For this reason, the event observer is in the sandwich, between the received requests and the distribution of the "rights of way". The event observer is based on the P-Timed PN model presented in the next section.

III. AUTONOMOUS INTERSECTION

Without loss of generality, we assume the elementary autonomous intersection of two symmetric lanes depicted into Fig 4. To model the elementary intersection system by Petri net, we take into consideration the following parameters.

Temporal parameters of the intersection:

- According to the speed of the robots, we can define the time required to cover each zone delimited by two successive position markers. This travel time is denoted by τ_{j-j+1} . Hence, τ_{j-j+1} is the temporal distance between the position marker j and the next position marker $j+1$. For instance, τ_{1-2} is the minimum travel time required to cover the storage zone.
- We take also into consideration h which is the minimum headway time between two successive robots.

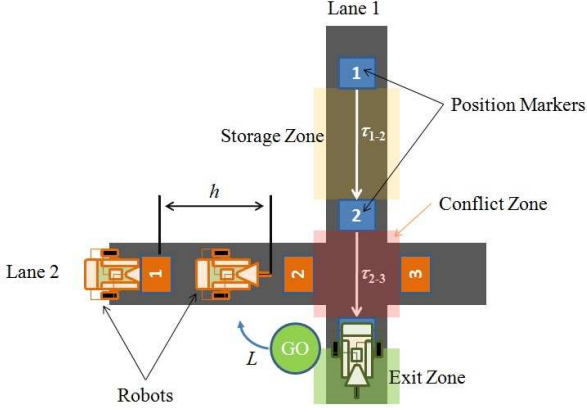


Fig. 4. An elementary autonomous intersection

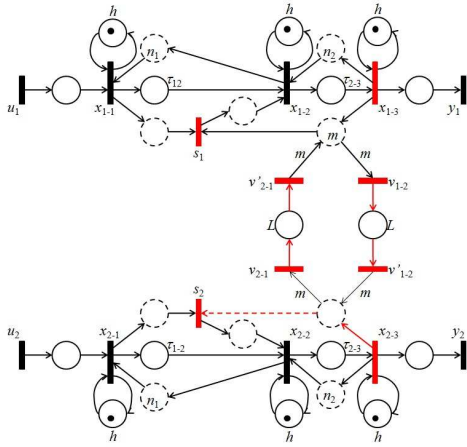


Fig. 5. P-timed PN model of an elementary autonomous intersection

- A lost time due to the kinematic of the stopped robots and to the communication delay is considered and denoted by L . We note that the different parameters τ_{j-j+1} , h and L defined above are determined empirically.

Capacity parameters of the intersection:

- n_{j-j+1} is the maximal possible number of robots between two successive position markers j and $j+1$.
- m is the maximal "rights of way" that can be distributed simultaneously.

1) *P-timed Petri Net Model*: The P-timed PN model is presented by Fig 5. Since the two lanes are symmetric the model is symmetric too. The firing of transition x_{i-j} represents a robot that moves in the lane i and observes the position marker j . Once a robot observes the first position marker transition x_{i-1} is fired. The robot waits for receiving the "right of way" before crossing the conflict zone. The reception of the "right of way" is represented by transition s_i . The robot is able to cross the conflict zone only if the transition s_i is fired. The firing of transition x_{i-2} means that the robot is accessing to the conflict zone. The firing of transition x_{i-3} denotes the event of leaving the conflict zone. It sends the received "right

of way" to the intersection manager. The intersection manager can continue to distribute the "rights of way" to the robots that move in the same lane or to the robots that moves in the conflicting lane. In other words, the token that represents the "right of way" can fires either transition s_i or transition $v_{i-i'}$, respectively. However, for safety reasons, transition $v_{i-i'}$ cannot be fired before receiving all distributed "rights of way". This is the reason why a weight m is added to the input arc of transition $v_{i-i'}$. In practice, a time L separates the moment when of receiving all the distributed "rights of way" and the distribution of the "rights of way" to the robots that move in the other lane i' . This is represented by the timed place between transitions $v_{i-i'}$ and $v'_{i-i'}$. The intersection manger is able to distribute the "rights of way" to the robots that move in lane i' only if transition $v'_{i-i'}$ is fired. A time $L \neq 0$, is only used for the performance evaluation.

