Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-12 (2016) 497–502

Decentralized management of intersections of automated guided vehicles

Alexandre Lombard * **Florent Perronnet** **
Abdeljalil Abbas-Turki * **Abdellah El Moudni** *

* *IRTES-SeT, Université de Technologie de Belfort-Montbéliard*
Belfort, France
** *Voxelia, Belfort, France*

Abstract: The emerging topic of cooperative intersection management for vehicles has raised up new solutions for traffic control in order to avoid collisions, deadlocks, and improve traffic efficiency. The solutions developed for road traffic can easily be applied to automated guided vehicles to overcome the common drawbacks, and improve the number of vehicles in a network. In the context of a network of vehicles only regulated at intersections, we propose an algorithm in order to prevent deadlock at intersections in a network of automated guided vehicles.

Keywords: automated guided vehicles; cooperative intersection management; wireless communication; deadlock

1. INTRODUCTION

In a network with a large number of automated guided vehicles (AGV), traffic control is crucial to the system performance. The traffic control must avoid collisions, deadlocks/gridlocks and ensure that every AGV can reach its destination, and for this purpose different solutions are proposed:

- To avoid collisions, zone control with wireless transmission is the favorite traffic control of most environments as it is simple to install and expand.
- To avoid deadlocks/gridlocks in the network of AGV, the mainly used solutions are improved banker's algorithms (Lawley et al., 1998; Ezpeleta et al., 2002; Bobanac and Bogdan, 2008), and the use of Petri nets (Wu and Zhou, 2007), but they both suffer from drawbacks. The banker's algorithm can eliminate valid solutions, significantly compromising the efficiency of the management, while Petri nets can cause livelocks. Moreover the number of AGV in the network with these solutions is limited: a resource is an arc between two nodes and can only be held by one AGV at a time. Other solutions relying on siphon-based deadlock prevention are optimal but the minimal siphon is a NP problem, thus it cannot be considered for real time applications.
- Conflict-free routing based solutions are also proposed in Nishi and Tanaka (2012) and Miyamoto and Inoue (2016), they prevent the intersection problem but they require a centralized management, and some valid solutions are eliminated.

Due to the recent development of cooperative intersection management for road vehicles, new solutions have emerged to improve traffic efficiency with both collision and deadlock avoidance. One can quote the well-known Reservation-Based Protocol (RBP) where the vehicle sends a reservation request of space and time to the server in Dresner and Stone (2004) and de La Fortelle (2010). Other works are based on Cooperative Adaptive Cruise Control at Intersections (CACCI). The server detects conflicts and accordingly sends acceleration and deceleration messages to vehicles in order to avoid collisions in Zohdy and Rakha (2012) and Zohdy et al. (2012). Furthermore a Sequence-Based Protocol (SBP) is proposed in Perronnet et al. (2013). It assumes that the intersection is controlled as follows : either the intersection manager or a decentralized negotiation explicitly determines the sequence of vehicles in each conflict area. The sequence determines which vehicle is the first, which one is the second and so on.

Though these protocols are designed for isolated intersection, they can be integrated in a more global solution in charge of managing (routing and gridlock prevention) a fleet of vehicles, like Bocewicz et al. (2007) and Perronnet et al. (2014). For instance, in Perronnet et al. (2014) a protocol is proposed to avoid gridlock (deadlock caused by the interaction of multiple intersections) in a network of intersections allowing more than one vehicle per zone between intersections. This algorithm is based on the principle of path reservation: a vehicle has to follow a given path, it asks the global server for the authorization to follow this path, then the global server reserves the path of this vehicle and informs intersection servers. Then, eventual intersections of vehicles are locally managed by one of the previously presented solution with respect to the constraints stated by the global protocol.

These solutions are originally designed for road vehicles but can be applied to AGV in order to increase the number of AGV in a network, and improve the overall performance. Therefore, in the context of this paper, we assume a network with a traffic control protocol in charge of the gridlock prevention allowing more than one vehicle between two intersections. We then consider an intersection in this network with multiple unlocked vehicles, i.e. vehicles that

10.1016/j.ifacol.2016.07.669

can move without causing a gridlock. In order to avoid collisions between AGV, we assume that the intersection is regulated with an intersection server using a SBP and a first-in, first-out scheduling policy.

According to the chosen protocol, the inconsistency of the presence list brings different risks. Deadlock and collision are then possibles. In the following we will consider a centralized architecture of SBP (C-SBP) named Transparent Intersection Manager (TIM) (Perronnet et al., 2013). In TIM vehicles synchronize their speeds according to the presence list received from the server. There are two advantages of C-SBP. The first one is the default-deny: if a vehicle is not able to establish a communication with the server, it has to brake before the box junction. The second one is that the results of C-SBP can easily be extended to RBP.