The timed places are presented by solid circles. The meaning of times and notations are given previously. The process of the distribution of the "rights of way" is represented in red. The conflict between transitions s_i and $v_{i-i'}$ is represented by dashed arrows. The model represented in red is in the core of the safety issue. Indeed, it prevents two conflicting robots from receiving the "right of way" simultaneously. Thus, the event observer has to allow the distribution of the "rights of way", accordingly. This is the reason why a correct analytical model is required, which is given follows in dioid algebra. Thus, a brief introduction of this algebra is given below.

2) *Brief Introduction to Dioid Algebra*: Dioid algebra is a powerful mathematical tool that allows a linear description of a Timed Event Graph (TEG). A TEG is a PN in which each place has exactly one input transition and one output transition. It is possible to write the $(\min, +)$ or $(\max, +)$ -linear state equations that describe the firings of the transitions of a TEG. To obtain a linear model, we can associate each transition T either with a counter function, which is a mapping $Z \rightarrow \bar{Z}_{\min}$, $t \rightarrow T(t)$ that gives the number of times that T has been fired before the date t , or dually with a dater function, which is a mapping $Z \rightarrow \bar{Z}_{\min}$, $t \rightarrow T(k)$ that gives the date of the k -th firing of transition T . The dioid \bar{Z}_{\min} , commonly known as $(\min, +)$ -algebra, is the set $\bar{Z} = Z \cup \{-\infty, +\infty\}$ endowed with these two following operations: $a \oplus b = \min(a, b)$ and $a \otimes b = a + b$. Its neutral element is $\epsilon = +\infty$, its identity element $e = 0$ and its top element is $\top = -\infty$. In $(\max, +)$ -algebra, $a \oplus b = \max(a, b)$, $\epsilon = -\infty$ and $\top = +\infty$. A partial order in a dioid can be defined by the following equivalency, $\forall a, b \in \mathcal{D}$: $a \succeq b \iff a = a \oplus b$, where \mathcal{D} is a set endowed with two closed operations called addition (\oplus) and multiplication (\otimes).

The residuation theory provides a pseudo division. We denote $R_a^\# = y/a = \oplus\{x \in \mathcal{D} | x \otimes a \leq y\}$ the residuated of the mapping $R_a : x \mapsto x \otimes a$. In [15], the authors remind that in \bar{Z}_{\min} , $b \oslash a = b - a$ if a and b are finite. In [9], the authors introduce the operator \triangleright , which has the interesting property of preserving markings. \triangleright is defined as follows: $c \triangleright c = \begin{cases} e & \text{if } c = \epsilon \\ c \otimes 1 & \text{elsewhere} \end{cases}$. Like the Z -

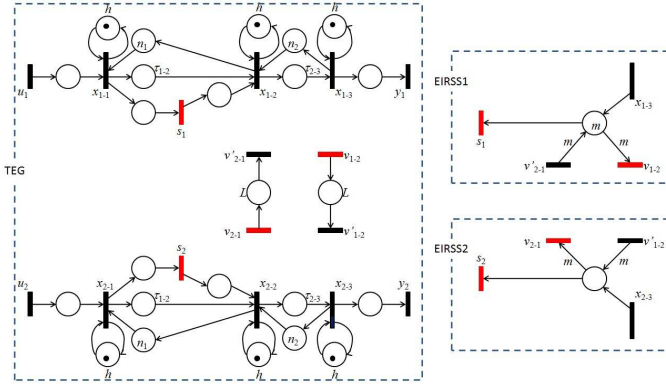


Fig. 6. Decomposition of the P-timed PN model

transform for series in classical algebra, the γ and δ -transforms allow to translate dater and counter functions to formal series, respectively: γ is the backward shift operator in the event domain, i.e. $x(k-1) \xrightarrow{\gamma} \gamma x(\gamma)$. δ is the backward shift operator in the time domain, i.e. $x(t-1) \xrightarrow{\delta} \delta x(\delta)$.