However, due to potential communication problems (message losses), the sequence built according to the order of arrival of messages from the AGV can be different from the physical order of the vehicles, resulting in a deadlock situation. The scope of this paper is to propose a re-sequencing algorithm able to avoid the deadlock without introducing any risk of collision (3), even with an unreliable communication. In order to assess the proposed algorithm, simulations and intersections of robots are performed.

This paper is organized as follows; first it presents the protocol TIM and the conditions of the problem. Then, the paper introduces the deadlock problem as well as the collision risk due to a bad re-sequencing. Therefore, it presents the re-sequencing algorithm and shows that the resulting sequence is collision-free. Before concluding, the paper discuss the results of simulations as well as the results of the cooperative intersection of robots.

2. PROBLEM STATEMENT

2.1 Brief presentation of the Transparent Intersection Manager

Vehicles in TIM are able to observe the obstacles in the surrounding environment and to adapt their speed accordingly. AGV move autonomously if all obstacles are visible by the sensors. In order to enhance the intersection safety and throughput, a negotiation protocol based on wireless communication to synchronize the AGVs speed is proposed by TIM. Every AGV has to use wireless communication to inform an intersection server about its movement. More precisely, every AGV communicates its current position and speed as well as the desired destination and the remaining distance before the exit of the box junction. Accordingly, the intersection server broadcasts these data through an ordered presence list to all the AGV closed to the intersection. In the received presence list, the first vehicle has the highest priority and so on. Each AGV considers three kinds of obstacles:

- Visible obstacles: mainly a precedent AGV in the same buffer lane. Visible obstacles are detected by the sensors without any message received from the server.
- Virtual obstacles: conflicting AGV with higher priority.
- Stop line: the nearest border of the box junction. For safety reason, the AGV has to stop by default before the stop line (default-deny principle).

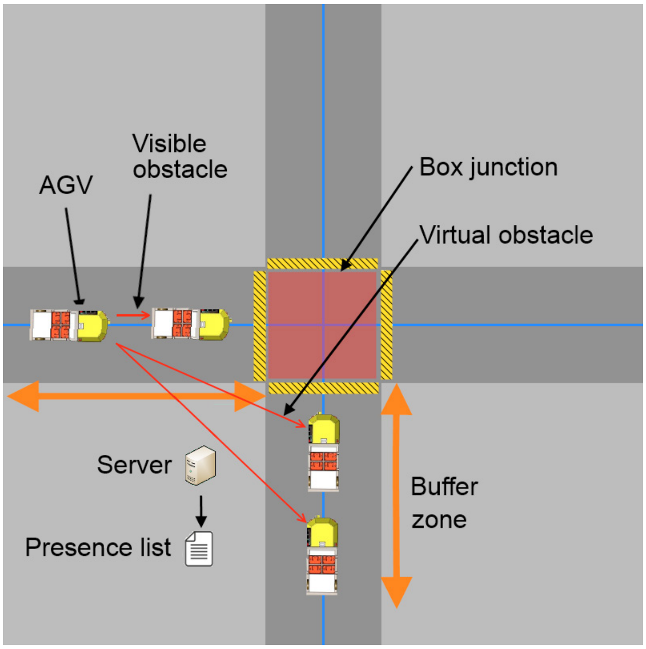

Fig. 1. TIM

The three obstacles are presented in 1. For each obstacle, the AGV computes an acceleration. Hence, there are three computed accelerations:

- a_r to keep a safe distance from the visible obstacles.
- a_i to keep a safe distance from priority conflicting
- a_s to allow a vehicle to stop in the case of a dangerous situation.

The three accelerations contribute to determine the acceleration of the AGV as follows :

- a_i is determined as the minimum acceleration from all conflicting vehicles with a higher priority.
- If the AGV has not received a presence list it has to stop before the box junction. In other words, the intersection map is known before requesting the presence list.
- Vehicles are not allowed to overtake in the buffer zone of the intersection.
- The speed synchronization is done near the junction box.

Formally, the acceleration of each AGV is computed as $a = min(a_r, max(a_i, a_s))$

We draw the reader attention to the fact that only a_i depends on the presence list sent by the server. The vehicle must know the intersection map before getting into the buffer zone. We highlight also the fact that, default deny (a_s) is used when the vehicle has not yet received a presence list or when it is not in the presence list. It is also used when the vehicle is not able to keep a safe distance from the virtual obstacles.