The behavior of a whole TEG can be described by state equations that are linear in a dioid of formal series in two commutative variables γ, δ with exponent in \mathbb{Z} and with Boolean coefficients (see [17]). This dioid denoted $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ allows to study both time and event domains together. An element $s \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ is defined by $s(\gamma, \delta) = \bigoplus_{i \in \mathbb{Z}} f(n_i, t_i) \gamma^{n_i} \delta^{t_i}$ with $f: \mathbb{Z}^2 \rightarrow B$ a Boolean mapping. When an element of $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ is used to code information concerning a transition of a TEG, each of its nominal $\gamma^k \delta^t$ can be interpreted as the k -th event occurs at least at date t .

3) *Analytical Model:* A TEG cannot afford concurrency. In [9] the authors has proposed a general model of IRSS using dioid algebra. IRSS is a DEDES in which it is possible to identify one set of a finite and a constant (i.e. an invariant) number of resources and its users. One can note from Fig 5 that the system of the distribution of the "rights of way" is a process of sharing a resource (conflict zone). More precisely, we have the following two IRSS formed by (see Fig.6): (i) Transitions x_{1-3} , s_1 , v_{1-2} and v'_{2-1} (IRSS1) and (ii) Transitions x_{2-3} , s_2 , v_{2-1} and v'_{1-2} (IRSS2). The resources are the "rights of way". For IRSS1, the users are robots that come from Lane 1 and the intersection manager who would distribute the "rights of way" to the robots that move in Lane 2. Similarly, the users of IRSS2, are the vehicles that move in Lane 2 and the intersection manager. In both IRSS, the intersection manager needs all the "rights of way" free at the same time. Thus, the arcs that link transitions v_{1-2} , v'_{2-1} , v_{2-1} and v'_{1-2} are weighted. There are initially m available resources in IRSS1 and no available resources in IRSS2 (see Fig.6).

Setting the following column-vectors $U = [u_1, u_2]^T$, $X = [x_{1-1}, x_{1-2}, x_{1-3}, v'_{1-2}, x_{2-1}, x_{2-2}, x_{2-3}, v'_{2-1}]^T$, $Y = [y_1, y_2]^T$, $D = [v_{1-2}, s_1, v_{2-1}, s_2]^T$, with $[.]^T$ denotes the transpose operator, the analytical model is given in Fig 7. Equalities (1) and (2) represent the behavior of the extracted TEG (see Fig.6). One note a vector D which its entries

concern the transitions that take a part of the decision making in order to resolve the two conflicts of IRSS1 and IRSS2. These transitions are red in Fig.6. The reminder equations are the constraints of the system. Inequality (3) means that the "right of way" cannot be distributed without a request from a robot. This highlights the difference between the autonomous intersection management and the traffic lights. Indeed, the green light can be given to a lane without any presence of robots. Inequalities (4) and (5) concern IRSS1 and IRSS2, respectively. The weights on arcs that link transitions v_{1-2} , v'_{2-1} , v_{2-1} and v'_{1-2} are considered by using m times-multiplication of the resource held by the intersection manager.

4) *Practical Issue :* The analytical model presented in Fig 5 is programmed into the microprocessor that represents the intersection manager, according to the architecture shown in Fig 3. So, the event observer is the module that needs the analytical model, in order to appropriately prevent robots from collisions. Equalities (1) and (2) are used to evaluate the instantaneous performance of the network and inequalities (3), (4) and (5) are used to safely distribute the "rights of way" to the robots. Since the dater and the counter functions are monotonic increasing series, the system is initialized if the intersection is empty. Initially, the m "rights of way" are available according to the lane that observes the first arrival of a robot. The behavior of a robot needs also to be controlled according to the model. Thus, the PN model is implemented into each embedded microprocessor that controls the movement of the robot. The blue transitions order the robot to send a message to the intersection manager. The red transition is fired if a "right of way" is received by the robot. However, sending and receiving messages follow the robust protocol presented in [17], in order to overcome the lack of missed and delayed messages.

When the robot observes the first position marker, it sends a message that informs the intersection manager about the ID of the robot, the ID of the transition, the date of firing of transition x_{i-1} . From this moment, the robot is waiting for a message ("right of way") from the intersection manager. The robot stops prior to the conflict zone if it does not obtain the "right of way". When the robot fires transition x_{i-3} it sends a message to the intersection manager.

One can note that the firing of transitions x_{1-2} and x_{2-2} is not a subject for the communication between the intersection manager and the robot. Indeed, inequalities (4) and (5) are only based on transitions s_1 and s_2 . So, for testing the mathematical model we limit our investigation to only these two constraints.

IV. EXPERIMENTS

In this paper, we present an autonomous intersection of robots. For the experiments we used the NXT technology. This technology is capable to assess the ability of the proposed analytical model to make the intersection system operates accurately. Two policies are tested in order to check whether or not the proposed model is open for integrating different scheduling algorithms. Moreover, missed message and delays allows testing the feasibility of the autonomous intersection

$$\begin{cases}
(1): X = \begin{pmatrix} \gamma^{\delta^h} & \gamma^{n_1} & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \delta^{\tau_{1-2}} & \gamma^{\delta^h} & \gamma^{n_2} & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \delta^{\tau_{2-3}} & \gamma^{\delta^h} & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \gamma^{\delta^h} & \gamma^{n_1} & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \delta^{\tau_{1-2}} & \gamma^{\delta^h} & \gamma^{n_2} & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \delta^{\tau_{2-3}} & \gamma^{\delta^h} & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{pmatrix} X \oplus \begin{pmatrix} \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & e & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \delta^L & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & e \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \delta^L & \epsilon \end{pmatrix} D \oplus \begin{pmatrix} e & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & e \\ \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \end{pmatrix} U \\
(2): Y = \begin{pmatrix} \epsilon & \epsilon & e & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & e & \epsilon \end{pmatrix} \\
\text{Under the following constraints} \\
(3): D \preceq \begin{pmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ e & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & e & \epsilon & \epsilon \end{pmatrix} \\
(4): (s_1^\triangleright(t) \otimes x_{1-3}^\triangleright(t)) \otimes (v_{1-2}^\triangleright(t) \otimes v'_{2-1}{}^\triangleright(t))^{\otimes m} \succeq_{Z_{min}} m \\
(5): (s_2^\triangleright(t) \otimes x_{2-3}^\triangleright(t)) \otimes (v_{2-1}^\triangleright(t) \otimes v'_{1-2}{}^\triangleright(t))^{\otimes m} \succeq_{Z_{min}} e \quad \text{where by notation: } (x^\triangleright(t))^{\otimes m} = \underbrace{(x^\triangleright(t) \otimes \dots \otimes x^\triangleright(t))}_{m\text{-times multiplication}}
\end{cases}$$

Fig. 7. The analytical model

management. So, first we discuss the NXT wireless communication that is based on Bluetooth.

A. NXT communication

The NXT Bluetooth system is designed to run in a master/slave configuration. In other words, three or more NXT micro-computers can only communicate together under a centralized architecture. Hence, each robot cannot directly communicate with the other robots. The intersection manager is chosen to be the node of the wireless network (master). The NXT communication is used to support all the negotiation between each pair of robots.

The NXT master can only communicate with one slave source at a time. For instance, if the intersection manager is communicating with robot 2 it must finish sending or receiving data before it can communicate with robot 3. So, the intersection manager receives messages from robot 1, from robot 2 and after from robot 3. This allows a better order of the received messages. However, it is only a partial solution of the order problem that leads to a deadlock. For instance, if robot 1 sends a request of "right of way" before robot 3, the request received from robot 3 can be treated before, since: (i) The first request of robot 1 is missed, (ii) The request of robot 1 is sent when the intersection manager is hearing messages from robot 2. Nevertheless, since the date of firing of transition x_{i-1} is sent by a robot, the scheduler resolves the problem of deadlock.

The communication between a robot and the intersection manager is periodically maintained until they are sure that the message is received. The optimal periods are empirically obtained. Each robot sends a message every 90ms and the intersection manager sends a message every 150ms. The results of this combination are presented in Fig 8 and Fig 9. From those figures, one observe that the performances of messages sent by the robots are low. Indeed, the communication time is about 2s and the success rate is lower than 7%.

Controller to All targets:	All targets to Controller:
Messages status:	Messages status:
Attempt send: 991	Attempt send: 1806
Success send: 290	Success send: 138
Ratio: 29.26 %	Ratio: 7.64 %
Messages communication delay:	Messages communication delay:
Average: 230.80 ms	Average: 785.12 ms
Min: 31 ms	Min: 674 ms
Max: 402 ms	Max: 1097 ms
Std deviation: 93.07	Std deviation: 74.99
Success messages time interval:	Success messages time interval:
Average: 918.88 ms	Average: 1945.56 ms
Min: 272 ms	Min: 750 ms
Max: 6825 ms	Max: 7350 ms
Std deviation: 752.44	Std deviation: 1313.81

Fig. 8. Global statistics: Robot's cycle time 150ms. Intersection manager's cycle time 90ms.

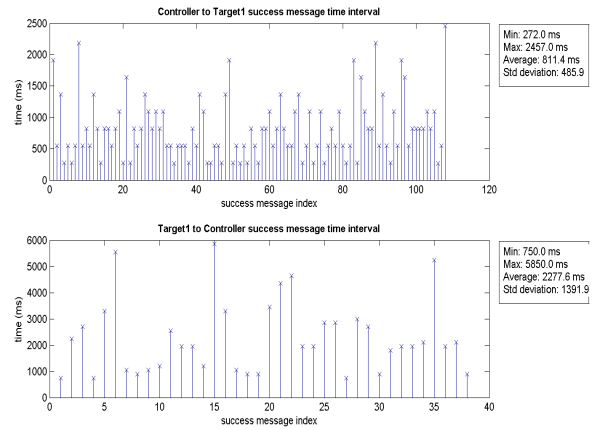


Fig. 9. Communication delays between robot 1 and the intersection manager during 9s.

B. The Two Tested Policies

The first tested policy is presented in [10]. A First Come First Served (FCFS) policy is used for directing robots through

the intersection. This policy does not need a negotiation. The scheduler just orders the presence list of the robots according to the dates of firing of transitions x_{i-1} . The main objective of testing FCFS is to easily assess the behavior of the distribution of the "right of way". Indeed, it is easy to compare the order of robots' arrivals and the order of robots' departures.

The second policy is based on the dynamic programming algorithm presented in [11]. The algorithm aims to empty the intersection as soon as possible by minimizing the completion time. Nevertheless, the complexity of the algorithm does not match the computation capacity of the NXT microprocessor. So, it was simplified by introducing a kind of negotiation between the robots through the intersection manager. So, we keep the same architecture. However, one digit of negotiation is added to the messages sent by the robots. This digit is not treated by the event observer. It is directly transmitted to the scheduler that accordingly computes the sequence of the distribution of the "rights of way".

Both policies are programmed by using LabView. The basic configuration is FCFS. The event observer selects the "rights of way" that can be transmitted to the robots without any risk of collision. As previously discussed, this selection is based on the inequalities constraints (4) and (5).

C. Feedbacks

Many tests of both policies have been done in our Laboratory. A test takes at most 30s. Each policy has been tested 200 times. The robots have been randomly distributed and started. Moreover, the prototype has been exposed to public, in different events. The most important one was: "Journée portes ouvertes sur l'UTBM" [18], where a public with a wide range of age and of scholar level has manipulated the prototype and has tested several scenarios.

First, the test of the FCFS policy shows us that the order of the robots arrivals is the same than the order of the robots departures. However, one deadlock is recorded and resolved by the scheduler. It is due to the fact that messages are not received according to the sequence of the arrival of robots. For both policies, stopped robots are only due to the communication delay. Indeed, the screen of the intersection manager displays that the "right of way" is addressed to the stopped robot. Moreover, not more than 1s after the stop, the delayed robot passes over the second position marker. Finally, all tests, done in the laboratory, do not reveal any collision.

However, one collision has been observed during the exhibition of the robots to the public. The manipulation of the robot has modified the position of the color sensor. So, the robot did not detect the position marker. This means that the proposed AIM depends on a good evaluation of the position. Regardless this fact, we can conclude that the behavior of the intersections validates the analytical model.

V. CONCLUSION AND FUTURE WORKS

This paper has presented the use of mini robot to test the feasibility of AIM. AIM requires wireless communications and positioning technologies. The presented prototype is based on

NXT technology. Low performance of the provided wireless communication is observed, with regard to the ambition of a real time application. This raises issues of safety and of eventual deadlock. In order to overcome these drawbacks a robust protocol is proposed. This protocol is based on centralized intersection manager. The latter collects the events observed by the robot and decides exactly the robots that receive the "right of way". The intersection manager is based on Petri net and dioid algebra models. This is tested through two policies for directing robots through the intersection.

Several issues deserve further investigations. The first one consists into extending the proposed protocol to ordinary vehicles, by using the onboard signalization. In a near future, this protocol will be tested in real world. The second one is dedicated to the scheduling algorithms.

REFERENCES

- [1] F. Baccelli, G. Cohen, G.J. Olsder and J-P. Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems*. J. Wiley, 1992.
- [2] R. David and H. Alla. *Du GraFCET aux réseaux de Petri*. Hermès, 1992.
- [3] B. Trouillet and A. Benasser. *Cyclic scheduling problems with assemblies: an approach based to the search of an initial marking in a marked graph*. IEEE Int. Conf. on Systems, Man and Cybernetics, vol.3, Tunisia, 2002.
- [4] T.J.J. Vanden Boom and B. De Schutter. *Modelling and control Of discrete event systems using switching max-plus-linear systems*. Control engineering practice, vol.14, n.10, pp.1199-1211, 2006.
- [5] M. Al Saba, J.-L. Boimond and S. Lahaye. *On just-in-time control of flexible manufacturing systems via dioid algebra*. IFAC Symp. on Information Control in Manufacturing (INCOM), vol.2, pp.137-142, 2006.
- [6] O. Boutin, B. Cottenceau, J.-J. Loiseau and A. L'Anton. *Shared resources in production systems: (max,+) analysis*. Int. J. of Mathematics in Operational Research, vol.3, n.2 pp.125-147, 2011.
- [7] K. Munenori, T. Hirota and G. Hiroyuki. *Resolution of Resource Conflict Between the Different Two Projects in MPL-CCPM Representation*. Information Science and Application. WSEAS transaction on, Issue.6, vol.7, pp 774-783, 2010.
- [8] N. Neuendorf and T. Bruns. *The vehicle platoon controller in the decentralised, autonomous intersection management of vehicles*. Mechatronics. ICM '04. Proceedings of the IEEE International Conference on, 2004.
- [9] A. Corrêa, A. Abbas-Turki, R. Bouyekhf and A. El Moudni. *A dioid model for invariant resource sharing problems*. IEEE Trans. Syst. Man Cybernet. Part A Syst. Hum. 39 770-781. 2009.
- [10] K. Dresner and P. Stone. *Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism*. The 3rd Int. Joint Conf. on Autonomous Agents and Multiagent Systems, 530-537, 2004.
- [11] J. Wu, A. Abbas-Turki, A. Corrêa and A. EL Moudni. *Discrete Intersection Signal Control*. IEEE International Conference on Service Operations and Logistics, and Informatics, pp 27-29, Aug 2007.
- [12] A. De La Fortelle. *Analysis of reservation algorithms for cooperative planning at intersections*. ITSC, 13th International IEEE Conf. on., 2010.
- [13] M. Quinlan, A. Tsz-Chiu, J. Zhu, N. Sturca and P. Stone. *Bringing simulation to life: A mixed reality autonomous intersection*. IROS, IEEE/RSJ International Conference on., 2010.
- [14] M. Grünewald, C. Rust and U. Witkowski. *Using mini robots for prototyping intersection management of vehicles*. Proc. of the 3rd Int. Symposium on Autonomous Minirobots for research and edutainment, pp 288-294, 2006.
- [15] G. Cohen, S. Gaubert, R. Nikoukhah and J.-P. Quadrat. *Second order theory of min-linear systems and its application to discrete event systems*. IEEE international (CDC), England, Dec. 1991.
- [16] G. Cohen, P. Moller, J.-P. Quadrat and M. Viot. *Algebraic tools for the performance evaluation of discrete event systems*. Proc. of the Institute of Electrical and Electronics Engineers, vol.77, n.1, pp 39-58, 1989.
- [17] M.K. Lars and K. Jensen. *Teaching Modelling and Validation of Concurrent Systems Using Coloured Petri Nets*. Transactions on Petri Nets and Other Models of Concurrency I. Lecture Notes in Computer Science, Vol.5100, 19-34, 2008.
- [18] http://www.utbm.fr/sites/default/files/gestionFichiers/docs/Actualites/Dossiers-a-telecharger/FLY_JPO2011_WEB.pdf